Міністерство освіти і науки України
Карпатський національний університет імені Василя Стефаника
Кафедра комп’ютерних наук та інформаційних систем

А. В. Ізмайлов

Методичні вказівки до виконання лабораторних робіт з дисципліни «Інтелектуальний аналіз даних»

Івано-Франківськ
2025

УДК 004
І37

Рекомендовано до друку Вченою радою факультету математики та інформатики Карпатського національного університету імені Василя Стефаника (протокол № 9 від 22 жовтня 2025р.)

Рецензенти:

Петришин М.Л., кандидат технічних наук, доцент кафедри КНІС (Карпатський національний університет імені Василя Стефаника)

Превисокова Н.В., кандидат технічних наук, доцент кафедри КНІС (Карпатський національний університет імені Василя Стефаника)

І37 Ізмайлов А.В. Методичні вказівки до виконання лабораторних робіт з дисципліни «Інтелектуальний аналіз даних». Івано-Франківськ : КНУВС, 2025. 33 с.

Наведено методичні вказівки і завдання до лабораторних робіт з дисципліни «Інтелектуальний аналіз даних». Призначено для проведення лабораторних занять з курсу «Інтелектуальний аналіз даних» та інших, у яких вивчаються різні аспекти аналізу даних.

Для здобувачів освіти за спеціальностями галузі F «Інформаційні технології». Може бути корисним для читачів, які займаються вивченням аналізу та обробки даних.

Зміст

Лабораторна робота №1

Знайомство із середовищем інтелектуального аналізу даних WEKA та первинна підготовка даних

***Мета роботи*:** Ознайомитись із середовищем інтелектуального аналізу даних WEKA, отримати навички роботи із даними у форматі arff, вивчити методи первинної підготовки даних для задач їх інтелектуального аналізу.

1.1 Теоретичні відомості

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.

Основний формат файлів даних, який використовується в WEKA, – це ARFF. У каталозі data, який знаходиться в каталозі встановленої програми, можна знайти 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

1.1,A

Заголовок містить інформацію про ім'я файлу і метадані про представлені у ньому дані. Метадані описують атрибути представлених у файлі даних. Інформація про кожний атрибут записується в окремому рядку і включає ім'я атрибуту і його тип. 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.

1.2 Хід роботи

1.2.1 Визначити (і навести обчислення) свій варіант за формулою:

$$\text{num} \ mod\ 3 + 1,$$

де num – номер у журналі, *mod* – операція взяття остачі від ділення.

1.2.2 На основі номеру варіанту обрати із таблиці 1.1 відповідний набір даних та завантажити його з репозитарію UC Irvine Machine Learning Repository, доступного за посиланням https://archive.ics.uci.edu/datasets/.

Таблиця 1.1 – Набори даних

Варіант	**Назва набору**
1	post-operative
2	tic-tac-toe
3	zoo

Набір даних складається з файлів із розширенням .data та .names. Перший містить безпосередньо дані, другий – метадані та опис набору. На основі цих файлів створити ARFF-файл для обраного набору даних, який міститиме поряд із даними, опис набору.

1.2.3 Використовуючи вкладки попередньої обробки даних та візуалізації, провести детальний опис вибірки даних. Вказати:

1.2.3.1 яке практичне завдання вирішується;

1.2.3.2 скільки екземплярів у вибірці;

1.2.3.3 атрибути, які характеризують екземпляри вибірки, їхні типи та опис (у вигляді таблиці);

1.2.3.4 чи є екземпляри із відсутніми значеннями і якщо є, то вказати атрибути, для яких є відсутні значення;

1.2.3.5 який атрибут є цільовим, які значення він приймає, скільки екземплярів кожного класу у вибірці;

1.2.3.6 перші 5 екземплярів вибірки (у вигляді таблиці).

1.2.4 Встановити додатковий пакет scatterPlot3D та візуалізувати дані з його допомогою. Включити у звіт результати візуалізації у обсязі 1-3 знімків екрану, залежно від складності візуалізації заданого набору даних.

1.2.5 Зробити висновки стосовно результатів використання середовища WEKA для первинної підготовки даних. Виділити особливості розглянутого набору даних та зробити припущення щодо проблем, які можуть виникнути у процесі подальшої його обробки та аналізу.

1.3 Зміст звіту

1. Титульна сторінка та тема роботи.
2. Мета роботи.
3. Короткі теоретичні відомості (до 1 сторінки).

