Міністерство освіти і науки України
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт
з дисципліни
“ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ”
для студентів напряму підготовки
6.050103 “Програмна інженерія”
всіх форм навчання

2013

Методичні вказівки до лабораторних робіт з дисципліни “Інтелектуальний аналіз даних” для студентів напряму підготовки 6.050103 “Програмна інженерія” всіх форм навчання / Т.В. Юр, В.М. Льовкін. – Запоріжжя: ЗНТУ, 2013. – 62 с.

Автори: Т.В. Юр, к.т.н., доцент
В.М. Льовкін, ст. викладач

Рецензент: В.І. Дубровін, к.т.н., проф.

Відповідальний
за випуск: В.І. Дубровін, к.т.н., проф.

Затверджено
на засіданні кафедри
програмних засобів

Протокол №1
від “21” серпня 2013 р.

ЗМІСТ

ВСТУП

Метою практичної частини курсу "Інтелектуальний аналіз даних" є отримання практичних навичок з використання розглянутих на лекціях методів аналізу для вирішення практичних задач.

WEKA (Waikato Environment for Knowledge Analysis) – бібліотека алгоритмів машинного навчання для вирішення завдань інтелектуального аналізу даних (data mining). Система дозволяє безпосередньо застосовувати алгоритми до вибірок даних, а також викликати алгоритми з програм на мові Java.

WEKA – продукт університету Уайкато (Нова Зеландія), який вперше був випущений в його сучасному вигляді в 1997 році. WEKA поширюється по ліцензії GNU General Public License (GPL). Це програмне забезпечення написано на мові Java та забезпечує графічний користувальницький інтерфейс для роботи з файлами даних і генерації візуальних результатів (у вигляді таблиць і графіків). Крім того є можливість інтегрувати WEKA, як і будь-яку іншу бібліотеку, у свої власні додатки, наприклад, для автоматизації аналізу даних на стороні сервера, використовуючи стандартний API.

Цілі проекту – створити сучасне середовище для розробки методів машинного навчання та застосування їх до реальних даних, зробити методи машинного навчання доступними для повсюдного застосування. Передбачається, що за допомогою даного середовища фахівець у прикладній області зможе використовувати методи машинного навчання для вилучення корисних знань безпосередньо з даних дуже великого обсягу.

Користувачами WEKA є дослідники в області машинного навчання і прикладних наук. Вона також широко використовується в навчальних цілях.

Теоретичні відомості, присвячені використовуваним алгоритмам інтелектуального аналізу даних, можна отримати з конспекту лекцій з дисципліни «Інтелектуальний аналіз даних» або з рекомендованих літературних джерел. При виконанні лабораторних робіт слід розібратися з кодом реалізації розглянутих алгоритмів у програмі WEKA.

ЛАБОРАТОРНА РОБОТА № 1
ЗНАЙОМСТВО З ПРОГРАМОЮ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ WEKA

1.1 Мета роботи

Ознайомитися та отримати навички роботи з GUI інтерфейсом бібліотеки data mining алгоритмів WEKA. Вивчити можливості, що надаються програмою WEKA.

1.2 Основні теоретичні відомості

Розглянемо можливості GUI інтерфейсу програми WEKA.

1.2.1 Головне вікно програми

Основне вікно програми – це Weka GUI Chooser (рис. 1.1). Опишемо більш докладно призначення керуючих елементів даного вікна.

Рисунок 1.1 – Головне вікно програми WEKA

Основне вікно програми надає доступ до чотирьох модулів програми:

– Explorer – середовище для дослідження даних;

– Experimenter – середовище для проведення порівняльного аналізу роботи різних алгоритмів при обробці наборів даних;

– KnowledgeFlow – середовище, що підтримує таку ж функціональність, як і Explorer, але із застосуванням графічного представлення потоків даних;

– SimpleCLI – командний інтерфейс для безпосереднього виконання команд WEKA.

Головне меню програми складається з чотирьох пунктів.

1. Program:
- LogWindow – відкриває вікно логів, яке зберігає всю інформацію, виведену в потоки stdout або stderr.
- Memory usage – використання пам'яті.
- Exit – вихід.

2. Tools:
- ArffViewer – редактор arff-файлів.
- SqlViewer – модуль перегляду баз даних, для SQL запитів до баз даних за допомогою JDBC.
- Bayes net editor – модуль для редагування, візуалізації та навчання Байесови мереж (Bayes nets).

3. Visualization – засоби візуалізації даних WEKA.
- Plot – відображення 2D-графіка набору даних.
- ROC – відображає раніше збережену ROC-криву.
- TreeVisualizer – відображає спрямовані графи, тобто дерева рішень.
- GraphVisualizer – візуалізує графіку в форматі XML BIF або DOT для Байесови мереж.
- BoundaryVisualizer – дозволяє візуалізувати кордону рішень класифікаторів у двох вимірах.

4. Help – розділ довідкової інформації.
- Weka homepage – домашня сторінка проекту WEKA.
- HOWTOs, code snippets, etc. – приклади, що стосуються розробки та використання WEKA.
- Weka on Sourceforge – сторінка проекту WEKA на Sourceforge.net.
- SystemInfo – значення деяких змінних середовища Java/WEKA.

1.2.2 Модуль Explorer

Це основний модуль програми, який дозволяє завантажити і попередньо обробити дані (вкладка Preprocess), вирішити задачу класифікації або регресії (Classify), кластеризації (Cluster), пошуку асоціативних правил (Associate), відбору атрибутів (Select Attributes) і візуалізації (Visualize).

Кожна задача має свою вкладку в загальному вікні. Спочатку доступна тільки вкладка Preprocess, оскільки для виконання інших завдань потрібні дані. Зазначимо, що послідовність вкладок не завжди

відповідає етапам вирішення задачі. Наприклад, після завантаження даних можна перейти до відбору атрибутів.

Внизу кожної вкладки відображається рядок статусу. Клік правою кнопкою миші на рядку статусу видає контекстне меню:

– Memory information – кількість доступної пам'яті.

– Run garbage collector – запуск збирача сміття Java, що очищає області пам'яті, які більше не використовуються, дозволяючи звільнити пам'ять для нових завдань.

Кнопка «LOG» дозволяє побачити лог подій що відбулися за час роботи WEKA.

У правій частині статусного рядка зображена пташка Ківі. Якщо вона рухається, то програма робить обчислення, якщо сидить нерухомо, то програма знаходиться в режимі очікування. Після значка «X» відображається кількість запущених процесів.

1.2.3 Формат файлів даних ARFF

Основний формат файлів даних, який використовується в WEKA, – це ARFF. У каталозі data знаходяться приклади arff-файлів.

ARFF файл є ASCII текстовим файлом, який описує список об'єктів із загальними ознаками (атрибутами). Структурно такий файл розділяється на дві частини: заголовок і дані.

У заголовку описується ім'я даних та їх метадані (імена атрибутів і їх типи). Наприклад,

```
% коментар
@RELATION myproblem
@ATTRIBUTE firstfeature REAL
@ATTRIBUTE class {A,B}
```

У другій частині представлені самі дані. Наприклад,

```
@ DATA
0,1.1,A
0,4.3,B
```

Заголовок містить інформацію про ім'я файлу і метадані про представлені у ньому даних. Ім'я описується в наступному форматі:

```
@relation <ім’я>
```

Іменем може бути будь-яка послідовність символів. Якщо імя містить пробіли, то воно має бути взято в лапки. Наприклад,

```
@relation weather
@relation ‘weather nominal’
```

Метадані описують атрибути представлених у файлі даних. Інформація про кожний атрибут записується в окремому рядку і включає ім'я атрибуту і його тип. Очевидно, що всі імена повинні бути унікальними. Порядок їх опису повинен збігатися з порядком колонок в описі самих даних. Загальний формат опису атрибуту наступний:

```
@attribute <ім’я атрибута> <тип атрибута>
```

Наприклад,

```
@attribute temperature real
```

Ім'я атрибуту має починатися з символу @. У разі якщо в імені містяться пробіли, воно має бути взято в лапки.

Поле <тип> може мати одне з таких значень:

- real;
- integer;
- <категорія>;
- string;
- date [<формат дати>].

Типи real і integer є числовими. Категоріальні типи описуються переліком категорій (можливих значень). Наприклад:

```
@attribute outlook {sunny, overcast, rainy}
```

Дані представляються в ARFF форматі у вигляді списку значень атрибутів об'єктів після тегу @ data. Кожен рядок списку відповідає одному об'єкту, кожна колонка – атрибуту, описаному в заголовку. Часто в термінології data mining такі рядки називають векторами.

Дані можуть містити припущення (невідомі) значення. У ARFF вони представляються символом «?», Наприклад:

```
@data
4.4,?,1.5,?,Iris-setosa
```

Строкові дані, у разі якщо вони містити символи, що розділяють, повинні братися в лапки. Наприклад,

```
@relation LCCvsLCSH
@attribute LCC string
@attribute LCSH string

@data
AS262, 'Science - Soviet Union - History.'
```

При описі дати можна вказати формат, в якому вона записується. Дати також повинні братися в лапки.

```
@relation Timestamps
@attribute timestamp DATE "yyyy-MM-dd HH:mm:ss"
```

```
@data
"2001-04-03 12:12:12"
```

Кожен набір даних, який використовується в лабораторних роботах, представлений у форматі ARFF. На початку кожного файлу міститься вичерпна інформація про задачу, представлену цим набором даних.

1.2.4 Завантаження і попередня обробка даних (Preprocess)

Спочатку на першій вкладці «Preprocess» вгорі вікна активні чотири кнопки, які дозволяють завантажити дані з файлу (Open file), з віддаленого джерела (Open URL), з бази даних (Open DB) або згенерувати модельні дані (Generate). Найчастіше доводиться користуватися даними з файлу.

Після завантаження файлу на панелі попередньої обробки даних з'являється інформація про дані (рис 1.2).

Натискання на кнопку «Edit» дозволяє редагувати вихідні дані у табличній формі (з'являється вікно «Viewer»).

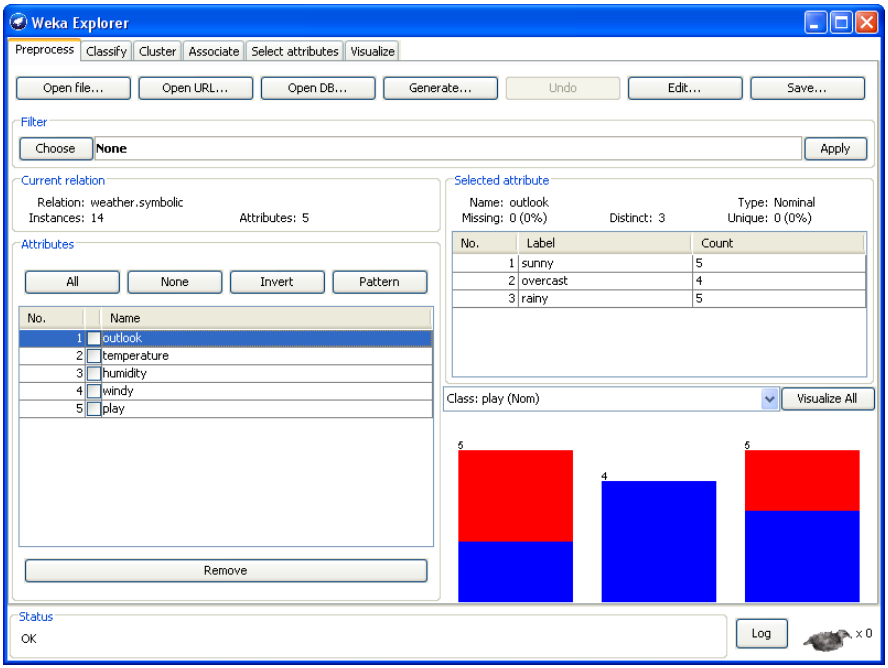


Рисунок 1.2 – Вкладка завантаження та попередньої обробки даних

Панель «Current relation» містить 3 значення:

– Relation – назва відношення (задачі), яка була зазначена у файлі (застосування фільтрів може змінювати цю назву);

– Instances – кількість екземплярів (записів, об'єктів);

– Attributes – кількість ознак (атрибутів).

Нижче знаходиться панель «Attributes» (атрибути чи іншими словами ознаки об’єктів вибірки даних). На ній знаходяться чотири кнопки і список атрибутів. Атрибути розташовані в тому порядку, в якому вони визначені в файлі даних. Список містить три колонки:

– No – номер, що ідентифікує атрибут та використовується для вказівки на атрибут в фільтрах (attributeIndex);

– Selection tick boxes – дозволяє вибрати атрибути для подальших операцій (як правило для видалення).

– Name – ім'я атрибуту, що визначено у файлі даних.

При виборі одного з атрибутів, його ім’я підсвічується в списку синім кольором. При цьому змінюється вміст правої панелі «Selected attribute»:

– Name – ім'я атрибуту;

– Type – тип атрибуту, наприклад, Nominal або Numeric.

– Missing – кількість і відсоток примірників, для яких значення цього атрибуту втрачено (не визначено).

– Distinct – кількість різних значень атрибуту;

– Unique – кількість і відсоток примірників, для яких значення атрибуту має унікальне значення (такого значення немає у жодного іншого екземпляра).

Якщо атрибут має тип nominal (номінальний або категоріальний, значення з певної множини), список містить усі можливі значення атрибуту з кількістю їх появи. Якщо атрибут відноситься до numeric (числовий), то список надає наступну статистичну інформацію: мінімум, максимум, математичне очікування і стандартне відхилення.

Нижче представлена кольорова гістограма значень атрибуту, яка залежно від атрибуту, обраного в якості цільового (що визначає клас), показує розподіл значень атрибуту. Можна подивитися гістограми всіх атрибутів натисканням кнопки «Visualize all».

Вище списку всіх атрибутів знаходяться кнопки, що дозволяють вибирати атрибути (за допомогою чекбоксів). Обрані таким чином

атрибути можна видалити кнопкою «Remove».

Вкладка попередньої обробки даних дозволяє відфільтрувати і трансформувати дані в необхідний формат.

