Міністерство освіти і науки України
Запорізький національний університет

С. М. Іванов, Д. В. Очеретін

DATA MINING

Навчально-методичний посібник для здобувачів ступеня вищої освіти магістра спеціальності «Економіка» освітньо-професійної програми «Економічна кібернетика»

Затверджено
вченою радою ЗНУ
Протокол №____від 2024

Запоріжжя
2024

УДК: 004.67(075.8)
І-207

Іванов С. М., Очеретін Д. В. Data Mining : навчально-методичний посібник для здобувачів ступеня вищої освіти магістра спеціальності «Економіка» освітньо-професійної програми «Економічна кібернетика». Запоріжжя : Запорізький національний університет, 2024. 138 с.

Навчально-методичне видання розроблено з метою надання студентам необхідних знань і навичок із застосування методів Data Mining. У навчально-методичному посібнику наведено теоретичний матеріал до кожної теми, що розглядається, надано докладні роз'яснення змісту завдань лабораторних робіт. Крім цього, містяться індивідуальні завдання до лабораторних робіт для кожного студента. Для діагностики рівня засвоєння знань запропоновано питання для самоконтролю до кожної розглянутої теми.

Навчально-методичне видання з дисципліни «Data Mining» сприятиме оволодінню студентами теоретичних аспектів технології Data Mining, методів, можливістю їх застосування та використання інструментальних засобів Data Mining.

Зміст видання відповідає робочій програмі дисципліни «Data Mining». Навчально-методичний посібник призначений для здобувачів ступеня вищої освіти магістра спеціальності «Економіка» освітньо-професійної програми «Економічна кібернетика».

Рецензент
М.М. Іванов, доктор економічних наук, професор кафедри управління персоналом і маркетингу ЗНУ

Відповідальний за випуск
Н.К. Максишко, доктор економічних наук, професор кафедри економічної кібернетики ЗНУ

ЗМІСТ

ВСТУП

Робота з даними в бізнесі вимагає використання сучасних технологій. До таких технологій відносяться: машинне навчання (Machine Learning) та наука про дані (Data Science), штучний інтелект (AI) та обробка природньої мови (Natural Language Processing, NLP), візуалізація даних (Data Visualization) та створення дашбордів (Dashboarding), хмарні обчислення та великі дані (Big Data). Усі ці технології спрощують роботу бізнес-аналітиків та є важливою частиною їх діяльності. Машинне навчання та наука про дані дозволяють збирати, аналізувати та інтерпретувати великі обсяги даних, виявляти закономірності та передбачувати майбутні тренди. Використання шучного інтелекту дозволяє автоматизувати поточні завдання й надати цінні рекомендації на основі даних. Обробка природньої мови допомагає бізнес-аналітикам аналізувати та обробляти текстову інформацію, визначати настрої клієнтів, аналізувати відгуки тощо. Візуалізація даних дозволяє представити дані у вигляді зрозумілих графіків, діаграм та інтерактивних візуалізацій. А об'єднання даних в інформативні дашборди дозволяє переглядати головні показники діяльності та приймати оперативні рішення. Хмарні сервіси дозволяють бізнес-аналітикам працювати з даними віддалено й масштабувати свої обчислення. Big Data сприяє збору та аналізу великих обсягів даних, дозволяє виявити приховані тренди. Тому володіння бізнес-аналітиками технологією Data Mining є необхідною складовою для компетентностей фахівців з обробки даних.

Дисципліна «Data Mining» належить до циклу дисциплін вільного вибору студента в межах спеціальності 051 «Економіка». Цей курс спрямований на вивчення технологій, методів, інструментальних засобів та застосування Data Mining. Опис кожного методу супроводжується конкретним прикладом використання.

Метою викладання навчальної дисципліни «Data mining» є формування системи фундаментальних знань щодо процесу виокремлення, дослідження та моделювання великих обсягів даних для виявлення невідомих до цього структур з застосуванням статистичних та математичних методів.

Основними завданнями вивчення дисципліни «Data Mining» є:

– оволодіння студентами основними поняттями Data mining;

– вивчення класифікації методів та стадій Data mining;

– засвоєння завдань Data mining;

– оволодіння основними поняттями мови програмування для статистичної обробки даних R;

– формування у студентів навичок проводити пошук у великих обсягах даних неочевидних, об'єктивних та корисних на практиці закономірностей; навчитися застосовувати статистичні та кібернетичні методи Data mining.