4. Результати роботи.
5. Висновки

1.4 Контрольні запитання

1.4.1 Що таке інтелектуальний аналіз даних?

1.4.2 Для чого використовується середовище WEKA, які його можливості?

1.4.3 Яке призначення модулів Explorer, Knowledge Flow, Experimenter, Command-Line Interface?

1.4.4 Опишіть формат arff файлу. Наведіть приклади.

1.4.5 З чого складається процес попередньої обробки даних?

1.4.6 Які Ви знаєте способи усунення пропущених значень у наборі даних?

Лабораторна робота №2

Попередня обробка даних

***Мета роботи*:** Навчитись здійснювати попередню обробку даних засобами середовища інтелектуального аналізу даних WEKA.

1.1 Теоретичні відомості

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.

Розмах варіації - різниця між максимальним і мінімальним значенням вибірки:

$$R=Xmax – Xmin$$

З одного боку, показник розмаху може бути цілком інформативним і корисним. З іншого боку, розмах може бути дуже широким і не мати практичного сенсу, тому що залежить лише від двох спостережень. Таким чином, розмах варіації дуже нестійка (неробастна) величина.

У статистиці для аналізу вибірки часто вдаються до іншого показника варіації – міжквартильного розмаху (Interquartile Range (IQR)). Міжквартильний розмах – це різниця між 3-м та 1-м квартилями. Перевагою цього показника є те, що він є робастним, тобто не залежить від аномальних відхилень.

Міжквартильний діапазон IQR повідомляє про діапазон, у якому лежить основна маса значень.

1. На відміну від розмаху, IQR вказує на те, де лежить більшість даних.

2. IQR може використовуватися для ідентифікації викидів даних.

3. Дає центральну тенденцію даних.

Є показники варіації, які враховують відразу всі значення, а не тільки окремі спостереження (типу максимуму або мінімуму). Одним з таких є середнє лінійне відхилення.

Екстремальні значення – це викиди, які знаходяться на верхній та нижній межах значень у вибірці.

Поняття екстремального значення дає можливість розділити викиди на дві групи:

- Викиди – трапляються час від часу.
- Екстремальні значення – трапляються дуже рідко.

Процеси попередньої обробки даних у середовищі WEKA реалізовано у вигляді фільтрів.

За допомогою фільтрів здійснюють:

- нормалізацію даних
- пошук викидів та екстремальних значень
- очищення даних
- заповнення пропущених значень
- підбір атрибутів
- і т.д.

1.2 Хід роботи

1.2.1 Визначити (і навести обчислення) свій варіант за формулою:

$$\text{num } mod\ 5 + 1,$$

де num – номер у журналі, *mod* – операція взяття остачі від ділення.

1.2.2 Для набору даних diabetes.arff (зі стандартних наборів даних WEKA) здійснити нормалізацію числових атрибутів до проміжку, вибраного на основі номеру варіанту із таблиці 2.1. Результати зберегти у вигляді файлу diabetes_normalized.arff, який здати разом зі звітом.

Таблиця 2.1 – Проміжки нормалізації

Варіант	**Проміжок**
1	[2, 5]
2	[8, 13]
3	[-7, -3]
4	[-16, -9]
5	[4, 10]

Зробити висновки щодо отриманих результатів.

1.2.3 Для нормалізованих у попередньому завданні атрибутів даних здійснити стандартизацію. Результати зберегти у вигляді файлу diabetes_standardized.arff, який здати разом зі звітом.

Зробити висновки щодо отриманих результатів. Порівняти властивості нормалізованих та стандартизованих даних. Зробити висновки щодо необхідності застосування обох процедур обробки.

1.2.4 Для набору даних labor.arff (зі стандартних наборів даних WEKA) здійснити дослідження на наявність викидів та екстремальних значень на основі показника варіації IQR (для значень множників викидів та екстремальних значень за замовчуванням). Вказати, по яких атрибутах зустрічаються аномальні значення і якого характеру. Результати зберегти у вигляді файлу labor_outliers.arff, який здати разом зі звітом.

1.2.5 На основі отриманих результатів досліджень на аномальні значення провести очищення даних від викидів та екстремальних значень, а також, від допоміжних атрибутів, створених у процесі роботи фільтрів. Результати зберегти у вигляді файлу labor_cleared.arff, який здати разом зі звітом.