Зліва на панелі «Filter» знаходиться кнопка вибору фільтра «Choose». За її допомогою можна вибрати один з попередньо визначених фільтрів. Після вибору фільтра з ієрархічного списку його назва з'явиться праворуч від кнопки «Choose». Клацнувши по назві фільтра, викликається діалогове вікно налаштування параметрів фільтра (The GenericObjectEditor Dialog Box). Це вікно надає інформацію про призначення фільтру та дозволяє налаштувати параметри фільтру. Всі поля даного вікна мають спливаючі підказки.

Таке ж діалогове вікно використовується при налаштуванні інших об'єктів програми WEKA (наприклад, класифікаторів).

Внизу вікна налаштувань знаходяться 4 кнопки. Перші дві, «Open ...» і «Save...» дозволяють зберегти налаштування об'єкта для майбутнього використання. Кнопка «Cancel» дозволяє повернутися без збереження проведених змін. По натисканню кнопки «OK» зміни зберігаються, і користувач повертається в «Explorer window».

По натисканню на кнопку «Apply» у правій частині панелі «Filter» відбувається фільтрація (попередня обробка) вихідних даних відповідно до обраного алгоритму. Кнопка «Undo» призначена для скасування проведених над даними операцій. Всі зміні, що вносяться в дані в програмі, виконуються в оперативній пам’яті та не впливають на початковий файл. Для збереження змінених даних в новий файлі призначена кнопка «Save».

Зауваження! Деякі фільтри під час своєї роботи приймають в якості параметру атрибут класу, що задається за допомогою випадаючого списку над гістограмою. Так, наприклад, «supervised filters» вимагають, щоб клас був заданий, в той час, коли «unsupervised attribute filters» не приймають даний параметр до уваги.

1.2.5 Класифікація (Classify)

Вкладка «Classify» дозволяє вирішувати задачу класифікації для завантаженого набору даних (рис. 1.3).

Вгорі вкладки знаходиться панель вибору класифікатора «Classifier». Вибір і налаштування параметрів класифікатора подібна вибору фільтра попередньої обробки даних.

На панелі «Test options» визначається метод тестування

навченого класифікатора:

– на навчальній вибірці (use training set);
– на тестовій вибірці з окремого файлу (supplied test set);
– по блоках (cross-validation);
– за допомогою відсоткового розподілу початкової вибірки на навчання та контроль (percentage split).

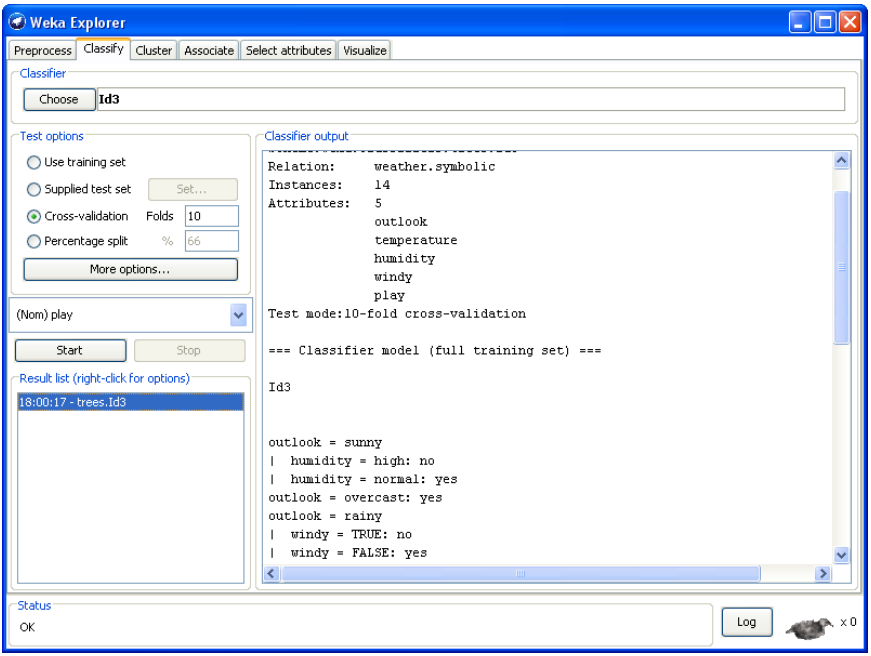


Рисунок 1.3 – Вкладка рішення задачі класифікації

При виборі деяких опцій доведеться вказати параметри тестування. Наприклад, при виборі «cross-validation» треба вказати, на скільки блоків (фолдів, складок) розбивати вибірку. Наприклад, якщо вказати 10 фолдів, то вибірка буде розділена на 10 рівних частин, потім 9 частин буде використано для навчання і 1 для тестування. Процес буде повторений 10 разів, а результати усереднені.

Leave-one-out – це спеціальний випадок крос-перевірки, при якому кількість фолдів дорівнює кількості примірників у вибірці. Таким чином, кожен раз тільки один примірник виступає в якості тестової вибірки.

Кнопка «More options», дозволяє вибрати вид звіту про навчання класифікатора. Розглянемо параметри, які можна задати:

– Output model – відображення побудованої моделі;

– Output per-class stats – відображення статистики точність/ефективність та істина/неправда для кожного класу;

– Output entropy evaluation measures – відображення оцінки ентропії на критеріях;

– Output confusion matrix – відображення матриці неточностей передбачення класифікатора;

– Store predictions for visualization – передбачення класифікатора згруповані таким чином, щоб їх можна було візуалізувати;

– Output predictions – відображення передбачених значень класів для тестової вибірки;

– Output additional attributes – використовується у разі якщо разом з передбаченими значеннями класів необхідно вивести додаткові атрибути об’єктів (наприклад, якщо необхідно вивести ID атрибуту для визначення неправильно класифікованих об'єктів);

– Cost-sensitive evaluation – помилка класифікації визначається з урахуванням матриці цінності;

– Random seed for xval/% Split – визначає випадковий сід (seed), який використовується при рандомізації (перетасуванні) даних перед їх поділом;

– Preserve order for % Split – скасовує рандомізацію даних перед їх поділом на навчальну і тестову вибірку.

– Output source code – вивід побудованої моделі у вигляді вихідного коду мовою Java, якщо це можливо.

У WEKA класифікатори спроектовані на прогнозування єдиного «класу», який є цільовим атрибутом в задачі класифікації. Деякі класифікатори можуть бути навчені для передбачення категоріальних (nominal) класів (класифікація), інші можуть бути навчені тільки для передбачення числових класів (задача регресії), а є класифікатори, що вирішують обидва типи задач. За замовчуванням у якості цільового обирається останій атрибут у списку. При необхідності цільовий атрибут можна змінити на панелі «Test options».

Процес навчання класифікатора починається натисканням на кнопку «Start». Поки класифікатор зайнятий навчанням, маленька

пташка в рядку статусу рухається. Процес навчання можна зупинити в будь-який момент натисканням кнопки «Stop».

Коли процес навчання закінчений, праворуч на панелі «Classifier output» з'являється інформація з результатами навчання і тестування, а також заповнюється панель «Result list».

Текст на панелі «Classifier output» містить інформацію, розбиту на декілька секцій. Розглянему кожну з них.

Секція «Run information» містить параметри схеми навчання, ім'я відношення, екземляри, атрибути і параметри тестування.

Секція «Classifier model (full training set)» містить текстове представлення моделі класифікації, отриманої в ході навчання.

Результати обраної стратегії навчання розбиті на наступні секції:

– Summary – статистична інформація, що підводить підсумок точності класифікації для перевірочних примірників.

– Detailed Accuracy By Class – більш детальна інформація по кожному класу.

– Confusion Matrix – матриця невідповідності моделі, елементи якої показують кількість тестових екземплярів, чий істинний клас є рядком і чий передбачений клас є колонкою.

Після навчання декількох класифікаторів панель «Result List» міститиме кілька записів. Клік лівою кнопкою миші на записі дозволяє переміщатися між результатами аналізу. Праве клацання на записі викликає контекстне меню наступного змісту:

– View in main window – показати результати в головному вікні (рівноцінно кліку лівою кнопкою миші);

– View in separate window – відкриває незалежне вікно з результатами;

– Save result buffer – виводить діалогове вікно, що дозволяє зберегти результати в текстовому файлі;

– Load model – завантажує попередньо навчену модель з бінарного файлу;

– Save model – зберігає навчену модель в бінарний файл у вигляді серіалізірованого java об'єкта;

– Re-evaluate model on current test set – тестує ефективність навченої моделі на наборі даних, завантаженому за допомогою пункту «Set» на панелі «Test option»;

– Visualize classifier errors – виводить вікно з графіком, що візуалізує результати класифікації: вірно класифіковані примірники позначаються хрестиками, а невірно класифіковані – квадратиками;

– Visualize tree чи Visualize graph – надає графічне представлення структури класифікаційної моделі, якщо таке можливо (наприклад, для дерев рішень і байєсівських мереж). Графічне подання можливе тільки для байєсівських мереж. У графі для дерев рішень, клік правою кнопкою миші на порожній панелі виводить контекстне меню. Клацнувши на вузлі дерева можна отримати навчальні примірники;

– Visualize margin curve – генерує графік, який ілюструє границі передбачення;

– Visualize threshold curve – генерує графік, який ілюструє різну ефективність передбачення, отриману шляхом варіювання порогового значення параметра класифікації;

– Visualize cost curve – генерує графік, який дає явне уявлення про очікувані витрати.

Опції неактивні, якщо вони зараз не можуть бути застосовані.

1.2.6 Кластеризація (Cluster)

Вкладка «Cluster» містить схему навчання кластерізаторов (рис. 1.4). Вибір функції кластеризації та налаштування її параметрів відбуваються таким же чином, як і для розглянутої вище задачі класифікації.

Панель «Cluster mode» використовується для того, щоб визначити, що кластеризувати і як оцінювати результати. Перші три опції такі ж, як і для задачі класифікації: Use training set, Supplied test set і Percentage split (при цьому дані використовуються для віднесення до кластеру, а не для передбачення певного класу).

Четвертий метод «Classes to clusters evaluation» оцінює наскільки добре був обраний кластер, порівнюючи його з попередньо заданим класом в даних.

Додаткова опція у вигляді чекбокса «Store clusters for visualization» визначає, чи можливо буде візуалізувати кластери по закінченні навчання. При вирішенні завдань з дуже великими обсягами даних слід відключити дану опцію, щоб уникнути проблем з нестачею пам'яті.

Кнопка «Ignoring Attributes» дозволяє визначити, які атрибути слід ігнорувати при проведенні кластеризації.

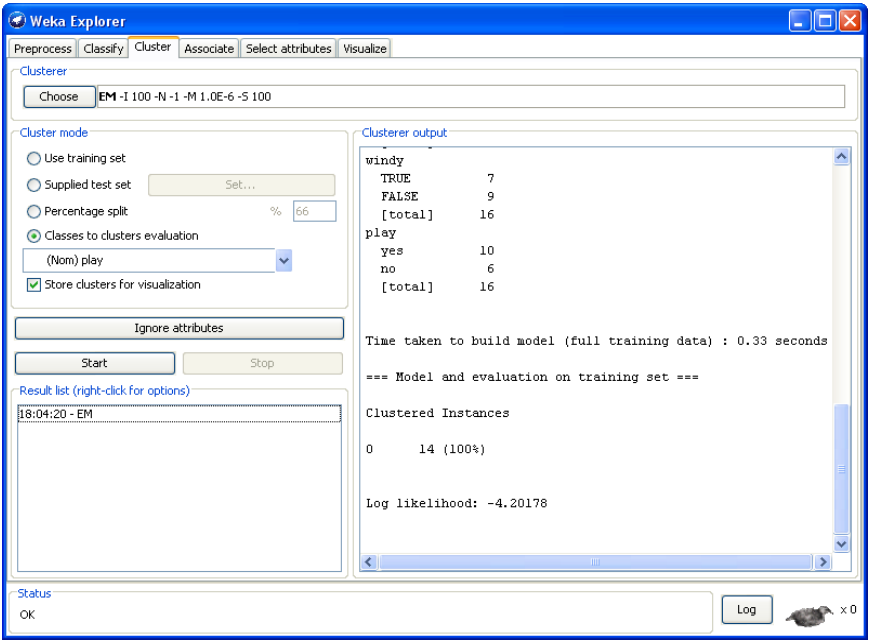


Рисунок 1.4 – Вкладка рішення задачі кластеризації

Вкладка Кластеризації також як і вкладка класифікації містить кнопки «Start / Stop», панель результатів і список результатів. Клацання правою кнопкою миші на запису у списку результатів дає контекстне меню з двома опціями візуалізації:Visualize cluster assignments и Visualize tree.

1.2.7 Асоціативні правила (Associate)

Вкладка «Associate» містить схему навчання асоціативних правил (рис. 1.5). Алгоритми пошуку асоціативних правил обираються, налаштовуються і виконуються так, як описано вище.

1.2.8 Відбір атрибутів (Selecting attributes)

Відбір атрибутів включає перебір всіх можливий комбінацій атрибутів даних для пошуку підмножини атрибутів, що дають

найкращий результат передбачення (рис. 1.6).

Для цього повинні бути налаштовані два об'єкти: оцінювач атрибутів і метод пошуку. Оцінювач (Atribute Evaluator) визначає який метод використовується для призначення значущості кожної підмножини атрибутів, а метод пошуку (Search Method) визначає стиль пошуку підмножин.