Згідно з вимогами освітньої програми студенти повинні досягти таких результатів навчання (компетентностей):

– здатність визначати й розв’язувати складні економічні задачі та проблеми, приймати відповідні аналітичні та управлінські рішення у сфері економіки або у процесі навчання, що передбачає проведення досліджень та/або здійснення інновацій за невизначених умов та вимог;

– здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності);

– здатність проводити дослідження на відповідному рівні;

– здатність до професійної комунікації в сфері економіки іноземною мовою;

– здатність збирати, аналізувати та обробляти статистичні дані, науково-аналітичні матеріали, які необхідні для розв’язання комплексних економічних проблем, робити на їх основі обґрунтовані висновки;

– здатність використовувати сучасні інформаційні технології, методи та прийоми дослідження економічних та соціальних процесів, адекватні встановленим потребам дослідження;

– здатність формулювати професійні задачі в сфері економіки та розв’язувати їх, обираючи незалежні напрями і відповідні методи для їх розв’язання, беручи до уваги наявні ресурси;

– здатність застосовувати науковий підхід до формування та виконання ефективних проєктів у соціальноекономічній сфері;

– здатність планувати і виробляти проєкти у сфері економіки, здійснювати її інформаційне, методичне, матеріальне, фінансове та кадрове забезпечення;

– здатність знаходити, обробляти, інтерпретувати економічні дані та їх використовувати для дослідження процесів в сфері економічної діяльності на базі застосування математичних методів, моделей та комп’ютерних технологій;

– здатність моделювати проблеми управління та їх наслідки і пропонувати можливі шляхи вирішення із використанням методів економічної кібернетики та сучасних інформаційних технологій;

– здатність до проведення досліджень у сфері інформатики, спрямованих на пошук інноваційного використання нових та існуючих комп’ютерних технологій в сфері економічної діяльності.

Курс передбачає тісний зв’язок з усіма дисциплінами, що вивчалися на рівні бакалаврату, зокрема з такими навчальними дисциплінами, як: «Інформаційні технології в управлінні економічними системами», «Прогнозування соціально-економічних процесів», «Економетрія».

Після вивчення курсу «Інформаційні технології в управлінні економічними системами» студент повинен володіти теоретичними основами інформатики, вміти працювати з основними видами функцій, а також із масивами даних у програмному забезпеченні Microsoft Excel, мати навички використання прикладних систем оброблення економічних даних для дослідження соціально-економічних систем.

Після вивчення курсу «Прогнозування соціально-економічних процесів» студент повинен володіти методами короткострокового прогнозування при аналізі тенденцій розвитку соціально-економічних процесів, визначення складових часового ряду (тренд, сезонність), мати навички визначення якості та точності прогнозів.

Після вивчення курсу «Економетрія» студент повинен знати етапи побудови економетричних моделей, вміти застосовувати їх у практиці управління економічними процесами, вміти застосовувати кількісні та якісні методи аналізу, прогнозування соціально-економічних процесів.

Курс передбачає тісний зв'язок із такими навчальними дисциплінами магістерського рівня, як: «Аналіз та моделювання соціально-економічних систем», «Методологія наукових досліджень в інформаційній економіці», «Професійно-орієнтований практикум іноземною мовою».

Після вивчення курсу «Аналіз та моделювання соціально-економічних систем» студент повинен володіти сучасними математичними моделями та методами аналізу соціально-економічних систем, що становлять основу кількісного обґрунтування управлінських рішень та сприяють підвищенню їх якості.

Після вивчення курсу «Методологія наукових досліджень в інформаційній економіці» студент повинен володіти знаннями з основ методології, методів і понять наукового дослідження; отримати практичні навички і уміння застосування методів проведення наукового дослідження.

Після вивчення курсу «Професійно-орієнтований практикум іноземною мовою» студент повинен володіти практичними навичками спілкування іноземною мовою в науковій та професійній діяльності, вміти розуміти та інтерпретувати інформацію з міжнародних науково-метричних баз та видань.

У навчально-методичному виданні розглядаються відмінності Data Mining від класичних статистичних методів аналізу, розглядаються типи закономірностей Data Mining (асоціація, класифікація, послідовність, кластеризація, прогнозування). Описується область застосування Data Mining. Детально розглядаються методи Data Mining: нейронні мережі, дерева рішень, методи пошуку асоціативних правил, класифікації та кластеризації, аналізу часових рядів тощо. Знайомство з кожним методом проілюстровано рішенням практичної задачі за допомогою інструментального засобу, що використовує технологію Data Mining. Викладаються основні поняття мови програмування для статистичної обробки даних R.

Запропоноване авторами видання сприятиме набуттю студентами практичних навичок щодо аналізу великих обсягів даних та його використанню в управлінській діяльності.