Зробити висновки щодо зміни кількості екземплярів даних та їх характеру. Зробити висновки стосовно того, до чого можуть привести зміни значень множників викидів та екстремальних значень у застосованому фільтрі.

1.2.6 Для наборів даних cpu.arff та ionosphere.arff (зі стандартних наборів даних WEKA) здійснити дискретизацію нецільових атрибутів на рівні за шириною проміжки із налаштуваннями за замовчуванням. Порівняти

отримані результати із результатами дискретизації цих наборів даних на рівні за частотою проміжки із налаштуваннями за замовчуванням; вказати, для яких наборів даних налаштування за замовчуванням дають прийнятні результати за кожним з типів дискретизації. Для тих випадків, у яких отримані із налаштуваннями за замовчуванням результати є неприйнятними, здійснити підбір налаштувань для отримання прийнятних результатів. Зберегти прийнятні результати у вигляді файлів cpu_width.arff, cpu_frequency.arff, ionosphere_width.arff та ionosphere_frequency.arff, які здати разом зі звітом.

Зробити висновки щодо отриманих результатів у контексті впливу на них налаштувань фільтру дискретизації, зокрема щодо залежності якості результатів від заданої дослідником кількості проміжків. Зробити висновки стосовно того, до чого може привести неякісно реалізована дискретизація та які переваги для подальшого аналізу можуть бути отримані у випадку якісної реалізації цієї процедури.

1.2.7 Зробити висновки стосовно результатів використання середовища WEKA для попередньої обробки даних та про особливості його застосування щодо виявлення аномальних значень у наборах даних.

1.3 Зміст звіту

1. Титульна сторінка та тема роботи.
2. Мета роботи.
3. Короткі теоретичні відомості (до 1 сторінки).
4. Результати роботи.
5. Висновки

1.4 Контрольні запитання

1.4.1 Що таке інтелектуальний аналіз даних?

1.4.2 Для чого використовується середовище WEKA, які його можливості?

1.4.3 Опишіть формат arff файлу. Наведіть приклади.

1.4.4 На основі чого у WEKA відбувається попередня обробка даних?

1.4.5 Які Ви знаєте способи нормалізації значень у наборі даних?

1.4.6 Що таке IQR?

1.4.7 Чим викиди відрізняються від екстремальних значень атрибутів даних?

1.4.8 На які класи поділений набір фільтрів WEKA?

Лабораторна робота №3

Задача класифікації

***Мета роботи*:** Навчитись розв’язувати задачі класифікації даних засобами середовища інтелектуального аналізу даних WEKA.

1.1 Теоретичні відомості

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.

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), які вважають основою для розв’язання довільної задачі класифікації, оскільки більшість із них можуть бути використані, як для класифікації, так і для порівняння ефективності, наведено у списку 3.1.

Список 3.1:

- 0R чи Zero Rule класифікація (rules.ZeroR);
- класифікація за одним правилом 1R чи One Rule (rules.OneR);
- покриваючий метод PRISM (rules.Prism);
- наївна Баєсова класифікація (bayes.NaiveBayes);
- метод побудови дерев рішень CART (trees.simpleCart)
- метод побудови дерев рішень C4.5 (trees.J48);
- метод опорних векторів (functions.SMO);
- метод k найближчих сусідів kNN (lazy.IBk).

Для додавання деяких із наведених методів у WEKA, необхідно встановити пакети simpleEducationalLearningSchemes і simpleCART.

Параметри налаштування деяких із наведених алгоритмів наведено у таблиці 3.1.

Таблиця 3.1 – 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 кращий у випадку, коли менше класів і багато атрибутів. *useKernelEstimator* – для оцінки числових атрибутів використовувати оціночну функцію відмінну від нормального розподілу. *useSupervisedDiscretization* – використовувати дискретизацію з вчителем для перетворення числових атрибутів у номінальні.
J48	*binarySplits* – використовувати бінарний поділ на категоріальних атрибутах для побудови дерев. *collapseTree* – визначає чи буде зменшуватись обсяг дерева за рахунок усунення частин, які не зменшують похибки навчання. *confidenceFactor* – довірчий рівень, використовується для відсікання гілок (менші значення – сильніше відсікання). *minNumObj* – мінімальна кількість екземплярів у листку. *reducedErrorPruning* – який алгоритм відсікання гілок використовувати. *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 з метою згладжування дерева.
SMO	*buildCalibrationModels* – чи застосовувати калібруючі моделі до виходів (для належної оцінки ймовірностей). *c* – параметр складності C. *calibrator* – вибір калібруючої моделі. *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 екземплярів, дозволених у навчальному пулі. Додавання додаткових екземплярів понад цього значення призведе до видалення старих екземплярів. Значення 0 означає відсутність межі.