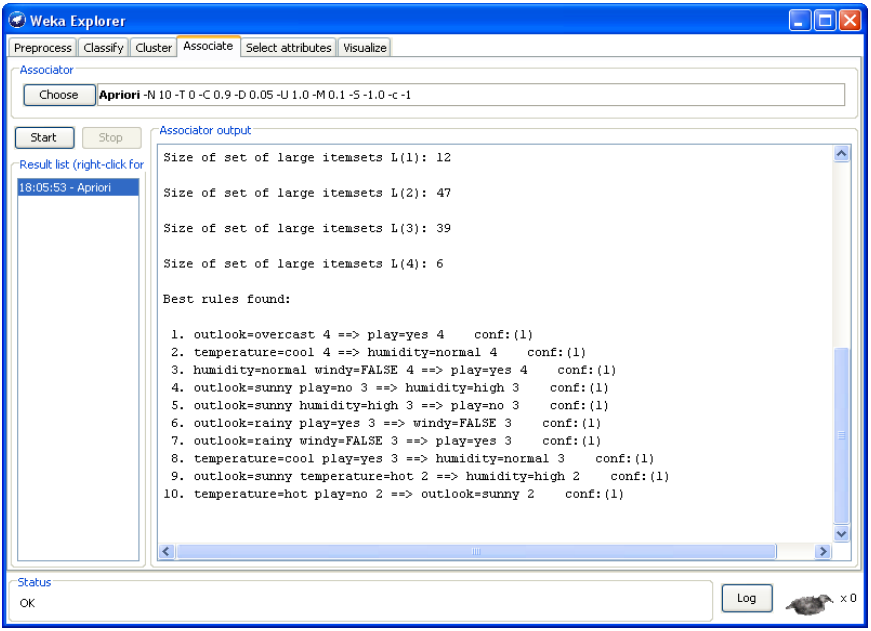


Рисунок 1.5 – Вкладка для пошуку асоціативних правил

Панель «Attribute Selection Mode» має два параметри:

– Use full training set – значимість підмножини атрибутів визначається для повного набору навчальних даних;

– Cross-validation (ковзний контроль, крос-перевірка) – значимість підмножини атрибутів визначається за допомогою крос-перевірки. Поля Fold і Seed визначають кількість блоків (folds) і випадковий сид (seed), використовуваний при перетасування даних.

Знизу знаходиться список, що випадає, який задає цільовий атрибут, яка буде використовуватися в якості класу.

По натисканню на кнопку «Start» запускається процес відбору атрибутів. Коли процес закінчено, результати виводяться на панель

результатів «Attribute selection output» та додаються до списку результатів. Натискання правої кнопки миші на результати видає контекстне меню. Перші три пункти цього меню (View in main window, View in separate window and Save result buffer) такі ж як і для вкладку класифікації. Додатковими є Visualize reduced data, або якщо був обраний метод Principal-Components, то Visualize transformed data.

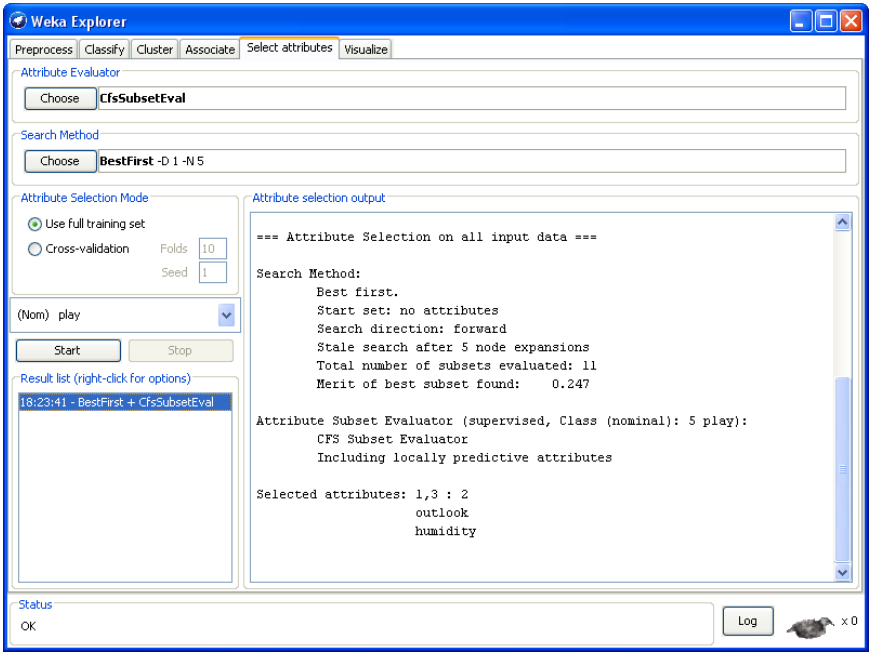


Рисунок 1.6 – Вкладка вирішення задачі відбору атрибутів

1.2.9 Візуалізація (Visualizing)

Вкладка візуалізації дозволяє представити вихідні дані в графічному вигляді (рис. 1.7). При відкритті вкладки візуалізації даних відображається діаграма розкиду даних для всіх атрибутів з кольоровим кодуванням згідно обраного класу. Розміри кожної з діаграм можуть бути змінені, можуть бути змінені розміри точок. У дані можна додати шум (jitter) для виявлення слабких точок. Кожен графік також можна відкрити в окремому вікні натисканням на нього.

Для застосування внесених змін та оновлення інформації на графіках необхідно натиснути кнопку «Update».

На окремому графіку можна вибрати окремі точки за допомогою списку, що випадає «Select Instance». На графіку можна залишити тільки обрані точки, їх же можна зберегти в окремий файл.

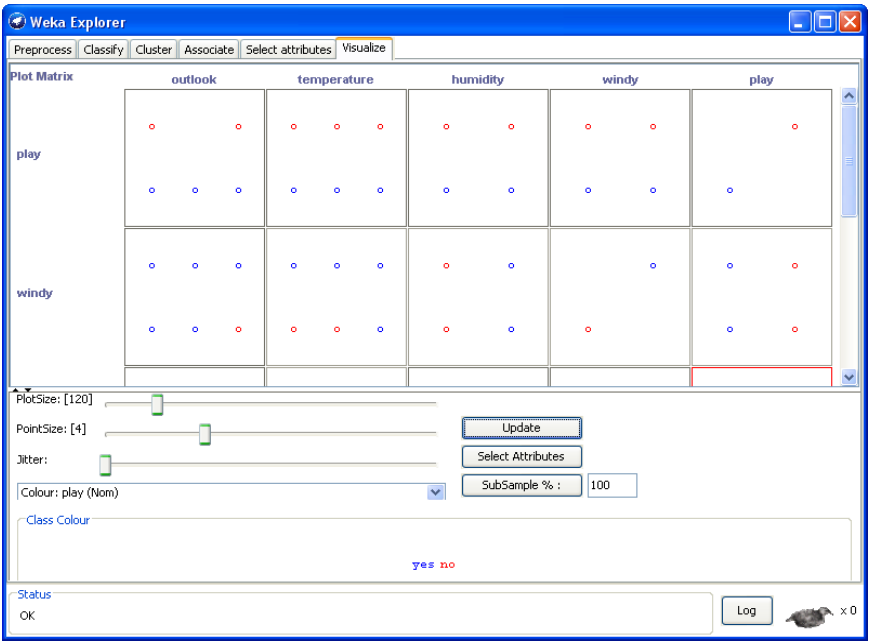


Рисунок 1.7 – Вкладка візуалізації вихідних даних

1.2.10 Модуль Experimenter

Цей модуль дозволяє проводити експерименти: запускати кілька алгоритмів на декількох задачах і отримувати зведений звіт.

Вкладка «Setup» (рис. 1.8). На вкладці спочатку активні дві кнопки: «Open» (відкрити файл експерименту) і «New» (створити новий експеримент), а також дві опції списку параметрів експерименту: «Simple» (простий), «Advanced» (складний).

Створимо новий експеримент. При натисканні на кнопку "New" активуються всі опції. На панелі «Result Destination» необхідно обрати файл для запису звіту. За замовчуванням файл результатів має розширення arff, тому не затріть файли вихідних даних.

На панелі «Experiment Type» вибирається тип експерименту:

– Cross-validation (default) проводити верифікацію методом

контролю по блоках/фолдах;

– Train/Test Percentage Split (data randomized) відсотковий поділ вибірки на контроль/навчання з випадковим порядком проходження примірників;

– Train/Test Percentage Split (order preserved) відсотковий поділ вибірки із збереженням порядку; застосовується в тому випадку, коли в одному файлі зберігається і навчальна і тестова вибірка.

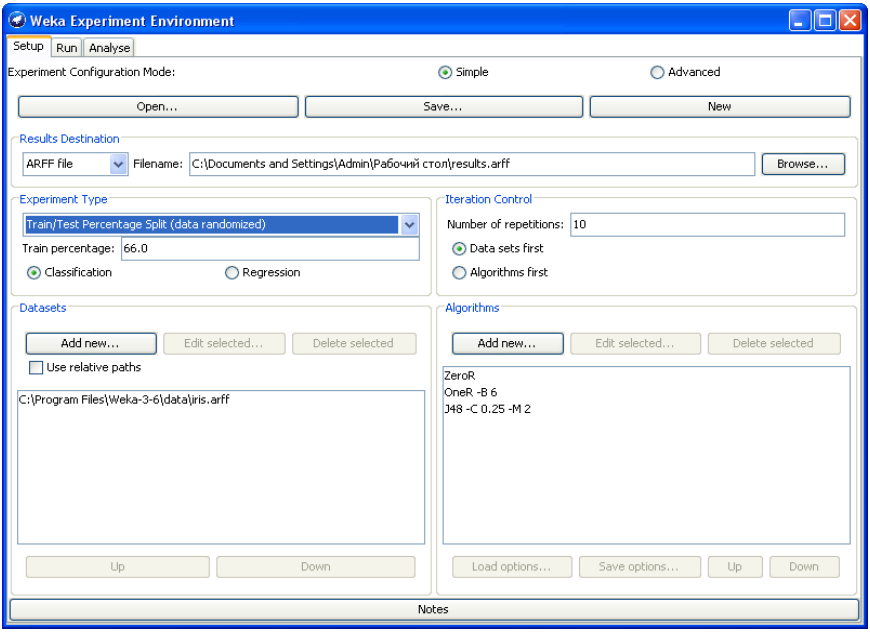


Рисунок 1.8 – Установка параметрів експерименту

На панелі «Iteration Control» обирається число ітерацій (повторень експериментів). При повтореннях відбуваються нові розбиття на навчання/контроль і на фолди. Також тут вказується порядок проведення експерименту (перебирати спочатку всі задачі або всі алгоритми).

На панелі «Datasets» можна обрати набори даних, на панелі «Algorithms» – алгоритми (параметри їх вибираються аналогічно тому, як це робилося в модулі «Explorer»).

Вкладка «Run». Єдина кнопка «Start» запускає експеримент на виконання.

Вкладка «Analyze» (рис. 1.9) дозволяє аналізувати результати експериментів. На панелі «Source» обирається, який експеримент аналізувати (тільки що проведений чи записаний у файлі).

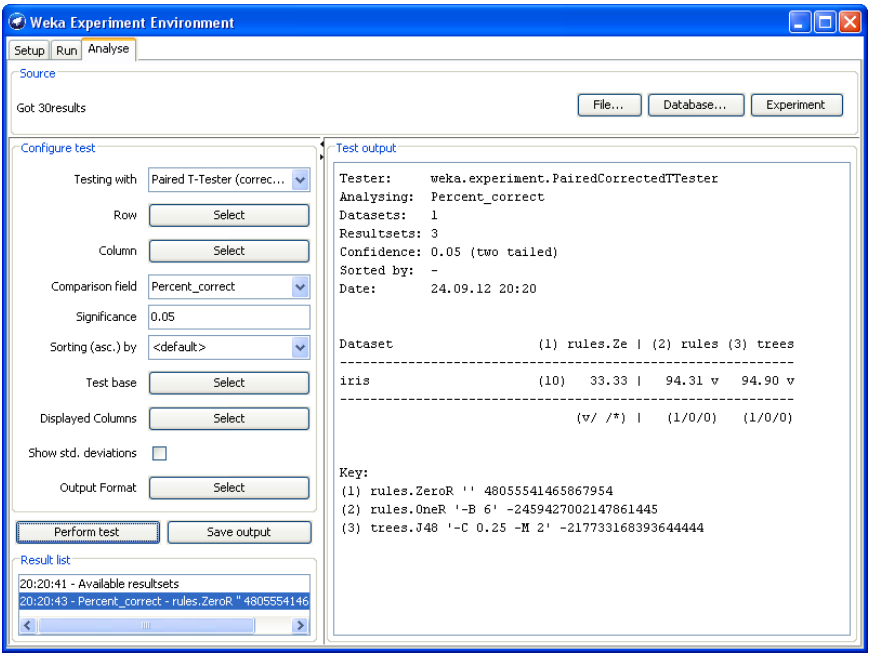


Рисунок 1.9 – Оцінка результатів експерименту

На панелі «Configure test» визначається вид статистики для аналізу. Кнопка «Row» дозволяє вибрати дані, що будуть записані по рядках виведеної матриці (для прикладу виберемо з випадаючого списку Dataset), а кнопка Column – що буде записано за стовпцями (виберемо Scheme). У полі Comparison field вибирається тип даних, якими буде заповнена таблиця (вибір Percent_correct забезпечує заповнення відсотками вірної класифікації).

Порівняння результатів роботи алгоритмів виконується за допомогою критерію Стьюдента (t-test, Student's tt est).

Для генерації звіту натисніть кнопку «Perform test».

На рис.1.9. показана статистика роботи трьох алгоритмів на задачі «iris». У колонках статистики вказані методи класифікації, а в рядках – набори даних (задачі).

Значення «(10)» на початку рядка, що відповідає набору даних «iris», визначає кількість запусків експерименту (ітерацій, навчання і тестування).

Під час експериментів вибірка розбивалася у співвідношенні 66% примірників для навчання і 34% – для тестування.

Відсоток вірно класифікованих примірників для трьох методів показаний у рядку, що відповідає набору даних iris: 33,33% для ZeroR, 94,31% для OneR і 94,90% для J48. Відмітка «v» або «*» означає, що даний результат статистично краще (v) або гірше (*), ніж алгоритм обраний в якості базового (baseline scheme, в даному випадку це ZeroR) при зазначеному рівні значущості (significance, в даному випадку 0,05). Результати двох алгоритмів OneR і J48 статистично краще, ніж результати базового алгоритму ZeroR.

Внизу кожної колонки знаходиться значення (xx / yy / zz), яке показує кількість разів (за кількістю рядків) у яких результати роботи алгоритму, що знаходиться у назві стовпця, були краще (xx), такі ж (yy) або гірше (zz), ніж результати базового алгоритму на наборах даних, що використовувалися в експерименті.