ЗМІСТОВИЙ МОДУЛЬ 1. ОСНОВНІ ПОНЯТТЯ ПРО DATA MINING

Тема 1. Data Mining як мультидисциплінарна галузь

Мета: ознайомитися з основними поняттями «Data Mining», статистики та штучного інтелекту.

План

1.1 Основні поняття Data Mining
1.2 Мультидисциплінарна область Data Mining
1.3 Data Mining як частина ринку інформаційних технологій
1.4 Особливості використання Data Mining

Основні поняття

Data Mining. Статистика. Машинне навчання. Штучний інтелект. Аналітична система.

1.1 Основні поняття Data Mining

Розвиток інструментів, що ключають різноманітні методи для обробки інформації, призвів до появи терміну «Data Mining», яким користуються математики та статистики.

Термін «Data Mining» отримав свою назву з двох понять: data – пошуку цінної інформації у великій базі даних та mining – видобутку гірничої руди. Обидва процеси вимагають або просіювання величезної кількості сирого матеріалу, або розумного дослідження та пошуку шуканих цінностей.

Термін «Data Mining» часто перекладається як видобуток даних, отримання інформації, розкопка даних, інтелектуальний аналіз даних, засоби пошуку закономірностей, вилучення знань, аналіз шаблонів, «вилучення зерен знань з гір даних», розкопка знань у базах даних, інформаційна проходка даних, промивання даних. Поняття «виявлення знань у базах даних» (Knowledge Discovery in Databases, KDD) можна вважати синонімом Data Mining.

Поняття «Data Mining» з'явилося у 1978 році та набуло високої популярності в сучасному трактуванні приблизно з першої половини 1990-х років. До цього часу обробка та аналіз даних здійснювалися в рамках прикладної статистики, при цьому в основному вирішувалися завдання обробки невеликих баз даних.

Про популярність Data Mining говорить і той факт, що результат пошуку книжок англійською мовою з попереднім або повним переглядом з Data Mining в пошуковій системі Google за останні п'ять років складав 1,8 мільйонів записів.

Data Mining – мультидисциплінарна галузь, що виникла та розвивається на базі таких наук як прикладна статистика, розпізнавання образів, штучний інтелект, теорія баз даних тощо (рис. 1.1).

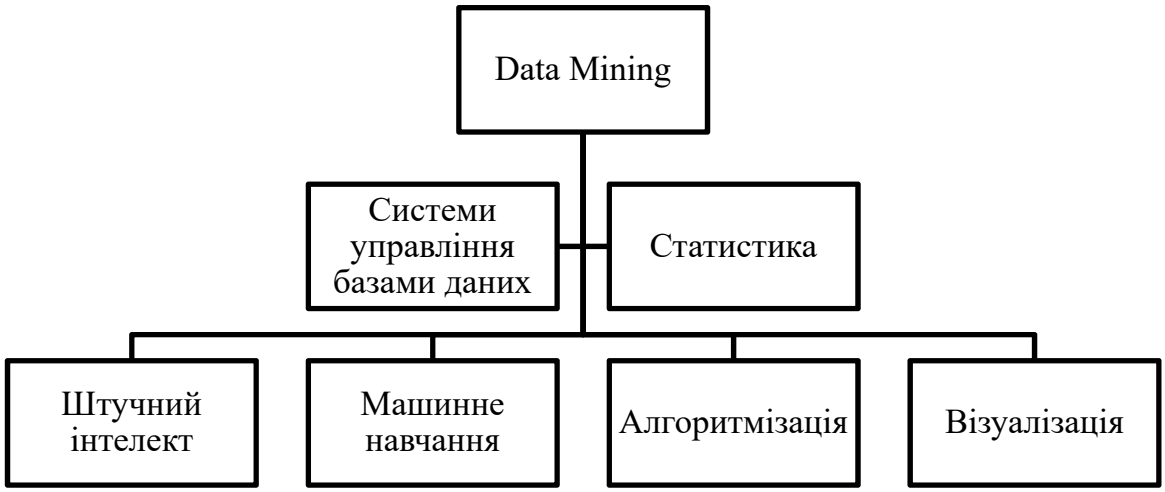

Рисунок 1.1 - Data Mining як мультидисциплінарна галузь

1.2 Мультидисциплінарна галузь Data Mining

Статистика – це наука про методи збору даних, їх обробки та аналізу для виявлення закономірностей, що властиві досліджуваному явищу.

Статистика є сукупністю методів планування експерименту, збору даних, їх подання та узагальнення, а також аналізу та отримання висновків на основі цих даних. Статистика оперує даними, отриманими в результаті спостережень чи експериментів.