Для алгоритмів ZeroR та Prism параметрів, які налаштовуються, немає. Для методу CART, значення параметрів та принципи їх роботи аналогічні відповідникам у J48.

Інтерпретація результатів класифікації в WEKA:

Результати роботи класифікаторів у WEKA наведено у вікні «*Classifier output*».

Секція «*Run information*» містить наступну інформацію:

- метод класифікації (scheme);
- назва набору даних, на якому проводилося навчання (relation);
- кількість екземплярів у вихідній вибірці (instances);
- атрибути, які 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, які будують правила, будуть відображені отримані правила;
- для баєсівських класифікаторів будуть перелічені розраховані ймовірності для всіх можливих комбінацій атрибут-значення-клас;
- для класифікаторів, заснованих на побудові дерев, відображається текстове представлення отриманого дерева; в дужках навпроти кожного листка вказана кількість екземплярів, які до нього віднесені; якщо в листок потрапляють екземпляри декількох класів, через «/» буде вказана кількість екземплярів, 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 у налаштуваннях тестування класифікатора опція «*Output predictions*» має встановлене значення відмінне

від «Null». У ній для всіх екземплярів 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 екземплярів (Correctly and Incorrectly Classified Instances), загальна кількість екземплярів (Total Number of Instances);
- параметр Каппа (Kappa statistic);
- 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 помилок.

1.2 Хід роботи

1.2.1 Визначити (і навести обчислення) свій варіант за формулою:

$$\text{num } mod\ 3 + 1,$$

де num – номер у журналі, *mod* – операція взяття остачі від ділення.

1.2.2 На основі номеру варіанту обрати із таблиці 3.2 відповідний набір даних. У випадку наборів post-operative та zoo завантажити набір даних із репозитарію UC Irvine Machine Learning Repository, доступного за посиланням https://archive.ics.uci.edu/datasets/, у випадку набору bank – скористатись посиланням: https://github.com/bluenex/WekaLearningDataset/blob/master/bank/bank.arff.

Таблиця 3.2 – Набори даних

Варіант	**Назва набору**
1	post-operative
2	bank
3	zoo

Файл із набором даних, який використано для розв’язання наступних завдань, необхідно здати разом зі звітом.

1.2.3 Розв’язати задачу класифікації для заданого набору даних за допомогою всіх наведених у списку 3.1 (у теоретичних відомостях) алгоритмів. Для цього, змінюючи параметри налаштування алгоритмів, спробувати досягти найвищої якості навчання кожного з класифікаторів. При досягненні найвищої якості навчання зберегти відповідні моделі (для кожного з класифікаторів) у вигляді файлів, для подальшої роботи з ними. Ці файли необхідно здати разом зі звітом.

Включити результати роботи по кожному з класифікаторів у звіт (значення параметрів для найякіснішої моделі та результати класифікації, отримані на її основі).

Для класифікаторів, для яких можливі зміни параметрів, навести результати роботи ще 3-5 моделей із різними параметрами у вигляді, описаному вище.

1.2.4 Порівняти отримані моделі за допомогою модуля Experimenter, використовуючи збережені файли з моделями із попереднього пункту. Файл із результатами експерименту необхідно здати разом зі звітом.

Включити результати порівняння у звіт, використовуючи відображення відносно 3-5 методів, вибраних у якості Test base, а також, для значень Test base: Summary та Ranking.

1.2.5 Зробити розгорнуті висновки стосовно результатів порівняння та обґрунтувати, який із класифікаторів забезпечує найбільш якісний розв’язок задачі класифікації для заданого набору даних.

1.3 Зміст звіту

1. Титульна сторінка та тема роботи.
2. Мета роботи.
3. Короткі теоретичні відомості (до 1 сторінки).
4. Результати роботи.
5. Висновки

1.4 Контрольні запитання

1.4.1 У чому полягає задача класифікації? Наведіть практичний приклад.

1.4.2 Опишіть один із розглянутих методів класифікації.

1.4.3 Що таке навчання з вчителем і без вчителя? До якого типу належить задача класифікації?