У даному прикладі був тільки один набір даних і OneR один раз був краще ніж ZeroR і ніколи гірше або еквівалентний йому (1/0/0); J48 також один раз був кращим, ніж ZeroR.

Вибираючи «Number_correct» в якості значення для порівняння (comparison field), в статистиці отримуємо середню кількість вірно розпізнаних примірників для тестової вибірки (з 50 тестових примірників, що є 33% від загальної кількості в 150 екземплярів для набору даних «iris»).

Базовий алгоритм для порівняння може бути змінений за допомогою кнопки «test base».

Вибравши опцію «Summary» в якості бази для порівняння, отримаємо наступну інформацію.

```
a b c (No. of datasets where [col] >> [row])
 - 1 (1) 1 (1) | a = (1) rules.ZeroR
0 (0) - 1 (0) | b = (2) rules.OneR
0 (0) 0 (0) - | c = (3) trees.J48
```

У даному експерименті перший рядок (- 1 1) показує, що метод у колонці b (OneR) краще, ніж метод в рядку a (ZeroR) і метод у колонці c (J48) також краще методу в рядку a. У дужках вказано кількість статистично значущих перемог колонки по відношенню до

рядка. Значення 0 показує, що відповідний стовпець не був статистично кращим за рядок.

Вибравши опцію «Ranking» в якості бази для порівняння, отримаємо наступну інформацію.

```
>-< > < Resultset
 1 1 0 trees.J48
 1 1 0 rules.OneR
 -2 0 2 rules.ZeroR
```

Даний тест ранжує методи за загальною кількістю значущих перемог (>) і поразок (<) проти інших методів. Перша колонка (> - <) показує різницю між кількість перемог і поразок.

1.2.11 Модуль Knowledge flow

Інтерфейс KnowledgeFlow відображає концепцію потоків даних (рис. 1.10).

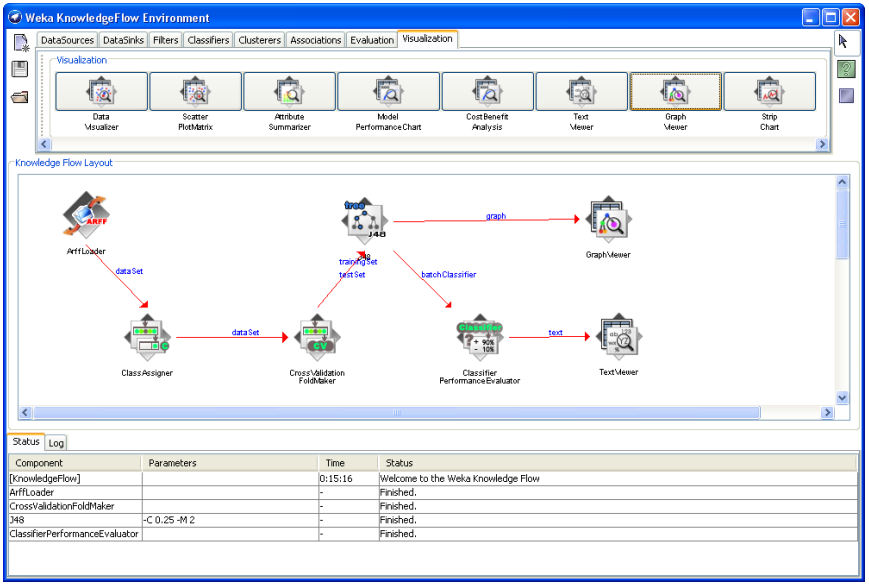


Рисунок 1.10 – Модуль Knowledge flow

Спочатку з примітивів різних видів будується весь шлях, яким мають пройти дані, від джерел до виходів. Зазвичай він такий: джерело даних (DataSources) – фільтр (Filter) (необов'язковий) – розбиття на навчальну і тестову множину (Evaluation) – класифікатор

(Classifiers) чи кластерізатор (Clusterers) чи асоціатор (Associators)) – оцінка (Evaluation) – виведення даних (DataSinks). Практично до будь-якого етапу можна приєднати відображення (Visualization).

Примітиви перетягуються в робоче поле з відповідних вкладок (клік на примітив – клік на робоче поле). Входи і виходи примітивів з'єднуються зв'язками, причому не можна з'єднати різнотипні за даними виходи і входи. У примітиву може бути кілька виходів з різними даними. Зв'язки теж бувають різні, залежно від того, які дані по них передаються (з якого виходу вони йдуть). Зв'язуються примітиви шляхом вибору в контекстному меню, що з'являється при натисканні правою кнопкою миші на примітиві в робочому полі, у підпункті меню Connection певного виходу примітиву і наступного кліку на примітив, з яким слід встановити зв'язок. Всі примітиви, доступні для зв'язку, при цьому виділяються.

Після прокладки шляху за допомогою джерела даних завантажують дані, і вони автоматично проходять побудований шлях.

Розглянемо приклад побудови потоку даних на рис. 1.10.

2. Відкрийте вкладку джерел даних «DataSources», оберіть завантажувач даних «ArffLoader» і розмістіть його на робочому полі.

3. Виберіть arff файл для завантаження. Для цього клікніть правою кнопкою миші на завантажувачі, розміщеному на полі, і виберіть зі списку в секції «Edit» пункт «Configure».

4. Відкрийте вкладку «Evaluation», виберіть компонент «ClassAssigner» (дозволяє вибрати, який з атрибутів є цільовим) і розмістіть його на полі.

5. Тепер необхідно з’єднати компоненти «ArffLoader» та «ClassAssigner». Для цього клікніть правою кнопкою миші на завантажувачі «ArffLoader» і виберіть з випадаючого списку в секції «Connections» пункт «dataSet». З’явиться синя лінія, яку необхідно підвести до іконки компоненту «ClassAssigner» на полі та клікнути на ньому лівою кнопкою миші. З’явиться красна лінія потоку даних з підписом «dataSet», що з’єднує два компоненти.

6. Викличте налаштування компонента «ClassAssigner» і виберіть цільовий атрибут (клас).

7. З вкладки «Evaluation» виберіть компонент «CrossValidationFoldMaker» і розмістіть його на робочому полі. З'єднайте «ClassAssigner» і «CrossValidationFoldMaker» зв'язком «dataSet».

8. З вкладки «Classifiers» в секції «Trees» виберіть компонент «J48» і розмістіть його на робочому полі. З'єднайте «CrossValidationFoldMaker» і «J48» двома зв'язками: «trainingSet» і «testSet».

9. З вкладки «Evaluation» виберіть компонент «ClassifierPerformanceEvaluator» і розмістіть його на робочому полі. З'єднайте «J48» з цим компонентом зв'язком «batchClassifier».

10. З вкладки «Visualization» виберіть компонент «TextViewer» і розмістіть його на робочому полі. З'єднайте «ClassifierPerformanceEvaluator» з цим компонентом зв'язком «Text».

11. Запустіть обробку потоку даних за допомогою пункту «Start loading» у випадаючому списку завантажувача «ArffLoader». Залежно від розміру даних і часу виконання крос-перевірки буде відображена анімація деяких з компонентів в робочій області. Крім того, з'явиться інформація в нижній частині вікна «Status»/«Log».

12. По закінченні обробки даних перегляньте результати, викликавши пункт «Show results» компоненту «TextViewer».

13. Крім того, можна підключити компонент «TextViewer» та / або «GraphViewer» до «J48» для того, щоб побачити представлення дерев, побудованих на кожному з фолдів крос-перевірки. Така можливість не передбачена в модулі Explorer.

1.3 Завдання на лабораторну роботу

Частина А.

1. Як завантажити програму Weka?
2. Яке призначення модулів Explorer, Knowledge Flow, Experimenter, Command-Line Interface?
3. Перелічіть основні джерела даних в Weka, опишіть формат arff файлу.
4. Відкрийте модуль Explorer, завантажте набір даних 'weather.arff' (описує дані для прийняття рішення про проведення спортивного змагання при заданих погодних умовах) або 'iris.arff' (передбачення класу квітки ірису на підставі довжини і ширини його чашолистка і пелюстки) і дайте відповіді на питання:
   - скільки примірників у вибірці;
   - назвіть імена атрибутів, їх типи і значення;
   - вкажіть цільовий атрибут, тобто атрибут класу;

- опишіть гістограму внизу праворуч;
- скільки примірників кожного класу у вибірці;
- що відбудеться після натискання кнопки Visualize All;
- як переглянути всі примірники у наборі даних.

5. Опишіть призначення вкладок в Explorer Panel: Preprocess panel (попередня обробка даних), Classify (класифікація), Cluster (кластеризація), Associate (асоціативні правила), Select Attributes (відбір атрибутів), Visualize (візуалізація)?

6. Відкрийте вкладку попередньої обробки даних і дайте відповіді на питання:

- які основні панелі на вкладці попередньої обробки;
- що мається на увазі під фільтрацією в Weka;
- які два типи фільтрів у Weka і в чому різниця між ними;
- у чому відмінністю між фільтрами атрибутів і фільтрами примірників.

7. По черзі завантажте набори даних 'weather.arff' й 'weather.nominal.arff'. а допомогою unsupervised фільтру RemoveWithValues видаліть всі екземпляри, у яких атрибут ‘humidity’ має значення ‘high’ для категоріальних значень і більше математичного очікування для числових.

8. Завантажте набір даних 'iris.arff' і виберіть вкладку візуалізації даних. Вкажіть мету візуального представлення даних. Виберіть один з графіків і проекспериментуйте з кнопками.

9. Побудуйте потік даних в модулі Knowledge flow як описано в п.1.2.11.

Частина Б. Індивідуальне завдання.

10. В додатку А оберіть індивідуальне завдання. Використовуючи вкладки попередньої обробки даних та візуалізації, проведіть детальний опис вибірки даних.

1.4 Зміст звіту

1. Тема і мета роботи.
2. Завдання до роботи.
3. Результати виконання завдань розділу 1.3.
4. Висновки, що відображують результати виконання роботи та їх критичний аналіз.

ЛАБОРАТОРНА РОБОТА № 2
ЗАДАЧА КЛАСИФІКАЦІЇ

2.1 Мета роботи

На практиці вивчити роботу алгоритмів класифікації, навчитися інтерпретувати результати роботи класифікаторів і вибирати найкращий метод для вирішення поставленої задачі.

2.2 Основні теоретичні відомості

У лабораторній роботі розглядаються наступні методи класифікації (у дужках наведено назву в програмі WEKA, при цьому перше слово перед крапкою означає тип алгоритму класифікації і також вказує назву папки, в якій знаходиться метод):

– 0R чи Zero Rule класифікація, тобто прогнозування середнього значення для числового класу та моди для категоріального (rules.ZeroR);

– класифікація за одним правилом 1R чи One Rule (rules.OneR);

– покриваючий метод PRISM (rules.Prism);

– наївна Байєсова класифікація (bayes.NaiveBayes);

– методи побудови дерев рішень ID3 (trees.Id3) та C4.5 (trees.J48);

– метод опорних векторів (functions.SMO);

– метод k найближчих сусідів (lazy.IBk).

2.2.1 Параметри налаштування алгоритмів класифікації

Розглянемо параметри налаштування алгоритмів, що використовуються в роботі (табл. 2.1).

При виконанні завдань до кожної з лабораторних робіт необхідно дослідити вплив параметрів налаштування на результати роботи алгоритмів.

Додаткову інформацію про алгоритми, їх параметри і вимоги до оброблюваних даних можна отримати у вікні налаштувань алгоритмів на панелі «About» в програмі WEKA і в джерелах [2-4].

Для алгоритмів Prism, Id3, ZeroR параметрів, що настроюються, немає.

Таблиця 2.1 – Параметри налаштування класифікаторів

Метод	**Параметр**
OneR	MinBucketSize – використовується для дискретизації числових атрибутів
NaiveBayes	displayModelInOldFormat – відображення побудованої моделі у старому форматі, що підходить, коли атрибут класу приймає багато значень. Новий формат краще у випадку, коли менше класів і багато атрибутів. useKernelEstimator – для оцінки числових атрибутів використовувати оціночну функцію відмінну від нормального розподілу. useSupervisedDiscretization – використовувати дискретизацію з учителем для перетворення числових атрибутів у номінальні.
J48	binarySplits – використовувати бінарний поділ на категоріальних атрибутах для побудови дерев. confidenceFactor – довірчий рівень, використовується для відсікання гілок (малі значення - сильніше відсікання). minNumObj – мінімальна кількість примірників у листі. reducedErrorPruning – використовувати відсікання з приведеною похибкою або алгоритм відсікання C.4.5 (заснований на ймовірностях). Вибірка розбивається на дві частини: для побудови дерева і для перевірки (відсікання). saveInstanceData – чи зберігати навчальну інформацію для візуалізації. subtreeRaising – чи використовувати операцію підняття піддерев при відсіканні гілок. unpruned – чи залишити дерево повним. useLaplace – використовувати оціночну функцію Лапласа для підрахунку ймовірностей в листках.
SMO	buildLogisticModels – чи застосовувати логістичні моделі до виходів (для належної оцінки ймовірностей). c – параметр складності C. checksTurnedOff – вимкнути витратні за часом перевірки (використовувати з обережністю). kernel – функція ядра.

Продовження табл.2.1.

Метод	**Параметр**
SMO	buildLogisticModels –застосовувати логістичні моделі до виходів (для належної оцінки ймовірностей). c – параметр складності. checksTurnedOff – вимкнути витратні за часом перевірки (використовувати з обережністю). kernel – функція ядра. epsilon – параметр точності (не змінювати). filterType – чи буде змінена початкова інформація і яким чином (нормалізація або стандартизація). toleranceParameter – допустиме відхилення (не змінювати).
IBk	KNN – кількість сусідів. crossValidate – чи буде використовуватися для вибору оптимальної кількості сусідів крос-перевірка hold-one-out. distanceWeighting – метод вибору вагових коефіцієнтів для відстаней. meanSquared – чи використовується середньоквадратична помилка, чи середня абсолютна помилка для крос-перевірки під час вирішення завдання регресії. nearestNeighbourSearchAlgorithm – алгоритм пошуку найближчих сусідів. windowSize – максимальна кількість примірників, дозволених в навчальному пулі. Додавання додаткових примірників понад цього значення призведе до видалення старих екземплярів. Значення 0 означає відсутність межі.