Машинне навчання можна охарактеризувати як процес здобуття програмою нових знань. Американський дослідник Том Мітчелл у 1996 році дав таке визначення: «Машинне навчання – це наука, яка вивчає комп'ютерні алгоритми, що автоматично покращуються під час роботи». Єдиного визначення машинного навчання на сьогодні немає. Одним із найбільш популярних прикладів алгоритму машинного навчання є нейронні мережі.

Штучний інтелект – науковий напрям, у межах якого ставляться і вирішуються завдання апаратного чи програмного моделювання видів людської діяльності, які традиційно вважаються інтелектуальними.

Термін інтелект (intelligence) походить від латинського intellectus, що означає ум, глузд, розум, розумові здібності людини.

Відповідно, штучний інтелект (AI, Artificial Intelligence) тлумачиться як властивість автоматичних систем брати на себе окремі функції інтелекту людини. Штучним інтелектом називають властивість інтелектуальних систем виконувати творчі функції, які зазвичай вважаються прерогативою людини.

Кожен із напрямків, що сформували Data Mining, має свої особливості (табл.. 1.1).

Таблиця 1.1 – Порівняльні характеристики статистики, машинного навчання та Data Mining

Статистика	Машинне навчання	Data Mining
➢ більше, ніж Data Mining, базується на теорії; ➢ більше зосереджується на перевірці гіпотез	➢ більш евристично; ➢ концентрується на покращенні роботи агентів навчання	➢ інтеграція теорії та евристик; ➢ концентрація на єдиному процесі аналізу даних, який включає очистку даних, навчання, інтеграцію та візуалізацію результатів

Поняття Data Mining тісно пов’язане з технологіями баз даних та поняттям «дані».

Виникнення та розвиток Data Mining обумовлено різними факторами, основні серед яких є такі:

- вдосконалення апаратного та програмного забезпечення;
- вдосконалення технологій зберігання та запису даних;
- накопичення великої кількості ретроспективних даних;
- вдосконалення алгоритмів обробки інформації.

Data Mining – це процес підтримки прийняття рішень, заснований на пошуку даних прихованих закономірностей (шаблонів інформації). Один із засновників цього напряму Григорій Піатецький-Шапіро (Gregory Piatetsky-Shapiro) визначає технологію Data Mining як «процес виявлення в сирих даних раніше невідомих, нетривіальних, практично корисних і доступних інтерпретації знань, які необхідні для прийняття рішень у різних сферах людської діяльності».

Сутність та мету технології Data Mining можна охарактеризувати як технологію, що призначена для пошуку у великих обсягах даних неочевидних, об’єктивних та корисних на практиці закономірностей.

Неочевидних закономірності – знайдені закономірності не виявляються стандартними методами обробки інформації або експертним шляхом.

Об’єктивні закономірності – виявлені закономірності повністю відповідатимуть дійсності, на відміну від експертної думки, яка завжди є суб’єктивною.

Практично корисні закономірності – висновки мають конкретне значення, якому можна знайти практичне застосування.

Знання – це сукупність відомостей, яка утворює цілісний опис, що відповідає деякому рівню поінформованості про описуване питання, предмет, проблему тощо.

Використання знань (knowledge deployment) означає дійсне застосування знайдених знань задля досягнення конкретних переваг (наприклад, у конкурентній боротьбі за ринок).

За визначенням SAS Institute Data Mining – це процес виділення, дослідження та моделювання великих обсягів даних для виявлення невідомих до цього структур (patterns) з метою досягнення переваг у бізнесі.

Gartner Group визначає Data Mining як процес, метою якого є виявити нові значущі кореляції, зразки та тенденції в результаті просіювання значного обсягу даних, що зберігаються, з використанням методик розпізнавання зразків плюс застосування статистичних і математичних методів.

В основу технології Data Mining покладено концепцію шаблонів (patterns), які є закономірностями, властивими підвибіркам даних, що можуть бути виражені у формі, зрозумілій людині. Метою пошуку закономірностей є уявлення даних у вигляді, що відображає шукані процеси. Побудова моделей прогнозування також є метою пошуку закономірностей.

1.3 Data Mining як частина ринку інформаційних технологій

Агентство Gartner Group, яке займається аналізом ринків інформаційних технологій, у 1980-х роках запровадило, а у 1996 році уточнило визначення терміну "Business Intelligence" (BI), діловий інтелект або бізнес-інтелект. ***Business Intelligence*** – програмні засоби, що функціонують у рамках підприємства та забезпечують функції доступу та аналізу інформації, що знаходиться у сховищі даних, а також забезпечують прийняття правильних та обґрунтованих управлінських рішень. Цей термін запропонований для опису різних концепцій та методів, які покращують бізнес-рішення шляхом використання систем підтримки прийняття рішень.