1.4.4 Задача класифікації є описовою чи прогнозуючою і чому?

1.4.5 Навіщо потрібні дві вибірки: навчальна і тестова?

1.4.6 Які існують підходи для поділу вихідної вибірки на навчальну і тестову?

1.4.7 Як оцінити якість побудованої моделі класифікації?

1.4.8 Що таке матриця помилок? Як її інтерпретувати?

1.4.9 Що означають параметри «чутливість», «специфічність», «точність»? Як їх розрахувати?

1.4.10 Що таке параметр Каппа? Що він показує?

1.4.11 Що таке аналіз втрати-виграші? Для чого його застосовують?

1.4.12 Як порівняти роботу двох класифікаторів?

Лабораторна робота №4

Прогнозування. Задача регресії

***Мета роботи*:** Навчитись розв’язувати задачі регресії даних засобами середовища інтелектуального аналізу даних WEKA.

1.1 Теоретичні відомості

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.

Регресійні алгоритми (у дужках наведено назву в програмі WEKA, при цьому перше слово перед крапкою означає тип алгоритму і також вказує назву папки, в якій знаходиться метод), які вважають основою для розв’язання довільної задачі регресії, оскільки, більшість із них можуть бути використані, як для регресії, так і для порівняння ефективності, наведено у списку 4.1.

Список 4.1:

- 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).

Параметри налаштування наведених алгоритмів наведено у таблиці 4.1.

Таблиця 4.1 – Параметри налаштування регресійних алгоритмів

Метод	Параметр
LinearRegression	*attributeSelectionMethod* – метод відбору атрибутів. *eliminateColinearAttributes* – виключити корелюючі атрибути. *ridge* – штраф за великі значення коефіцієнтів регресії (регуляризація Тихонова).
M5P	*buildRegressionTree* – регресійне або модельне дерево. *minNumInstances* – мінімальна кількість екземплярів у листку. *saveInstances* – зберігати екземпляри у вузлах для візуалізації. *unpruned* – будувати дерево без обрізки *useUnsmoothed* – незгладжене прогнозування.
SMOreg	*c* – параметр складності C. *kernel* – функція ядра. *epsilon* – параметр точності (не змінювати). *filterType* – чи буде змінена початкова інформація і яким чином (нормалізація або стандартизація). *tolerance* – допустиме відхилення (не змінювати), *regOptimizer* – алгоритм навчання.
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 екземплярів, дозволених у навчальному пулі. Додавання додаткових екземплярів понад цього значення призведе до видалення старих екземплярів. Значення 0 означає відсутність межі.

Інтерпретація результатів регресії у WEKA відбувається аналогічно інтерпретації результатів класифікації.

1.2 Хід роботи

1.2.1 Визначити (і навести обчислення) свій варіант за формулою:

$$\text{num} \ mod\ 3 + 1,$$

де num – номер у журналі, *mod* – операція взяття остачі від ділення.

1.2.2 На основі номеру варіанту обрати із таблиці 4.2 відповідний набір даних і завантажити його із репозитарію UC Irvine Machine Learning Repository, доступного за посиланням https://archive.ics.uci.edu/datasets/. У випадку набору даних cpu-performance на етапі попередньої обробки видалити атрибут «ERP».

Таблиця 4.2 – Набори даних

Варіант	**Назва набору**
1	housing
2	auto-mpg
3	cpu-performance

Файл із набором даних, який використано для розв'язання наступних завдань, необхідно здати разом зі звітом.

1.2.3 Розв'язати задачу регресії для заданого набору даних за допомогою всіх наведених у списку 4.1 (у теоретичних відомостях) алгоритмів. Для цього, змінюючи параметри налаштування алгоритмів, спробувати досягти найвищої якості навчання кожної з моделей. При досягненні найвищої якості навчання зберегти відповідні моделі (для кожного з алгоритмів) у вигляді файлів, для подальшої роботи з ними. Ці файли необхідно здати разом зі звітом.

Включити результати роботи по кожному з алгоритмів у звіт (значення параметрів для найякіснішої моделі та результати регресійного аналізу, отримані на її основі).

Для усіх алгоритмів навести результати роботи ще 3-5 моделей із різними параметрами у вигляді, описаному вище.

1.2.4 Порівняти отримані моделі за допомогою модуля Experimenter, використовуючи збережені файли з моделями із попереднього пункту. Файл із результатами експерименту необхідно здати разом зі звітом.