2.2.2 Методи оцінки помилок класифікації

Розглянемо параметри оцінки побудованої моделі класифікації.

Матриця невідповідності - це матриця розміру L × L, де L - число класів, ij-й елемент матриці (i - рядок, j - стовпець) дорівнює числу об'єктів з i-го класу, які були віднесені до j-го. Число вірно класифікованих об'єктів дорівнює сумі елементів, що стоять на головній діагоналі.

Результати добре навченого класифікатора покажуть матрицю невідповідності, в якій найбільші значення стоять на діагоналі матриці, а невеликі значення (в ідеалі нулі) - на інших позиціях.

Розглянемо матрицю невідповідності для двох класів (табл. 2.2).

Таблиця 2.2 – Матриця невідповідності для двох класів

		Передбачений клас		
		Так	**Ні**	
Реальний клас	**Так**	істинно позитивні (TP)	хибно негативні (FN)	P=TP+FN
	Ні	хибно позитивні (FP)	істинно негативні (TN)	N=FP+TN
		P'=TP+FP	N'=FN+TN	

На діагоналі матриці знаходяться істинно позитивні (true positive, TP) і істинно негативні (true negative, TN) екземпляри. Примірники, які відносяться до класу «так», але були віднесені класифікатором до класу «ні» називаються хибно негативними (false negative, FN). Примірники, які відносяться до класу «ні», але були віднесені класифікатором до класу «так» називаються хибно позитивними (false positive, FP).

Для випадків класифікації, в яких кількість класів більше двох, для розрахунків приймається, що клас, що розглядається, є класом «так», а всі інші класи об'єднуються та утворюють клас «ні».

Розглянемо вирази для розрахунку параметрів точності класифікації для кожного з класів.

Параметр «TP rate» (чутливість, sensitivity) або «recall» (ефективність) розраховується наступним чином:

$$TP\ rate = recall = \frac{TP}{TP + FN} = \frac{TP}{P}.$$

Для класу, що розглядається, значення цього параметру дорівнює відсотку вірно класифікованих об'єктів класу (отримується діленням діагонального елементу матриці невідповідності на суму елементів в його рядку). Іншими словами, параметр чутливості показує долю позитивних екземплярів, які були вірно розпізнані.

Параметр «FP rate» розраховується за формулою:

$$FP\ rate = \frac{FP}{FP + TN} = \frac{FP}{N}.$$

Його значення дорівнює відсотку об’єктів інших класів, які помилково були занесені в клас, що розглядається (якщо з матриці викреслити рядок класу, що розглядається, то значення дорівнюватиме сумі елементів стовпця цього класу, поділеній на суму всіх елементів).

Параметр «TN rate» (специфічність, specificity) дорівнює:

$$TN\ rate = \frac{TN}{TN + FP} = \frac{TN}{N}$$

та показує частину негативних екземплярів, які були вірно розпізнані.

Параметр «precision» (точність) розраховується як:

$$precision = \frac{TP}{TP + FP} = \frac{TP}{P'}.$$

Його значення дорівнює відсотку вірно класифікованих об’єктів із всіх об’єктів, віднесених алгоритмом до класу, що розглядається (відношення діагонального елемента до суми елементів стовпця).

Параметр «F-measure» - це середнє гармонійне Precision і Recall:

$$F - measure = 2 * \frac{precision * recall}{precision + recall}.$$

Параметри success rate (accuracy, recognition rate – частка успішних спроб, точність, коефіцієнт розпізнавання) та error rate (misclassification – частка помилок, помилкова класифікація):

$$success\,rate = \frac{TP + TN}{TP + TN + FP + FN},$$

$$error\ rate = 1 - success\,rate = \frac{FP + FN}{TP + TN + FP + FN}.$$

Також буває важливо оцінити втрати і виграші (costs and benefits), пов'язані з класифікаційної моделлю. Втрати, пов'язані з хибно позитивними передбаченнями (як наприклад, невірне передбачення, що хворий пацієнт є здоровим), набагато більші, ніж хибно негативні передбачення (здоровий пацієнт віднесений до хворих). В таких випадках ми можемо віддати перевагу одному типу

помилки над іншим шляхом призначення їм різних значень втрат, пов’язаних з перейменуванням класів.

Наприклад, розглянемо задачі видачі кредиту банком та цільової маркетингової розсилки. Втрати при видачі кредиту неплатнику набагато вище, ніж втрати від невидачі кредиту особі, яка зуміла би виплатити свій борг. Точно так само втрати від розсилки реклами сім'ям, які не відгукнуться на пропозицію можуть перевершити втрати пов'язані з відсутністю розсилки сім'ям, яких пропозиція зацікавила б.

Розглянемо тепер проблему нерівномірного розподілу класів, коли важливий для задачі клас є рідкісним в вибірці. Це означає, що розподіл набору даних відображає значну більшість негативних примірників і меншість позитивних примірників.

Наприклад, у задачі розпізнавання шахрайства цікавим для нас класом (позитивним) є клас «шахрайство», який з'являється набагато рідше, ніж негативний клас «не шахрайство». У медичній задачі до такого рідкісного класу може бути віднесений клас «злоякісна пухлина».

Припустимо, що ми навчили класифікатор класифікувати медичний набір даних, в якому цільовим атрибутом є атрибут «рак», який може приймати значення «так» і «ні».

Параметр точності success rate рівний 97% може показати, що класифікатор досить точний. Однак якщо у всій вибірці було тільки 3% примірників, що відносяться до раку? Ясно, що в такому випадку точність розпізнавання в 97% не може бути прийнятною. У цьому випадку класифікатор може правильно розпізнавати більшість негативних екземплярів (не рак) і помилково класифікувати всі позитивні примірники (рак).

Таким чином нам необхідні інші параметри, що дозволяють оцінити наскільки добре класифікатор розпізнає позитивні примірники (рак) і негативні (не рак). Для цієї задачі можуть бути використані параметри sensitivity та specificity.

Розглянемо наступний приклад навчання класифікатора.

Таблиця 2.3 – Приклад навчання класифікатора

Клас	Так	Ні	Всього	Розпізнавання (%)
Так	90	210	300	90/300=30,00 (sensitivity)
Ні	140	9560	9700	9560/9700=98,56 (specificity)
Всього	230	9770	10000	9650/10,000=96,50 (середнє)

Можна помітити, що хоча класифікатор показав загальну високу точність класифікації, його можливості правильно розпізнавати позитивний (рідкісний) клас дуже низькі (що видно з величини параметра чутливості). У той же час параметр специфічності високий, що означає, що класифікатор може точно розпізнавати негативні класи.

Параметри precision і recall також широко використовуються в класифікації. Precision може бути розглянутий як міра точності / влучності (тобто який відсоток примірників віднесених до позитивних такими і є), тоді як recall - це міра повноти (який відсоток позитивних примірників віднесені до позитивних).

Ідеальне значення параметра precision в 1,0 для класу C означає, що кожен екземпляр, який класифікатор відніс до класу C, насправді належить класу C. Однак, це нічого не говорить нам про кількість примірників клас C, які класифікатор неправильно класифікував. Ідеальне значення параметра recall в 1,0 для класу C означає, що кожен екземпляр класу C був віднесений класифікатором до класу C, але це нічого не говорить нам про те, скільки інших екземплярів були неправильно класифіковані та віднесені до класу C.

Існує тенденція зворотного взаємозв'язку між параметрами precision і recall, тобто існує можливість збільшити один параметр за рахунок зменшення іншого. Приміром, наш медичний класифікатор може досягти високого значення параметра precision відносячи всі екземпляри класу рак до класу рак, але при цьому може мати низьке значення параметра recall відносячи до класу раку також негативні екземпляри. Зазвичай ці два параметри розглядаються сумісно.

Альтернативний шлях застосування цих параметрів - це їх об'єднання в одному параметрі F-measure (F1 score або F-score).

Для оцінки співвідношення FP і FN використовуються криві, наведені в табл. 2.4.

Так, наприклад, ROC-крива (Receiver operating characteristic, операційна характеристика приймача) – це графік, що дозволяє оцінити якість бінарної класифікації, відображає залежність частки вірних позитивних класифікацій від частки помилкових позитивних класифікацій при варіюванні порогу вирішального правила. Також відома як крива помилок.

Аналіз класифікацій із застосуванням ROC-кривих називається ROC-аналізом. Кількісну інтерпретацію ROC дає показник AUC (area

under ROC curve) - площа, обмежена ROC-кривою і віссю частки помилкових позитивних класифікацій. Чим вище показник AUC, тим якісніше класифікатор, при цьому значення 0,5 демонструє непридатність обраного методу класифікації (відповідає випадковому гаданню).

Таблиця 2.4 – Криві оцінки якості класифікації

Назва	**Область застосування**	**Осі графіка**
Lift chart	Маркетинг	TP vs subset size $\frac{TP + FP}{TP + FP + TN + FN}$
ROC curve	Комунікації	TP rate vs FP rate
Recall-precision curve	Пошук інформації	Recall vs Precision

Розглянемо ще один параметр оцінки точності класифікації. У табл. 2.5 представлений приклад матриці невідповідності задачі класифікації з трьома класами.

Таблиця 2.5 – Приклад №1 матриці невідповідності

		Передбачений клас			
		A	**B**	**C**	**Total**
Реальний клас	**A**	88	10	2	100
	B	14	40	6	60
	C	18	10	12	40
	Total	120	60	20	

У цьому прикладі тестова вибірка містить 200 примірників (сума дев'яти елементів матриці). Класифікатор для тестової вибірки передбачив 120 екземплярів класу A, 60 - класу B, 20 - класу C, і при цьому 88 +40 +12 = 140 з них правильно класифіковані. Відсоток правильно класифікованих об'єктів для цього прикладу дорівнює 70%.

Що якби на цій самій вибірці працював би випадковий класифікатор, який передбачив би таку ж кількість примірників кожного класу. Розглянемо таблицю 2.6.

У першому рядку 100 екземплярів класу A розділені в такій же пропорції як (120:60:20). Точно так само розділені примірники другої і третього рядка. Загальні значення в рядках і стовпцях не змінилися, а

змінилися значення матриці. Таким чином, випадковий класифікатор дає 60 +18 +4 = 82 правильно класифікованих примірників.

Таблиця 2.6 – Приклад №2 матриці невідповідності

		Передбачений клас			
		A	**B**	**C**	**Total**
Реальний клас	**A**	60	30	10	100
	B	36	18	6	60
	C	24	12	4	40
	Total	120	60	20	

Параметр Каппа розраховує це очікуване значення (виводячи його з успішності класифікатора) і виражає результат у вигляді відношення: у чисельнику 140-82 = 58 примірників поліпшення в порівнянні з випадковим прогнозуванням, а в знаменнику – все можливе поліпшення 200-82 = 118. Для наведеного вище прикладупараметр Каппа дорівнює 49,2%. Максимальне значення параметра Каппа 100% для ідеального передбачення, а мінімальне 0 - для випадкового. Загалом, можна сказати, що статистичний параметр Каппа використовується для оцінки згоди між прогнозованою і спостережуваною категоризацією набору даних з поправкою на випадковість.

2.2.3 Критерії порівняння роботи класифікаторів

На додаток до параметрів, заснованих на точності, класифікатори можуть бути порівняні за такими наступними параметрами.

Швидкість роботи - іншими словами обчислювальні витрати, пов'язані з навчанням і застосуванням даного класифікатора.

Стійкість до помилок, робастність - можливість класифікатора робити правильні передбачення на зашумлених даних або даних з пропущеними значеннями. Даний параметр зазвичай оцінюється за допомогою синтетичних наборів даних із внесеними шумами і втраченими значеннями атрибутів.

Масштабованість - можливість будувати ефективний класифікатор на великих вибірках. Даний параметр оцінюється за допомогою наборів даних, що збільшуються.

Інтерпретованість - рівень розуміння та можливості

проникнути в суть даних, які надає класифікатор. Інтерпретованість є суб'єктивним параметром і тому його важко оцінити. Дерева рішень та класифікаційні правила можуть бути легко інтерпретовані, проте їх інтерпретованість зменшується з їх ускладненням.

2.2.4 Інтерпретація результатів класифікації в WEKA (Classifier output)

Секція «Run information» містить наступну інформацію:

– метод класифікації (scheme);

– назва набору даних, на якому проводилося навчання (relation);

– кількість примірників у вихідній вибірці (instances);

– атрибути, що характеризують об'єкти вибірки (attributes);

– відомості про тестову вибірку (test mode).

Секція «Classifier model» містить параметри налаштованого класифікатора і час, затрачений для побудови моделі. Залежно від типу класифікатора дана область буде містити різну інформацію:

– для алгоритмів, що будують правила, будуть відображені отримані правила;

– для байєсівських класифікаторів будуть перераховані розраховані ймовірності для всіх можливих комбінацій атрибут-значення-клас;

– для класифікаторів, заснованих на побудові дерев, відображається текстове представлення отриманого дерева; в дужках навпроти кожного листа вказана кількість примірників, які до нього віднесені; якщо в лист потрапляють екземпляри декількох класів, через слеш буде вказана кількість примірників, які відносяться до домішок;

– для функціональних методів виводяться значення коефіцієнтів побудованої функціональної моделі;

– для методу k найближчих сусідів відображаються налаштування класифікатора.

Секція «Predictions» буде відображена, якщо в налаштуваннях тестування класифікатора обрана опція «Output predictions». У ній для всіх примірників тестової вибірки будуть відображені результати класифікації, отримані за допомогою навченого класифікатора.

Секція оцінки побудованої моделі «Evaluation» містить кілька

підпунктів.

«Summary» містить загальну статистику роботи класифікатора:

– кількість та відсоток правильно і неправильно класифікованих примірників (Correctly and Incorrectly Classified Instances), загальна кількість примірників (Total Number of Instances);

– параметр Каппа (Kappa statistic);

– параметри ентропії (K & B Relative Info Score, K & B Information Score, Class complexity | order 0, Class complexity | scheme, Complexity improvement);

– статистичні параметри помилки класифікації (Mean absolute error, Root mean squared error, Relative absolute error, Root relative squared error).