Поняття BI поєднує в собі різні засоби та технології аналізу та обробки даних масштабу підприємства.

На основі цих засобів створюються BI-системи, мета яких – підвищити якість інформації для прийняття управлінських рішень.

BI-системи також відомі під назвою Систем підтримки прийняття рішень (СППР, DSS, Decision Support System). Ці системи перетворюють дані на інформацію, на основі якої можна приймати рішення.

Gartner Group визначає склад ринку систем Business Intelligence як набір програмних продуктів наступних класів:

– засоби побудови сховищ даних (data warehousing);

– системи оперативної аналітичної обробки (OLAP);

– інформаційно-аналітичні системи (Enterprise Information Systems, EIS);

– засоби інтелектуального аналізу даних (data mining);

– інструменти для виконання запитів та побудови звітів (query and reporting tools).

Класифікація Gartner базується на методі функціональних завдань, де програмні продукти кожного класу виконують певний набір функцій чи операцій із використанням спеціальних технологій.

Відповідно до посібника з придбання продуктів Data Mining (Enterprise Data Mining Buying Guide) компанії Aberdeen Group Data Mining є технологією видобутку корисної інформації з баз даних, результати застосування якої

залежать від рівня підготовки даних у більшій мірі, ніж від інструментів, що використовуються.

1.4 Особливості використання Data Mining

ІТ-команди захопилися міфом про те, що засоби Data Mining прості у використанні. Передбачається, що достатньо запустити такий інструмент на терабайтній базі даних, і миттєво з'явиться корисна інформація. Насправді успішний Data Mining проект вимагає розуміння суті діяльності, знання даних та інструментів, а також процесу аналізу даних.

Перш ніж використовувати технологію Data Mining, необхідно ретельно проаналізувати її проблеми, обмеження та критичні питання, пов'язані з нею, а також зрозуміти, чого ця технологія не може:

– технологія Data Mining не може замінити аналітика й дати відповіді на ті питання, що не були задані;

– оскільки технологія Data Mining є мультидисциплінарною областю, для розробки програми, що включає Data Mining, необхідно задіяти фахівців з різних областей, а також забезпечити їхню якісну взаємодію.

Різні інструменти Data Mining мають різний інтерфейс та потребують певної кваліфікації користувача. Тому програмне забезпечення має відповідати рівню підготовки користувача. Використання Data Mining має бути нерозривно пов'язане із підвищенням кваліфікації користувача. Важливо, щоб фахівці з Data Mining добре зналися на бізнесі.

Необхідний ретельний вибір моделі та інтерпретація залежностей чи шаблонів, які виявлені. Тому робота з такими засобами потребує тісної співпраці між експертом у предметній галузі та фахівцем із інструментів Data Mining. Побудовані моделі мають бути грамотно інтегровані в бізнес-процеси для можливості оцінки та оновлення моделей. Останнім часом системи Data Mining постачаються як частина технології сховищ даних.

Успішний аналіз потребує якісної попередньої обробки даних. За твердженням аналітиків та користувачів баз даних цей процес може зайняти до 80% всього Data Mining-процесу.

Таким чином, щоб технологія працювала на себе, знадобиться багато зусиль та часу, які йдуть на попередній аналіз даних, вибір моделі та її коригування.

За допомогою Data Mining можна відшукувати дійсно дуже цінну інформацію, яка незабаром дасть великі дивіденди у вигляді фінансової та конкурентної вигоди. Щоб уникнути помилкових відкриттів та статистично недостовірних результатів необхідна перевірка адекватності отриманих моделей на тестових даних.

Якісна Data Mining-програма може коштувати досить дорого для компанії. Варіантом служить придбання вже готового рішення із попередньою перевіркою його використання, наприклад, на демо-версії з невеликою вибіркою даних.

Засоби Data Mining, на відміну від статистичних, теоретично не вимагають наявності строго певної кількості ретроспективних даних. Ця особливість може стати причиною виявлення недостовірних, хибних моделей і, як наслідок, прийняття на їх основі неправильних рішень. Потрібно здійснювати контроль статистичної значущості виявлених знань.

Традиційні методи аналізу даних (статистичні методи) та OLAP в основному орієнтовані на перевірку заздалегідь сформульованих гіпотез (verification-driven data mining) та на «грубий» розвідувальний аналіз, що становить основу оперативної аналітичної обробки даних (OnLine Analytical Processing, OLAP), в той час як одне з основних положень Data Mining – пошук неочевидних закономірностей. Інструменти Data Mining можуть знаходити такі закономірності самостійно і самостійно будувати гіпотези про взаємозв'язки. Оскільки саме формулювання гіпотези щодо залежностей є найскладнішим завданням, перевага Data Mining у порівнянні з іншими методами аналізу є очевидною.