Включити результати порівняння у звіт, використовуючи відображення відносно 2-4 методів, вибраних у якості Test base, а також, для значень Test base: Summary та Ranking.

1.2.5 Зробити висновки стосовно результатів порівняння та обґрунтувати, який із алгоритмів забезпечує найбільш якісний розв'язок задачі регресії для заданого набору даних. Дати обґрунтовані відповіді на наступні питання:

- Які з атрибутів (проаналізувати як конкретні атрибути, так і типи загалом) є найбільш значущими для передбачення значень цільового атрибуту, судячи з побудованих моделей? Чому?
- Як зміниться точність передбачення, якщо залишити лише значущі атрибути? Навести ілюстрований приклад.

1.3 Зміст звіту

1. Титульна сторінка та тема роботи.
2. Мета роботи.
3. Короткі теоретичні відомості (до 1 сторінки).
4. Результати роботи.
5. Висновки

1.4 Контрольні запитання

1.4.1 У чому полягає задача регресії? Наведіть практичний приклад.

1.4.2 Чим задача регресії схожа і чим відрізняється від задачі класифікації?

1.4.3 Що таке навчання із вчителем і без вчителя? До якого типу належить задача регресії?

1.4.4 Задача регресії є описовою чи прогнозуючою і чому?

1.4.5 Опишіть один із розглянутих методів, які вирішують задачу регресії.

1.4.6 Як оцінити якість побудованої моделі для задачі регресії?

Лабораторна робота №5

Задача кластеризації

***Мета роботи*:** Навчитись розв’язувати задачі кластеризації даних засобами середовища інтелектуального аналізу даних WEKA.

1.1 Теоретичні відомості

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.

Алгоритми кластеризації (у дужках наведено назву в програмі WEKA), які вважають відправною точкою для розв’язання довільної задачі кластеризації, оскільки, вони дають початкове (часто і остаточне) уявлення про набір даних та його властивості з точки зору кластеризації, наведено у списку 5.1.

Список 5.1:

- поділяючий метод кластеризації K-середніх (SimpleKMeans);
- ієрархічний метод кластеризації (HierarchicalClusterer)
- імовірнісний метод кластеризації EM (EM).

Параметри налаштування наведених алгоритмів подано у таблиці 1.

Таблиця 5.1 – Параметри налаштування алгоритмів кластеризації

Метод	Параметр
SimpleKMeans	*displayStdDevs* – відображати значення СКВ для числових атрибутів та кількість 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 екземплярів у вибірці. *seed* – зерно для рандомізації вибірки.
Hierarchical Clusterer	*distanceFunction* – функція відстані. *distanceIsBranchLength* – у дендрограмі, висота лінії, яка зв'язує кластери, буде показувати відстань між ними. *linkType* – тип зв'язку для розрахунку відстані між двома кластерами. *numClusters* – кількість кластерів. *printNewick* – виводити кластери у форматі Newick.
EM	*displayModelInOldFormat* – використовувати старий формат представлення моделі (у випадках великої кількості кластерів). *maxIterations* – максимальна кількість ітерацій алгоритму. *minStdDev* – мінімальне значення СКВ. *numClusters* – кількість кластерів (встановити значення -1 для автоматичного вибору кількості кластерів). seed – зерно для рандомізації вибірки.

Інтерпретація результатів кластеризації у WEKA.

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 пропущених значень атрибутів.

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

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 у жоден з кластерів (Unclustered instances).

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

1.2 Хід роботи

1.2.1 Усі набори даних у роботі входять, крім zoo та bank, до стандартного набору зразків даних WEKA. У випадку набору zoo завантажити набір даних із репозитарію UC Irvine Machine Learning Repository, доступного за посиланням https://archive.ics.uci.edu/datasets/ у випадку набору bank – скористатись посиланням: https://github.com/bluenex/WekaLearningDataset/blob/master/bank/bank.arff

1.2.2 Виконайте наступні завдання для набору даних bank.arff:

- 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 і як змінюються його значення?
- Для значення K=5 вкажіть:
  - скільки кластерів було створено;
  - скільки екземплярів потрапило в кожен з кластерів (вказати кількість і відсоток);
  - скільки ітерацій знадобилося для кластеризації даних;
  - складіть таблицю з характеристиками центроїдів.
- Для значення K=5 візуалізуйте результати кластеризації (по осі абсцис відкласти назву (номер) кластера, по осі ординат – номер екземпляра у 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 неодружені (unmarried)?
- у який кластер потрапило найбільше людей з машинами?
- 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?