«Detailed Accuracy By Class» містить наступні параметри точності класифікації по кожному з класів: TP Rate, FP Rate, Precision, Recall, F-Measure, ROC Area.

«Confusion Matrix» містить матрицю невідповідності.

2.3 Завдання на лабораторну роботу

Частина А.

У наступних завданнях використовувати налаштування (test options = use training set).

1. Завантажте набір даних ‘weather.nominal.arff‘ та виконайте класифікацію за допомогою алгоритму Id3:
   - вивчіть результати роботи класифікатора;
   - намалюйте отримане дерево рішень;
   - розрахуйте значення ентропії для кожного атрибута;
   - вкажіть яке відношення між значеннями ентропії атрибуту та вузлами дерева рішень;
   - поясніть кожний елемент матриці невідповідності прогнозування класифікатора;
   - опишіть статистичний параметр Каппа;
   - опишіть наступні кількісні показники: TP Rate, FP Rate, Precision та Recall.
2. Завантажте набір даних ‘weather.arff’ та запустіть алгоритм класифікації Id3. Яка виникає проблема і які шляхи її вирішення?
3. Завантажте набір даних ‘weather.arff’ в Weka та запустіть алгоритм генерації правил OneR. Запишіть отримані правила.

4. Запустіть алгоритм генерації правил PRISM на двох наборах даних ‘weather.arff’ та ‘weather.nominal.arff‘. Яка виникає проблема? З яким набором даних може працювати алгоритм? Чому? Запишіть отримані правила.

Частина Б.

У наступних завданнях використовувати налаштування (test options = cross validation), якщо не вказано іншого.

5. Завантажте набір даних ‘glass.arff’ та опишіть задачу, що вирішується.

6. Виконайте наступні класифікаційні задачі:
   - запустіть класифікатор IBk для різних значень K (1, 2, 3, 5, 10, 20);
   - укажіть точність класифікації для кожного значення K;
   - до якого типу класифікаторів відноситься IBk.

7. Виконайте класифікацію набора даних за допомогою алгоритму J48.
   - яка точність класифікації;
   - до якого типу класифікаторів відноситься J48;
   - змінюючи параметри налаштування minNumObj, subtreeRaising та unpruned, порівняйте побудовані дерева та точність класифікації;
   - порівняйте результати роботи класифікаторів IBk і J48.

8. Запустіть класифікатори J48 і IBk, використовуючи:
   - крос-перевірку з різними значеннями параметру fold;
   - процентний поділ вибірки на навчальну і тестову з трьома різними пропорціями.
   - порівняйте точність результатів.

9. Виконайте наступні задачі:
   - видаліть примірники, які відносяться до наступних класів: «build wind float» та «build wind non-float»;
   - виконайте класифікацію за допомогою класифікаторів IBk і J48;
   - визначте як проведена фільтрація вплинула на точність класифікації.

10. Запустіть класифікатори J48 і NaiveBayes на наступних наборах даних і визначте точність класифікації:
   - vehicle.arff;

- kr-vs-kp.arff;
- glass.arff;
- wave-form-5000.arff.

На яких наборах даних класифікатор NaiveBayes працює більш ефективно? Чому? Це пов'язано з природою атрибутів даних?

11. Для однієї з вибірок виконайте наступні завдання:
    - використовуючи результати роботи класифікатора J48, визначте найбільш значущі атрибути;
    - видаліть найменш значущі атрибути;
    - запустіть класифікатори J48 і IBk та визначте, як проведені зміни вплинули на точність роботи класифікаторів.

Який висновок можна зробити із отриманих результатів?

12. Використайте класифікатор SMO для наступних задач:
    - для набора даних ‘breast-w.arff’;
    - завантажте набір даних ‘glass.arff’ та залиште екземпляри двох класів: ‘build wind float’ та ‘head lamps ’, в якості атрибутів залиште ‘RI’, ‘Na’ та клас.

Визначте точність класифікації в обох випадках та візуалізуйте помилки класифікатора.

Частина В.

13. Для індивідуального завдання (додаток А) вирішите задачу класифікації за допомогою всіх розглянутих в лабораторній роботі алгоритмів. Змінюючи параметри налаштування алгоритмів, спробуйте досягти найліпшої якості навчання класифікаторів. В звіті наведіть результати роботи алгоритму та його налаштування.

2.4 Зміст звіту

1. Тема і мета роботи.
2. Завдання до роботи.
3. Результати виконання завдань розділу 2.3.
4. Висновки, що відображують результати виконання роботи та їх критичний аналіз.

ЛАБОРАТОРНА РОБОТА № 3
ПОПЕРЕДНЯ ОБРОБКА ДАНИХ ДЛЯ ЗАДАЧ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

3.1 Мета роботи

На практиці вивчити методи попередньої обробки даних для задач інтелектуального аналізу даних. На практичних прикладах оцінити вплив попередньої обробки даних на результати аналізу.

3.2 Основні теоретичні відомості

Дані володіють таким параметром як якість, яке включає наступні параметри: точність, повнота, несуперечність, своєчасність, достовірність та інтерпретованість.

Для підвищення якості даних і підготовки їх до обробки методами інтелектуального аналізу існує кілька технологій попередньої обробки даних.

До основних задач попередньої обробки даних відносяться наступні.

Задача очищення даних, яка використовується для заповнення пропущених значень, видалення шумів, видалення суперечливості, ідентифікації та видалення викидів.

Задача інтеграції даних із різних джерел (баз даних, кубів даних, файлів) в одне узгоджене сховище. Ця задача передбачає об'єднання даних і усунення неузгодженостей, дублікатів, конфліктів.

Задача проріджування та стиснення даних використовується для зменшення розміру даних з мінімізацією втрати інформації. Ця задача включає зниження розмірності даних (відбір атрибутів) і чисельне зменшення (побудова мат. моделей для значень атрибутів).

Задача перетворення даних. До неї відносяться нормалізація, дискретизація, квантування, згладжування, агрегація даних, відображення даних за допомогою ядерних функцій.

Також до попередньої обробки даних можна віднести *перетворення задачі множинної класифікації в бінарну*.

3.2.1 Відбір атрибутів

У більшості практичних ситуацій набори даних містять занадто багато атрибутів, що збільшує час навчання алгоритмів. При цьому

деякі з атрибутів є незначущими чи надмірними. Таким чином дані повинні бути попередньо оброблені з метою відбору деякої мінімальної підмножини атрибутів для навчання.

Для вибору хорошої підмножини атрибутів існує два підходи.

Перший з них заснований на незалежній оцінці статистичних чи якихось інших характеристиках набору даних. Він називається фільтрацією і відбувається до початку безпосереднього аналізу даних.

У другому підході відбір підмножини атрибутів виконується всередині методів інтелектуального аналізу. Такий підхід називається методом обгортки (wrapper method), тобто алгоритм навчання «обгорнутий» в процедуру відбору атрибутів.

Самі методи інтелектуального аналізу також можуть бути використані для відбору атрибутів. Наприклад, можна застосувати алгоритм побудови дерев рішень до повного набору даних і потім залишити в наборі тільки ті атрибути, які використані в побудованому дереві. Слід зауважити, що даний відбір атрибутів не дасть ніякого ефекту при побудові нового дерева, проте виявиться корисним при використанні інших методів аналізу.

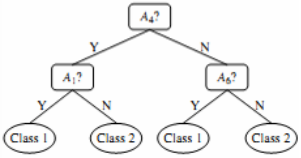
Інша можливість - це застосувати до даних алгоритм, який будує лінійну модель (наприклад, метод опорних векторів), і ранжувати атрибути на підставі величин коефіцієнтів моделі. Атрибути з найменшими коефіцієнтами можуть бути відкинуті. Дану процедуру можна повторити кілька разів.

Крім того для відбору атрибутів можуть бути застосовані методи аналізу, засновані на порівнянні близькості екземплярів вибірки. Для порівняння беруться сусідні примірники однакових і різних класів. Якщо у примірників одного класу значення певної атрибути різні, то можна припустити, що даний атрибут є незначущою і її вага повинна бути зменшена. З іншого боку, якщо у екземплярів різних класів атрибут має різні значення, то даний атрибут значимий і його вага повинна бути збільшена. Після повтору даної процедури кілька разів, відбувається відбір атрибутів з найбільшими вагами. До недоліків даного методу можна віднести той факт, що даний метод не зможе визначити надлишкові атрибути, пов'язані тісним кореляційним зв'язком.

Зазвичай пошук в просторі атрибутів відбувається в одному з двох напрямків: зверху вниз (починаючи з повного набору атрибутів і відкидаючи на кожному кроці найгірший з них) або знизу вгору

(починаючи з порожньої множини атрибутів і додаючи найкращий з решти) (див. табл. 3.1).

Таблиця 3.1 – Пошук в просторі атрибутів

Прямий вибір (forward selection)	**Зворотнє виключення (backward elimination)**	**Застосування дерев рішень**
Початкова множина атрибутів {A1, A2, A3, A4, A5}		
{} => {A1} => {A1, A4} => {A1, A4, A5}	{A1, A2, A3, A4, A5} =>{A1, A2, A4, A5} =>{A1, A4, A5}	 =>{A1, A4, A5}

У деяких випадках для поліпшення точності класифікації та кращого розуміння атрибутів для вирішення поставленого завдання можлива побудова нового атрибуту на основі існуючих. Наприклад, можна ввести новий атрибут «Площа» на основі існуючих атрибутів «висота» і «ширина».

3.2.2 Пропущені значення

При роботі з даними, в яких є пропущені значення атрибутів для деяких екземплярів, існують наступні стратегії поведінки.

1. Відкинути екземпляри з пропущеними значеннями. Такий підхід застосовується насамперед для даних, у яких пропущено значення цільового атрибута (для задач класифікації).
2. Заповнити пропущені значення вручну.
3. Застосувати глобальну константу (наприклад, "Unknown").
4. Використати деяке статистично розраховане по всій вибірці значення (середнє арифметичне, медіану, моду).
5. Використати статистичне значення, розраховане для примірників, що відносяться до того ж класу, як і розглянутий екземпляр.
6. Використати найбільш ймовірне значення для атрибуту. Це значення може бути розраховане за допомогою регресії, дерева рішень або інших математичних підходів.

3.2.3 Нормалізація даних

Одиниці виміру, що використовуються в деякому атрибуті, можуть вплинути на результати аналізу. Так, наприклад, перетворення одиниць вимірювання з метрів в дюйми для атрибуту «висота» або перетворення з кілограмів у фунти для атрибуту «вага» можуть призвести до різних результатів. У загальному випадку, вираз деякої атрибуту в дрібніших одиницях виміру приведе до більш широкого діапазону значень для цього атрибуту, що може призвести до більшої значущості або ж ваги даного атрибуту.

Щоб уникнути залежності від вибору одиниць вимірювання та надати всім атрибутам однакову вагу дані повинні бути нормалізовані або нормовані. Нормалізація передбачає перетворення даних таким чином, щоб діапазон значень, прийнятих атрибутом, зменшився або став рівним [-1; 1] або [0; 1]. Нормалізація найбільш корисна в задачах з застосуванням нейронних мереж та задачах, алгоритми яких засновані на обчисленні відстаней.

Нехай у нас є числовий атрибут A з виміряними значеннями a_i.

Мінімаксна нормалізація. Нехай $\min_a$ – мінімальне значення атрибуту, $\max_a$ – максимальне, $[new\_\min_a; new\_\max_a]$ – новий діапазон для атрибуту, тоді:

$$a_i' = \frac{a_i - \min_a}{\max_a - \min_a}(new\_\max_a - new\_\min_a) + new\_\min_a .$$

Нормалізація з нульовим середнім. Значення атрибуту нормалізуються за допомогою математичного очікування $\bar{a}$ та стандартного відхилення σ_a атрибуту:

$$a_i' = \frac{a_i - \bar{a}}{\sigma_a} .$$

Існує варіація нормалізації з нульовим середнім, в якому замість стандартного відхилення атрибуту використовує середнє абсолютне значення атрибуту:

$$S_a = \frac{1}{n}\left(|a_1 - \bar{a}| + |a_2 - \bar{a}| + ... + |a_n - \bar{a}|\right) .$$

Нормалізація за допомогою десяткової шкали:

$$a_i' = \frac{a_i}{10^j},$$

де j - найменше ціле число, таке що ($|a_i|$)<1.

3.2.4 Дискретизація числових атрибутів

Дискретизація числових атрибутів є обов'язковою і необхідною у разі застосування алгоритмів інтелектуального аналізу, що працюють тільки з категоріальними атрибутами. Крім того, алгоритми, що працюють з числовими атрибутами часто дають кращі результати або ж працюють швидше, якщо значення атрибутів попередньо приведені до дискретної форми.

Методи дискретизації можуть бути класифіковані за двома параметрами:

– чи використовується в них інформація про класи: дискретизація з учителем (supervised discretization) або дискретизація без вчителя (unsupervised discretization);

– в якому напрямку відбувається дискретизація:
  - зверху-вниз (дискретизація починається з однієї або декількох точок поділу, а далі отримані інтервали рекурсивно розбиваються; метод розбиття);
  - знизу-вгору (спочатку всі значення атрибуту розглядаються як потенційні точки поділу, а далі сусідні значення рекурсивно об'єднуються, утворюючи інтервали; об'єднання).

3.2.5 Вибірка/семплювання (sampling)

Вибірка або семплювання застосовується в якості методу зменшення початкового набору даних з метою представлення великої вихідної множини екземплярів вибірки набагато меншою за розміром підмножиною.

Припустимо, що вихідний набір даних D містить N примірників. Розглянемо найбільш загальні шляхи зменшення його розміру.

Проста випадкова вибірка без повернення: з вихідного набору D випадковим чином вибирається S примірників (S<N), при цьому ймовірність вибору кожного примірника рівноймовірна.

Проста випадкова вибірка з поверненням: схожа на попередню, однак з тією відмінністю, що після вибору примірника, він повертається у вихідну вибірку і згодом знову може бути вибраний.