Більшість статистичних методів виявлення взаємозв'язків у даних використовують концепцію усереднення за вибіркою, що призводить до операцій над неіснуючими величинами, тоді як Data Mining оперує реальними значеннями.

Перспективами Data Mining можуть бути такі напрями розвитку:

– виділення типів предметних областей із відповідними їм евристиками, формалізація яких полегшить вирішення відповідних завдань Data Mining, які стосуються цих областей;

– створення формальних мов та логічних засобів, за допомогою яких буде формалізовано міркування та автоматизація яких стане інструментом вирішення завдань Data Mining у конкретних предметних галузях;

– створення методів Data Mining, здатних як витягувати з даних закономірності, так й формувати деякі теорії, що спираються на емпіричні дані;

– подолання суттєвого відставання можливостей інструментальних засобів Data Mining від теоретичних досягнень у цій галузі.

Якщо розглядати майбутнє Data Mining у короткостроковій перспективі, то очевидно, що розвиток цієї технології найбільш спрямований до областей, пов'язаних із бізнесом. У короткостроковій перспективі продукти Data Mining можуть стати такими ж звичайними та необхідними, як електронна пошта, і, наприклад, використовуватись користувачами для пошуку найнижчих цін на певний товар або найдешевших квитків.

У довгостроковій перспективі майбутнє Data Mining є справді захоплюючим – це може бути пошук інтелектуальними агентами як нових видів лікування різних захворювань, так і нового розуміння природи Всесвіту.

Однак Data Mining містить у собі і потенційну небезпеку – адже все більша кількість інформації стає доступною через всесвітню мережу, в тому числі і відомості приватного характеру, і все більше знань можна здобути з неї. Отримання компаніями персональної інформації клієнтів регулюється законодавством Європейського Союзу та окремих країн.

Дослідження відзначають, що є як успішні рішення, що використовують Data Mining, і невдалий досвід застосування цієї технології. Області, де застосування технології Data Mining, швидше за все, будуть успішними, мають такі особливості:

– вимагають рішень, що ґрунтуються на знаннях;

– мають навколишнє середовище, що змінюється;

– мають доступні, достатні та значущі дані;

– забезпечують високі дивіденди від правильних рішень.

Досить довго дисципліна Data Mining не визнавалася повноцінною самостійною областю аналізу даних. На сьогоднішній день визначилося кілька точок зору на Data Mining: це те, що відволікає увагу від класичного аналізу даних, або Data Mining приймається як альтернатива традиційному підходу до аналізу. Є й середина, де розглядається можливість спільного використання сучасних досягнень у сфері Data Mining та класичного статистичного аналіза даних.

Технологія Data Mining постійно розвивається, привертає до себе дедалі більший інтерес як з боку наукового світу, так і з боку застосування досягнень технології в бізнесі.

Питання для самоконтролю

1. Назвіть кілька визначень поняття Data Mining.
2. Опишіть приклади застосування Data Mining в різних сферах економіки.
3. Назвіть фактори що обумовили розвиток Data Mining.
4. Назвіть напрями розвитку Data Mining.
5. Назвіть області, де застосовуються технології Data Mining.

Лабораторне заняття №1

Тема: Завантаження даних та їх візуалізація на мові R

Мета роботи: ознайомити студентів з меню програм R та RStudio та навчити практичним навичкам використання бібліотек для аналітичної діяльності.

Завдання. Для обраного соціально-економічного показника з бази даних Світового банку (https://data.worldbank.org/indicator) для обраної країни виконати п. 5-9 лабораторної роботи та оформити звіт у системі Moodle.

Хід роботи.

1. Ознайомтесь з основними меню програм R та RStudio

За допомогою команд library() та dir(.libPaths()) перегляньте, які пакети встановлені у програмі

2. У разі необхідності встановіть бібліотеки *dplyr*, *readr*, *ggplot2* та інші, бібліотеки, які необхідні для роботи названих бібліотек.

Встановлення бібліотеки можна зробити у меню *Inslall package(s),* обравши необхідний репозиторий (за замовчуванням *CRAN*).

3. Підключіть вказані бібліотеки за допомогою функції library(), вказавши у дужках назву необхідної бібліотеки.

4. Встановіть бібліотеку *wbstats*, яка дозволяє завантажувати дані Світового банку.