· Запустіть алгоритм кластеризації EM та оцініть результати.

1.2.3 Виконайте наступні завдання для набору даних iris.arff:

1. Запустіть алгоритм кластеризації 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-means; для цього проведіть експерименти з різними значеннями параметру і порівняйте отримані результати.

1.2.4 Завантажте набір даних zoo.arff і виконайте наступні завдання:

1. Оберіть з вибірки частину тварин на власний розсуд (наприклад, ссавців);

2. Запустіть алгоритм Hierarchical Clusterer (тип тварини не використовувати в кластеризації, а назву за допомогою фільтру перетворити на рядковий тип – NominalToString);
3. Проекспериментуйте з налаштуванням алгоритму та візуалізуйте результати його роботи;
4. Оцініть, чи є логічний сенс у створюваних кластерах.

1.3 Зміст звіту

1. Титульна сторінка та тема роботи.
2. Мета роботи.
3. Короткі теоретичні відомості (до 1 сторінки).
4. Результати роботи.
5. Висновки

1.4 Контрольні запитання

1.4.1 У чому полягає задача кластеризації? Наведіть практичний приклад?

1.4.2 Що таке навчання з учителем і без учителя? До якого типу належить задача кластеризації?

1.4.3 Задача кластеризації є описовою або прогнозуючою і чому?

1.4.4 Чим визначається «схожість» об’єктів при вирішенні задачі кластеризації?

1.4.5 Що таке ієрархічна кластеризація?

1.4.6 Які є підходи до розрахунку відстані між кластерами?

1.4.7 Опишіть один з розглянутих методів, які вирішують завдання кластеризації.

1.4.8 Як оцінити якість побудованої моделі для задачі кластеризації?

Лабораторна робота №6

Пошук асоціативних правил

***Мета роботи*:** Навчитись розв’язувати задачі пошуку асоціативних правил у даних засобами середовища інтелектуального аналізу даних WEKA.

1.1 Теоретичні відомості

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.

Алгоритми пошуку асоціативних правил (у дужках наведено назву в програмі WEKA), які вважають основою для розв’язання довільної задачі пошуку асоціативних правил, наведено у списку 6.1.

Список 6.1:

- алгоритм Apriori;
- алгоритм FPGrowth.

Параметри налаштування наведених алгоритмів подано у таблиці 1.

Таблиця 6.1 – Параметри налаштування алгоритмів пошуку асоціативних правил

Метод	Параметр
Apriori	*car* – пошук класових (зі значенням цільового атрибута в правій частині) або звичайних асоціативних правил. *classIndex* – індекс цільового атрибута. Якщо встановлено значення -1, то 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* – чи прибирати ті колонки (атрибути), у яких всі значення відсутні. *significanceLevel* – рівень значущості (тільки для надійності). *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 заданої кількості правил. *lowerBoundMinSupport* – нижня межа порогу підтримки як частка кількості екземплярів. *maxNumberOfItems* – максимальна кількість екземплярів 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 (екземпляри), які містять задані об'єкти. *upperBoundMinSupport* – верхня межа мінімальної підтримки. Ітеративне зменшення підтримки починається з цього значення. *useORForMustContainList* – використовувати логічну зв'язку «або» замість «і» для списків обов'язкових елементів у транзакціях і правилах.

Інтерпретація результатів пошуку асоціативних правил у WEKA.

Секція «*Associator output*» містить інформацію про побудовану модель.

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

Для алгоритму FPGrowth буде вказана загальна кількість знайдених правил та задана кількість найкращих із них.

Представимо асоціативне правило у вигляді $X \rightarrow Y$.

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 ($N_{X \cup Y}$).

Підтримка (support) правила дорівнює:

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

Надійність (confidence) правила – це відношення підтримки всього правила до підтримки 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 надійності на підтримку правої 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 різниця між кількістю екземплярів, які покриваються правилом за умови статистичної залежності між його лівою і

правою частинами, та кількістю екземплярів, які покривалися б за умови їх статистичної незалежності.