Кластерна вибірка: якщо вихідна вибірка згрупована в деякі роз'єднаним «кластери» (наприклад, сторінки з бази даних або дані з різних географічних джерел), то до кожного з таких кластерів може бути застосована проста випадкова вибірка.

Стратифікована вибірка: якщо вихідна вибірка несиметрична щодо розподілу класів і може бути розділена на страти, то проста випадкова вибірка застосовується до кожної страти окремо (наприклад, якщо дані представляють відомості про покупців різних вікових груп і при цьому кількість представників різних груп не однакова, такий підхід дозволить не втратити відомості про рідкісні групи покупців).

3.3 Завдання на лабораторну роботу

1. Опишіть фільтри WEKA, які можуть бути використані для основних задач попередньої обробки даних.

Частина А. Застосування фільтрів дискретизації даних.

2. Завантажте набір даних ‘sick.arff’, застосуйте наївний Байєсовий класифікатор з крос-перевіркою та укажіть точність класифікатора по кожному з класів.

3. Перейдіть на вкладку попередньої обробки даних і виконайте наступні завдання, скасовуючи зміни після кожного:

- застосуйте фільтр дискретизації з учителем, оцініть ефект його роботи, визначте скільки різних діапазонів було створено для кожного атрибута;
- застосуйте фільтр дискретизації без учителя двічі: з параметром 'bins' рівним 5 та 10; дайте оцінку, як в залежності від значення параметру 'bins' змінюються дані.

4. Запустіть мета класифікатор 'FilteredClassifier' з класифікатором Naive Bayes і наступними параметрами фільтрації:

дискретизації без учителя з ‘bins’=5, 10, 20;

дискретизації з учителем.

5. Порівняйте точність навчання наївного Байєсового класифікатора для наступних випадків:

- без використання фільтрів дискретизації;

- з використанням фільтру дискретизації з учителем;
- з використанням фільтру дискретизації без учителя з різними значеннями параметра 'bins'.

У частинах Б та В спробуємо поліпшити точність роботи трьох різних класифікаторів шляхом відбору найкращих атрибутів. Ми використаємо дві різні моделі оцінки значущості атрибутів: ‘WrapperSubsetEval’, який оцінює підмножини атрибутів за допомогою методів інтелектуального аналізу, и ‘CfsSubsetEval’, який оцінює важливість кожної підмножини атрибутів шляхом оцінки прогнозувальних можливостей кожного з них окремо разом з оцінкою ступеня надмірності.

Частина Б. Відбір атрибутів 1.

6. Завантажте набір даних 'mushroom.arff' та виконайте класифікацію за допомогою алгоритмів J48, 1Bk і Naive Bayes з крос-перевіркою та дайте оцінку точності кожного із класифікаторів.
7. На підставі роботи класифікатора J48 визначте найбільш значущі атрибути, видаліть незначущі і знову повторіть класифікацію даних. Визначте як змінилася точність кожного з класифікаторів.
8. Поверніть дані в початковий стан і вирішіть задачу відбору атрибутів на вкладці ‘Select Attributes’. В якості алгоритму оцінки атрибутів оберіть метод ‘CFSSubsetEval’, в якості алгоритму пошуку – метод ‘Greedy Stepwise’, Attribute selection mode = cross-validation. Проаналізуйте інформацію у вікні результатів та порівняйте визначені значущі атрибути з результатами, отриманими за допомогою класифікатора J48.
9. Запустіть мета класифікатор ‘AttributeSelectedClassifier’ з наступними параметрами: evaluator = CFSSubsetEval, search = GreedStepwise, classifier = J48, 1Bk, NaiveBayes:
   - запишіть точність роботи класифікаторів;
   - порівняйте результати до та після відбору атрибутів;
   - чи збільшилась точність розпізнавання, чому так або ні;
   - які переваги відбору атрибутів.

Частина В. Відбір атрибутів 2

10. Завантажте набір даних ‘vote.arff’ , виконайте класифікацію за допомогою алгоритмів J48, 1Bk та Naive Bayes з крос-перевіркою та

запишіть точність роботи класифікаторів.

11. Перейдіть на вкладку відбору атрибутів і виконайте наступні завдання:

- в якості алгоритму оцінки атрибутів встановіть ‘WrapperSubsetEval’, алгоритм пошуку – ‘RankSearch’;
- в якості алгоритму оцінки атрибутів встановіть ‘InfoGainAttributeEval’, алгоритм пошуку – ‘Ranker’;
- порівняйте отримані результати;

12. Запустіть мета класифікатор ‘AttributeSelectedClassifier’ з наступними параметрами:

1. WrapperSubsetEval и RankSearch
2. InfoGainAttributeEval и Ranker
3. classifier = J48, 1Bk, NaiveBayes.

- порівняйте точність роботи класифікаторів до та після відбору атрибутів, чи збільшилась точність розпізнавання, чому так або ні?

Частина В. Семплювання.

13. Завантажте набір даних ‘letter.arff’ і для нього:

- для довільного атрибута запишіть його параметри: мінімальне, максимальне, середнє значення, стандартне відхилення;
- застосуйте фільтр 'Resample’ з параметром 'sampleSizePercent' встановленим в 50%;
- оцініть розмір відфільтрованого набору даних, оцініть значення записаних параметрів та відсоток їхньої зміни;
- які переваги семплювання великого набору даних?

Частина Г.

14. Для індивідуального завдання проведіть попередню обробку даних та спробуйте поліпшити результати класифікації.

3.4 Зміст звіту

1. Тема і мета роботи.
2. Завдання до роботи.
3. Результати виконання завдань розділу 3.3.
4. Висновки, що відображують результати виконання роботи та їх критичний аналіз.

ЛАБОРАТОРНА РОБОТА № 4
ЗАДАЧА РЕГРЕСІЇ

4.1 Мета роботи

На практиці вивчити роботу алгоритмів, що вирішують задачу регресії, і навчитися інтерпретувати результати їх роботи.

4.2 Основні теоретичні відомості

У роботі розглядаються такі методи (у дужках наведено назву в WEKA):

- лінійна регресія (functions.LinearRegression);
- регресійні дерева і модельні дерева (trees.M5P);
- метод опорних векторів, модифікований для вирішення задач регресії (functions.SMOreg);
- метод найближчих сусідів (lazy.IBk).

4.2.1 Параметри налаштування алгоритмів

Розглянемо параметри налаштування використовуваних алгоритмів у WEKA (табл. 2.1 та 4.1).

Таблиця 4.1 – Параметри налаштування методів

Метод	**Параметри**
Linear Regression	attributeSelectionMethod – метод відбору атрибутів. eliminateColinearAttributes – виключити колінеарні атрибути. ridge – штраф за великі значення коефіцієнтів регресії (регуляризація Тихонова).
M5P	buildRegressionTree –регресійне або модельне дерево. minNumInstances – мінімальна кількість примірників у листі. saveInstances – зберігати примірники у вузлах для візуалізації. unpruned – будувати дерево без відсікань. useUnsmoothed – незгладжене прогнозування.
SMOreg	Основні параметри можна знайти в табл. 2.1. regOptimizer – алгоритм навчання.

4.2.2 Методи оцінки якості прогнозування

Наведені в розділі 2.2.2 параметри більш корисні для опису задач класифікації ніж для завдань регресії. Для задачі регресії помилки прогнозування не просто присутні або відсутні, а мають різні числові значенні. Для оцінки успішності числових прогнозів можуть бути використані альтернативні міри, деякі з яких наведено в табл. 4.2.

Таблиця 4.2 – Міри оцінки якості вирішення задачі регресії

Параметр	**Формула для розрахунку**
Середній квадрат помилки (mean-squared error)	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
Середньоквадратична помилка (root mean-squared error)	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
Середня абсолютна помилка (mean-absolute error)	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
Відносний квадрат помилки (relative-squared error)*	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$
Root relative-squared error*	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
Relative-absolute error*	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
Коефіцієнт кореляції (correlation coefficient)**	$\frac{S_{PA}}{\sqrt{S_P S_A}}$,
де $S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_P=\frac{\sum_i (p_i-\bar{p})^2}{n-1}$, $S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}$	

де p_1 , p_2 , …, p_n – прогнозовані значення для цільового атрибута тестової вибірки;

a_1, a_2 , …, a_n – реальні значення цільового атрибута;

$\bar{a}$ – середнє арифметичне (* – навчальної вибірки, ** – тестової).

4.3 Завдання на лабораторну роботу

1. Оберіть в таблиці А.2 два набори даних. Виконайте для них наступні завдання.
2. Оберіть 10 довільних екземплярів з вибірки та винесіть їх в окремий файл тестової вибірки. В якості навчальної вибірки для побудови моделі використайте вихідний файл без відібраних екземплярів.
3. Візуалізуйте вихідні дані та зробіть припущення про можливість передбачення цільового атрибуту.
4. При необхідності обробіть вихідні дані (див. лаб. 3).
5. Вирішіть задачу регресії за допомогою чотирьох методів:
   - Linear regression;
   - SMOreg;
   - M5P (model trees and regression trees) з наступними параметрами налаштування: build regression tree: True/False, unpruned: True/False, useUnsmoothed: True;
   - kNN.

   В якості режиму тестування оберіть:
   - «Percentage split» (66% вибірки для навчання);
   - «Cross-validation» (10 фолдів).
6. Запишіть отримані моделі і порівняйте їхню ефективність (точність передбачення).
7. Оцініть точність побудованих моделей на створеній в п.2 тестовій вибірці.
8. Які з атрибутів є найбільш значущими для передбачення значень цільового атрибуту, судячи з побудованих моделей? Чому? Як зміниться точність передбачення, якщо залишити лише значущі атрибути?
9. Відберіть найбільш значущі атрибути (см. лаб. 3) та повторіть побудову моделей. Оцініть точність нових моделей.

4.4 Зміст звіту

1. Тема і мета роботи.
2. Завдання до роботи.
3. Результати виконання завдань п.4.3.
4. Висновки, що відображують результати виконання роботи та їх критичний аналіз.

ЛАБОРАТОРНА РОБОТА № 5
ЗАДАЧА КЛАСТЕРИЗАЦІЇ

5.1 Мета роботи

На практиці вивчити роботу алгоритмів кластеризації, навчитися інтерпретувати результати їх роботи і вибирати найкращий метод для розв'язуваної прикладної задачі.

5.2 Основні теоретичні відомості

У лабораторній роботі розглядаються наступні методи кластеризації (у дужках наведено назву на WEKA):

- поділяючий метод кластеризації K-середніх (SimpleKMeans);
- ієрархічний метод кластеризації (HierarchicalClusterer);
- імовірнісний метод кластеризації EM (EM);
- ієрархічний метод кластеризації COBWEB (COBWEB);
- метод заснований на щільності розташування об'єктів DBSCAN (DBScan).

5.2.1 Параметри налаштування алгоритмів

Розглянемо параметри налаштування використовуваних алгоритмів кластеризації в WEKA (табл. 5.1).

Таблиця 5.1 – Параметри налаштування кластеризаторів

Метод	**Параметри**
SimpleKMeans	displayStdDevs – відображати значення стандартного відхилення для числових атрибутів і підрахунки для номінальних атрибутів. distanceFunction – функція відстані. dontReplaceMissingValues – не замінювати пропущені значення середнім значенням або модою. maxIterations – максимальна кількість ітерацій алгоритму. numClusters – кількість кластерів. preserveInstancesOrder – зберігати порядок примірників у вибірці. seed – випадковий сид для рандомізації вибірки.

Продовження табл. 5.1.

Метод	**Параметри**
Hierarchical Clusterer	distanceFunction – функція відстані. distanceIsBranchLength – у дендрограмі висота лінії, що зв'язує кластери, буде показувати відстань між ними. linkType – тип зв'язку для розрахунку відстані між двома кластерами. numClusters – кількість кластерів. printNewick – виводити кластери в форматі Newick.
EM	displayModelInOldFormat – використовувати старий формат представлення моделі (у випадках великої кількості кластерів). maxIterations – максимальна кількість ітерацій алгоритма. minStdDev – мінімальне значення стандартного відхилення. numClusters – кількість кластерів (встановити значень -1 для автоматичного вибору кількості кластерів). seed – випадковий сид для рандомізації вибірки.
DBSCAN	database_Type – використовувана база даних. database_distanceType – функція відстані. epsilon – радіус пошуку. minPoints – мінімальна кількість об'єктів усередині радіуса.
COBWEB	acuity – мінімальне значення стандартного відхилення для числових атрибутів. cutoff - встановити поріг до якого відсікати вузли дерева. saveInstanceData – зберегти інформацію про примірники для візуалізації. seed – випадковий сід для рандомізації вибірки.

5.2.2 Інтерпретація результатів кластеризації в WEKA (Clusterer output)

Секція «Clustering model» відображає побудовану модель.

Для алгоритму SimpleKMeans ця секція буде містити кількість ітерацій алгоритму, загальну квадратичну помилку для всіх кластерів та центроїди побудованих кластерів. В ній також буде вказано

застосований вид обробки порожніх значень атрибутів у об’єктів.

Для алгоритму EM буде вказана кількість кластерів, на які було розбито дані, кількість ітерацій алгоритму та центроїди побудованих кластерів.

Для алгоритму COBWEB буде вказана кількість об’єднань та розділів даних, кількість побудованих кластерів та текстове представлення побудованої ієрархії.

Для алгоритму DBSCAN ця секція буде містити налаштування алгоритму, кількість побудованих кластерів, віднесення кожного з об’єктів вибірки до конкретного кластеру чи до викидів.

Секція «Model and evaluation» містить інформацію про кількісний розподіл екземплярів по кластерах. При цьому буде вказано, скільки об’єктів було кластеризовано (Clustered Instances), а скільки не увійшли в жоден з кластерів (Unclustered instances).

Якщо було обрано опцію «Classes to clusters evaluation» (порівняння попередньої заданих класів з кластерами), то ця секція також буде містити результати оцінки якості кластеризації. Буде вказано, який з побудованих кластерів відповідає якому класу, буде побудовано матрицю невідповідності та вказана кількість невірно кластеризованих екземплярів.

5.3 Завдання на лабораторну роботу

Частина А.