5. Для прикладу поглянемо на зміну частки населення України, яка використовує власний Інтернет (*Individuals using the Internet (% of population)*). Знайдемо сам показник, а потім, дізнавшись його код, висавимо інші параметри – країну та період часу, який нас цікавить:

```
wbsearch(pattern = "Individuals using the Internet", fields="indicator")
 # шукаємо точну назву (код) показника

      indicatorID                                indicator
8540 IT.NET.USER.ZS Individuals using the Internet (% of population)

Internet <- wb(country = c("UA"), indicator = c("IT.NET.USER.ZS"), startdate = 20
00, enddate = 2023)
View(Internet)
```

6. Ознайомтесь зі структурою об’єкту Internet, використовучи команду str():

```
str(Internet)
‘data.frame’:    22 obs. of  7 variables:
 $ iso3c      : chr  "UKR" "UKR" "UKR" "UKR" ...
 $ date       : chr  "2021" "2020" "2019" "2018" ...
 $ value      : num  79.2 75 70.1 62.6 58.9 ...
 $ indicatorID: chr  "IT.NET.USER.ZS" "IT.NET.USER.ZS" "IT.NET.USER.ZS" "IT.
NET.USER.ZS" ...
 $ indicator  : chr  "Individuals using the Internet (% of population)" "Individuals usi
ng the Internet (% of population)" "Individuals using the Internet (% of population)"
"Individuals using the Internet (% of population)" ...
 $ iso2c      : chr  "UA" "UA" "UA" "UA" ...
 $ country    : chr  "Ukraine" "Ukraine" "Ukraine" "Ukraine" ...
```

7. Підрахуйте кількість стовпців (команда ncol()),рядків (команда nrow()) та виведіть назви стовпців (команда colnames())

```
ncol(Internet)
[1] 7

nrow(Internet)
[1] 22
```

```
colnames(Internet)
[1] "iso3c"      "date"       "value"      "indicatorID" "indicator"   "iso2c"
[7] "country"
```

8. Виведіть на екран перші 6 рядків файлу (команда head())

```
head(Internet)
  iso3c date    value    indicatorID                                indicator iso2c
2   UKR 2021 79.21829 IT.NET.USER.ZS Individuals using the Internet (% of popula
tion)    UA
3   UKR 2020 75.03791 IT.NET.USER.ZS Individuals using the Internet (% of popula
tion)    UA
4   UKR 2019 70.12484 IT.NET.USER.ZS Individuals using the Internet (% of popula
tion)    UA
5   UKR 2018 62.55316 IT.NET.USER.ZS Individuals using the Internet (% of popula
tion)    UA
6   UKR 2017 58.88948 IT.NET.USER.ZS Individuals using the Internet (% of popula
tion)    UA
7   UKR 2016 53.00097 IT.NET.USER.ZS Individuals using the Internet (% of popula
tion)    UA
  country
2 Ukraine
3 Ukraine
4 Ukraine
5 Ukraine
6 Ukraine
7 Ukraine
```

9. Побудуйте графік динаміки частки користувачів Інтернет у форматі стовпчикової діаграми, лінійного графіка та графіку розсіювання.

Для візуалізації даних будемо використовувати бібліотеку *ggplot2*. Для побудови графіків використовується функція *ggplot()*. Після виконання коду ви побачите графік у вкладці *Plots* у нижній правій панелі RStudio.

Першим аргументом цієї функції є набір даних.

Далі ми вказуємо змінні з набору даних як параметр *aesthetic*, які будуть відображатися, наприклад по осях X та Y.

Наступним кроком ми додаємо ще один рівень (об'єднавши їх знаком +) щоб задати *geometric* об'єкт. Наприклад, для стовпчикової діаграми *geom_col* (рис.1.2), для лінійного графіка *geom_line* (рис.1.3), для графіка розсіювання це *geom_point* (рис.1.4).

```
ggplot(Internet, aes(x=date,y=value))+geom  col(fill="lightblue", col="gray")
+ylab('Відсоток від населення')
```