Наприклад, нехай, ми маємо 1000 екземплярів у вибірці (N), при цьому ліва частина правила покриває 200 з них (N_X), права окремо покриває 100 (N_Y) і все правило покриває 50 (N_{XUY}). Частка екземплярів, які покриваються правилом (підтримка) дорівнює 50/1000=0,05. Кількість екземплярів, які були б покриті правилом у випадку, коли ліва і права частини правила незалежні, дорівнює (200*100)/1000=20. Параметр Leverage для цього прикладу дорівнює 50-20=30, що у пропорції до загальної вибірки дорівнює 30/1000=0,03.

1.2 Хід роботи

1.2.1 Усі набори даних у роботі входять, окрім bank, до стандартного набору зразків даних WEKA. У випадку набору bank завантажити набір даних за посиланням:
https://github.com/bluenex/WekaLearningDataset/blob/master/bank/bank.arff

1.2.2 Виконайте наступні завдання для набору даних ‘vote.arff’.

- Запустіть алгоритм пошуку асоціативних правил Apriori.
- Яке значення для порогу підтримки було використано у побудованій моделі? Яке значення для порогу надійності було використано?
- Запишіть 10 найкращих знайдених правил (у вигляді списку або таблиці), вкажіть для них значення підтримки та надійності.
- Що позначають числа ліворуч і праворуч від стрілки в знайдених асоціативних правилах?
- Запустіть алгоритм пошуку асоціативних правил FPGrowth.
- Порівняйте списки десяти найкращих правил, отриманих двома алгоритмами. Поясніть відмінність в роботі двох алгоритмів.
- Запустіть алгоритм Apriori задавши значення car = true. Які асоціативні правила були отримані? Що знаходиться в правій частині знайдених асоціативних правил?

1.2.3 Вирішіть задачу пошуку асоціативних правил за допомогою двох алгоритмів зі списку 1 для набору даних ‘supermarket.arff’. У процесі – застосуйте різні метрики і оцініть, як це вплинуло на отримані правила. Проаналізуйте та поясніть знайдені правила. Вкажіть практичну цінність та застосування отриманих правил з огляду на галузь із якої отримано набір даних.

1.2.4 Спробуйте відшукати у наборі даних ‘bank.arff’ нові шаблони (асоціативні правила) за допомогою двох алгоритмів зі списку 1. Згенеруйте, також, правила, в правій частині яких буде знаходитися цільовий атрибут. У процесі – застосуйте різні метрики і оцініть, як це вплинуло на отримані правила. Проаналізуйте та поясніть усі знайдені правила. Вкажіть практичну цінність та застосування отриманих правил з огляду на галузь із якої отримано набір даних.

1.3 Зміст звіту

1. Титульна сторінка та тема роботи.
2. Мета роботи.
3. Короткі теоретичні відомості (до 1 сторінки).
4. Результати роботи.
5. Висновки

1.4 Контрольні запитання

1.4.1 У чому полягає задача пошуку асоціативних правил? Наведіть практичний приклад.

1.4.2 Що таке набір, який часто повторюється?

1.4.3 Як формуються правила зі знайдених наборів, які часто повторюються?

1.4.4 Опишіть алгоритм Apriori.

1.4.5 Що означають параметри support, confidence, lift, conviction та leverage, які застосовуються при пошуку асоціативних правил?

1.4.6 Опишіть алгоритм FPGrowth.

Рекомендовані джерела

1. Data Analytics Made Accessible: 2023 edition / A. Maheshwari. – eBook, 2023. – 314 p.
2. Методи інтелектуального аналізу та оброблення даних: навч. посібник / В.О. Гороховатський, І.С. Творошенко. – Харків: ХНУРЕ, 2021. – 92 с.
3. Аналіз даних та знань : навч. посібник / В. В. Литвин, Ю. В. Нікольський, В. В. Пасічник. – Львів : Магнолія, 2019. – 276 с.
4. Cluster Analysis (5th edition) / B. S. Everitt, S. Landau, M. Leese, D. Stahl. – Wiley, 2011. – 352 p.
5. Data Mining: Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in Data Management Systems) 4th Edition / I. H. Witten, F. Eibe, M. A. Hall, C. J. Pal. – The University of Waikato. (https://www.cs.waikato.ac.nz/ml/weka/book.html)
6. Weka 3 – Data Mining with Open Source Machine Learning Software in Java [Електронний ресурс] / Режим доступу: https://www.cs.waikato.ac.nz/ml/weka/
7. General Data Protection Regulation (GDPR) - Official Legal Text [Електронний ресурс] / Режим доступу: https://gdpr-info.eu/