1. Завантажте набір даних ‘bank.arff’.
2. Запустіть алгоритм кластеризації SimpleKMeans, задаючи значення параметра K (кількість кластерів) від 1 до 12. Запишіть в таблицю значення сум квадратичних помилок, одержуваних при різних значеннях K. Що означає цей параметр? Чи спостерігається який-небудь тренд в поведінці значень даного параметра?
3. Для значення K=5 укажіть:
   - скільки кластерів було створено;
   - скільки примірників потрапило в кожен з кластерів (вказати кількість і відсоток);
   - скільки ітерацій знадобилося для кластеризації даних;
   - складіть таблицю з характеристиками центроїдів.
4. Для значення K=5 візуалізуйте результати кластеризації (по осі абсцис відкласти назву (номер) кластера, по осі ординат - номер

примірника в кластері) та дайте оцінку отриманим результатам:

- чи є значна відмінність у значеннях атрибуту «вік» (age) між кластерами?
- у яких кластерах домінують жінки (female), а в яких чоловіки (male)?
- що можна сказати про значення атрибута «регіон» (region) у кожному кластері?
- що можна сказати про розкид значень атрибуту «дохід» (income) між кластерами?
- у яких кластерах домінують сімейні люди (married), а в яких холості (unmarried)?
- у якій кластер потрапило найбільше людей з машинами?
- у яких кластерах переважають люди з ощадними рахунками (savings accounts)?
- що можна сказати про розкид значень атрибуту «поточний банківський рахунок» (current account) між кластерами?
- що можна сказати про розкид значень атрибуту «іпотека» (mortgage holdings) між кластерами?
- які кластери в основному складаються з людей, які придбали PEP (особистий план купівлі акцій), і які з людей, які не придбали його?

5. Запустіть алгоритм кластеризації EM та оцініть результати.

Частина Б.

6. Виконайте наступні завдання для набору даних ‘iris.arff’.

7. Запустіть алгоритм SimpleKMeans з K=3 та оцініть якість кластеризації, порівнюючи кластери з попередньо заданими класами:

- запишіть значення суми квадратичних помилок, кількість об'єктів в кластерах та характеристики кожного центроїду;
- проаналізуйте як співвідносяться кластери та значення цільового атрибуту, скільки екземплярів було віднесено до «невірних» кластерів, який клас виявився «складним» для виділення;
- візуалізуйте результати, використовуючи різні атрибути для осі ординат (при візуалізації екземпляри, позначені квадратами були віднесені до «невірного» кластеру);
- визначте, на що впливає параметр «seed» і чому він є важливим при кластеризації методом k-середніх; для

цього проведіть експерименти з різними значеннями параметру і порівняйте отримані результати.

Частина В. Ієрархічна кластеризація.

8. Завантажте набір даних ‘weather.arff’ та запустіть алгоритм COBWEB, встановивши наступні параметри: saveInstanceData = True, cluster mode = Use training set. В результаті роботи алгоритму буде побудована дендрограма, що представляє ієрархічну структуру кластерів. Відобразіть її в текстовому вигляді та у вигляді дерева.

9. Завантажте набір даних ‘flagdata.arff, що представляє атрибути прапорів деяких європейських країн. Запустіть алгоритм COBWEB з параметром C=0,4 та проаналізуйте результати:

- візуалізуйте отриману дендрограму та запишіть її;
- укажіть, що спільного у прапорів, що опинилися в одному кластері.

10. Завантажте набір даних ‘zoo.arff’, оберіть з вибірки частину тварин на власний розсуд (наприклад, ссавців) та виконайте завдання:

- запустіть алгоритм Hierarchical Clusterer (назву тварини та його тип не використовувати при кластеризації);
- проекспериментуйте з налаштуванням алгоритму та візуалізуйте результати його роботи;
- оцініть, чи є логічний сенс в створюваних кластерах.

Частина Г. Алгоритм DBScan

11. Для використання алгоритму щільністної кластеризації згенеруйте набор даних за допомогою алгоритму BIRCHCluster. В наборі згенеруйте також флаг класу.

12. За допомогою налаштувань методу DBScan досягніть найкращої кластеризації даних.

13. Кластеризуйте набір даних за допомогою інших алгоритмів. Який з алгоритмів виявився найбільш ефективним?

5.4 Зміст звіту

1. Тема і мета роботи.
2. Завдання до роботи.
3. Результати виконання завдань розділу 5.3.
4. Висновки, що відображують результати виконання роботи та їх критичний аналіз.

ЛАБОРАТОРНА РОБОТА № 6
ПОШУК АСОЦІАТИВНИХ ПРАВИЛ

6.1 Мета роботи

На практиці вивчити роботу алгоритмів пошуку асоціативних правил і навчитися інтерпретувати результати їх роботи.

6.2 Основні теоретичні відомості

У лабораторній роботі розглядаються два методи пошуку асоціативних правил:

- алгоритм Apriori;
- алгоритм FPGrowth.

6.2.1 Параметри налаштування алгоритмів

Розглянемо параметри налаштування використовуваних алгоритмів пошуку асоціативних правил в WEKA (табл. 6.1).

Таблиця 6.1 – Параметри налаштування алгоритмів

Метод	**Параметри**
Apriori	car – пошук класових (зі значенням цільового атрибута в правій частині) або звичайних асоціативних правил. classIndex – індекс цільового атрибута. Якщо встановлено значення -1, буде обраний останній атрибут. delta – ітеративно зменшувати значення порогу підтримки на дане значення. Зменшення буде відбуватися до тих пір, поки не буде досягнуто мінімальне значення підтримки чи не буде згенеровано задану кількість правил. lowerBoundMinSupport – нижня межа порогу підтримки. metricType – встановлює тип метрики, за якою будуть ранжуватися правила (Confidence, Lift, Leverage, Conviction). minMetric – мінімальне граничне значення для обраної метрики. numRules – кількість правил, які необхідно знайти. outputItemSets – чи виводити часті набори. removeAllMissingCols – прибирати чи колонки (атрибути) в яких всі значення відсутні.

Продовження табл.6.1.

Метод	**Параметри**
	significanceLevel – рівень значущості (тільки для достовірності). upperBoundMinSupport – верхня межа мінімальної підтримки. Ітеративне зменшення підтримки починається з цього значення.
FPGrowth	delta – ітеративно зменшувати значення порогу підтримки на дане значення. Зменшення буде відбуватися до тих пір, поки не буде досягнуто мінімальне значення підтримки чи не буде згенеровано задану кількість правил. findAllRulesForSupportLevel – знайти всі правила, які задовольняють нижній межі мінімального значення підтримки та мінімальному значенню метрики. Включення цього режиму скасує виконання ітеративного зменшення підтримки для знаходження заданого кількості правил. lowerBoundMinSupport - нижня межа порогу підтримки як частка кількості примірників. maxNumberOfItems – максимальна кількість примірників у частому наборі; значення -1 означає без обмежень. metricType – встановлює тип метрики, за якою будуть ранжуватися правила. minMetric – мінімальне граничне значення для метрики. numRulesToFind – кількість правил, які необхідно знайти. positiveIndex – встановлює індекс бінарного атрибуту, який буде розглядатися як позитивний. rulesMustContain – виводити правила, які містять задані об'єкти (список об'єктів, розділених комою). transactionsMustContain – для роботи алгоритму використовувати транзакції (примірники), які містять задані об'єкти. upperBoundMinSupport – верхня межа мінімальної підтримки. Ітеративне зменшення підтримки починається з цього значення. useORForMustContainList – - використовувати логічний зв'язку «або» замість «і» для списків обов'язкових елементів у транзакціях і правилах.

6.2.2 Інтерпретація результатів

Секція «Associator model» містить інформацію про побудовану модель.

Для алгоритму Apriori секція містить значення мінімальної підтримки, мінімальне значення обраної метрики (зазвичай достовірність), кількість циклів алгоритму. Далі розташовані знайдені часті набори даних та знайдені правила.

Для алгоритму FPGrowth буде вказана загальна кількість знайдених правил та задана кількість найліпших з них.

Представимо асоціативне правило у вигляді X => Y. Для кожного з правил будуть відображені деякі числові параметри: число перед стрілкою вказує кількість екземплярів вибірки, для яких ліва умовна частина правила вірна (N_X), а число після стрілки вказує кількість екземплярів вибірки, для яких вірна ліва і права частини правила (N_{XUY}).

Підтримка (support) правила дорівнює:

$$\sup(X \Rightarrow Y) = N_{X \cup Y} / N .$$

Достовірність (confidence) правила – це відношення підтримки всього правила до підтримка лівої частини правила:

$$conf(X \Rightarrow Y) = \frac{\sup(X \Rightarrow Y)}{\sup(X)} = \frac{N_{X \cup Y}}{N_X} .$$

Параметр Lift визначається діленням достовірности на підтримку прової частини правила:

$$lift(X \Rightarrow Y) = \frac{\sup(X \Rightarrow Y)}{\sup(X) * \sup(Y)} = \frac{conf(X \Rightarrow Y)}{\sup(Y)} ,$$

а параметр Conviction дорівнює:

$$conv(X \Rightarrow Y) = \frac{1 - \sup(Y)}{1 - conf(X \Rightarrow Y)} .$$

Параметр Leverage – це відношення додаткових екземплярів, що

покриваються правилом, до очікуваних в тому випадку, коли ліва і права частини правила були би статистично незалежними.

Наприклад, нехай ми маємо 1000 екземплярів в вибірці (N), при цьому ліва частина правила покриває 200 з них (NX), права окремо покриває 100 (NY) і все правило покриває 50 (NXUY). Частка екземплярів, що покриваються правилом (підтримка) дорівнює 50/1000=0,05. Кількість екземплярів, які були б покриті правилом у випадку, коли ліва і права частина правила незалежні, дорівнює 200 * 100 / 1000 = 20. Параметр leverage для цього прикладу дорівнює 50 - 20=30, що в пропорції від загальної вибірки дорівнює 30/1000=0,03.

6.3 Завдання на лабораторну роботу

Частина А.

1. Виконайте наступні завдання для набору даних ‘vote.arff’.
2. Запустіть алгоритм пошуку асоціативних правил Apriori.
3. Яке значення для порогу підтримки було використано в побудованій моделі? Яке значення для порогу достовірності було використано?
4. Запишіть 6 найкращих знайдених правил, вкажіть для них значення підтримки та достовірності.
5. Що позначають числа ліворуч і праворуч від стрілки в знайдених асоціативних правилах?
6. Для правила номер 8 проведіть обчислення підтримки та достовірності, використовуючи вкладку попередньої обробки даних. Порівняйте отримані значення з граничними.
7. Що означають параметри support, confidence, lift, conviction, застосовувані в алгоритмі Apriori?
8. Запустіть алгоритм пошуку асоціативних правил FPGrowth.
9. Порівняйте списки десяти найкращих правил, отриманих двома алгоритмами. Поясніть відмінність в роботі двох алгоритмів.
10. Запустіть алгоритм Apriori задавши значення car = true. Які асоціативні правила були отримані? Що знаходиться в правій частині знайдених асоціативних правил?
11. Вирішіть задачу пошуку асоціативних правил для наборів даних 'bank-data.csv' та 'marketbasket.arff'. Проаналізуйте та поясніть знайдені правила.

12. Яку попередню обробку необхідно провести з набором даних 'anduin_data.arff', щоб мати можливість застосувати до нього алгоритм Apriori? Застосуйте необхідні фільтри і вирішіть задачу пошуку асоціативних правил. Проаналізуйте знайдені асоціативні правила.

Частина Б.

13. Спробуйте відшукати у власному індивідуальному завданні нові шаблони (асоціативні правила) за допомогою двох розглянутих алгоритмів. Згенеруйте також правила, в правій частині яких буде знаходитися ваш клас.

6.4 Зміст звіту

1. Тема і мета роботи.
2. Завдання до роботи.
3. Результати виконання завдань п.6.3.
4. Висновки, що відображують результати виконання роботи та їх критичний аналіз.

ЛІТЕРАТУРА

1. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. – СПб.: БХВ-Петербург, 2007. – 384 с.

2. Чубукова И.А. Data Mining: учебное пособие / И.А. Чубукова. – М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2006. – 382 с.

3. Witten, I.H. Data mining: practical machine learning tools and techniques.—3rd ed. / Ian H. Witten, Frank Eibe, Mark A. Hall. – Morgan Kaufmann Publishers, 2011. – 629 p.

4. Han J. Data Mining: Concepts and Techniques (Second Edition) / J. Han, M. Kamber – Morgan Kaufmann Publishers, 2006. – 800 p.

5. Weka 3: Data Mining Software in Java [Електронний ресурс] – Режим доступу: http://www.cs.waikato.ac.nz/ml/weka.

6. Weka 3 Wiki documentation [Електронний ресурс] – Режим доступу: http://weka.wikispaces.com/

Додаток А.
Варіанти індивідуальних завдань

Обрати з таблиці за номером варіанту (N) з журналу набор даних для дослідження. Дослідити поставлену задачу, характеристики набору даних (атрибути), за необхідності провести попередню обробку даних та зменшити кількість об'єктів у вибірці, виділити аномалії та викиди, обрати стратегію роботи з об'єктами з пропусками, визначити стратегію тестування навчених алгоритмів. Для кожного з алгоритмів провести дослідження їх роботи на поставленій задачі, змінюючи параметри налаштування алгоритму.

Таблиця А.1 – Набори даних для задачі класифікації

1	adult.arff	11	nursery.arff
2	bank-data.arff	12	credit_froad.arff
3	breast-cancer.arff	13	sick.arff
4	breast-w.arff	14	spambase.arff
5	autos.arff	15	zoo.arff
6	contact-lenses.arff	16	tic-tac-toe.arff
7	credit.arff	17	mushroom.arff
8	diabetes.arff	18	vehicle.arff
9	heart-statlog.arff	19	vote.arff
10	labor.arff	20	wine.arff

Обрати одну задачу (N mod 7)+1. Друга задача на власний вибір студента.

Таблиця А.2 – Набори даних для задачі регресії

1	cpu.arff’	5	housing.arff
2	auto_mpg.arff	6	bodyfat.arff
3	winequality-red.csv	7	fishcatch.arff
4	autoprice.arff	8	auto93.arff