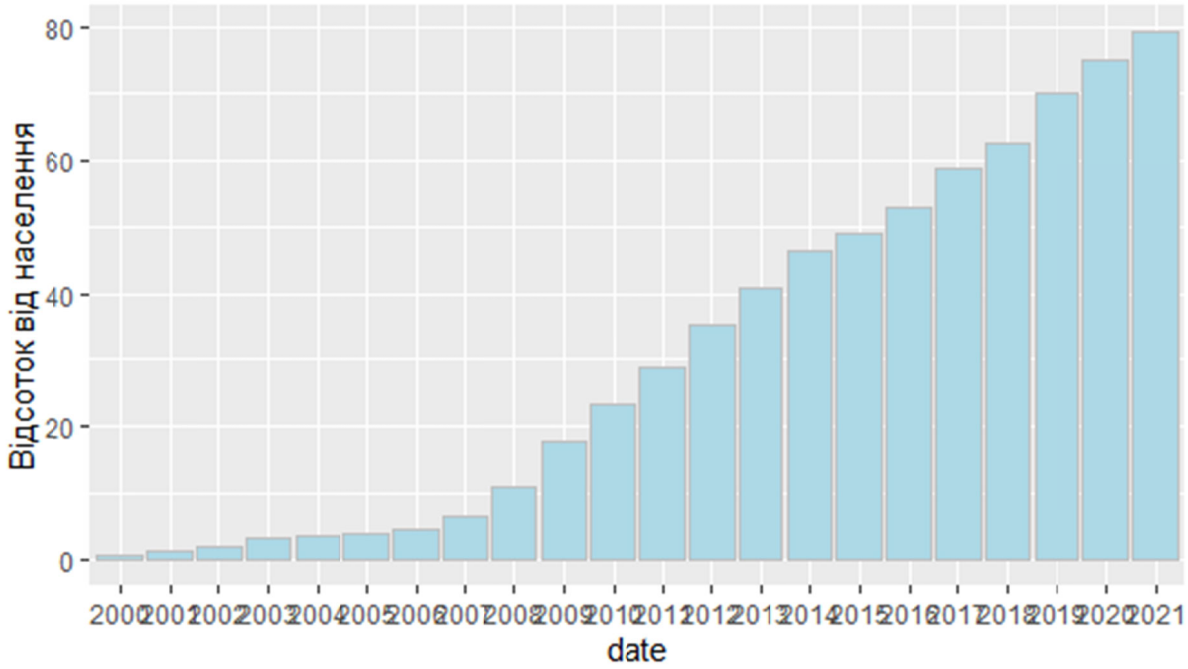

Рисунок 1.2 - Стовпчикова діаграма

ggplot(Internet, aes(date,value, group=1))+geom_line()+ylab('Відсоток від населення')

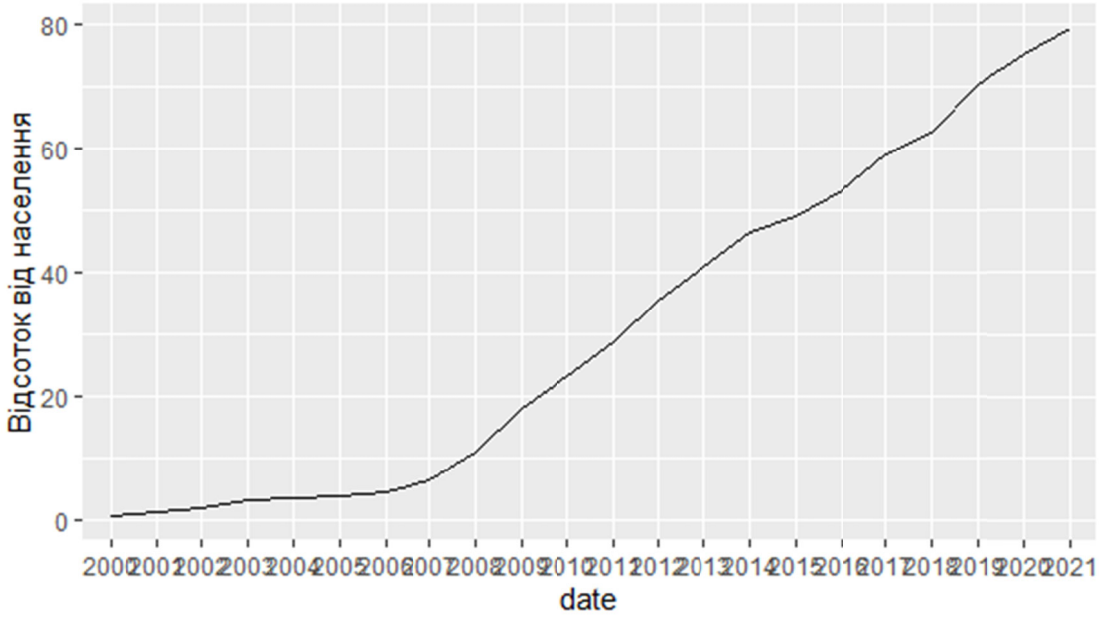

Рисунок 1.3 - Лінійний графік

ggplot(Internet, aes(x=date,y=value))+geom_point()+ylab('Відсоток від населення')

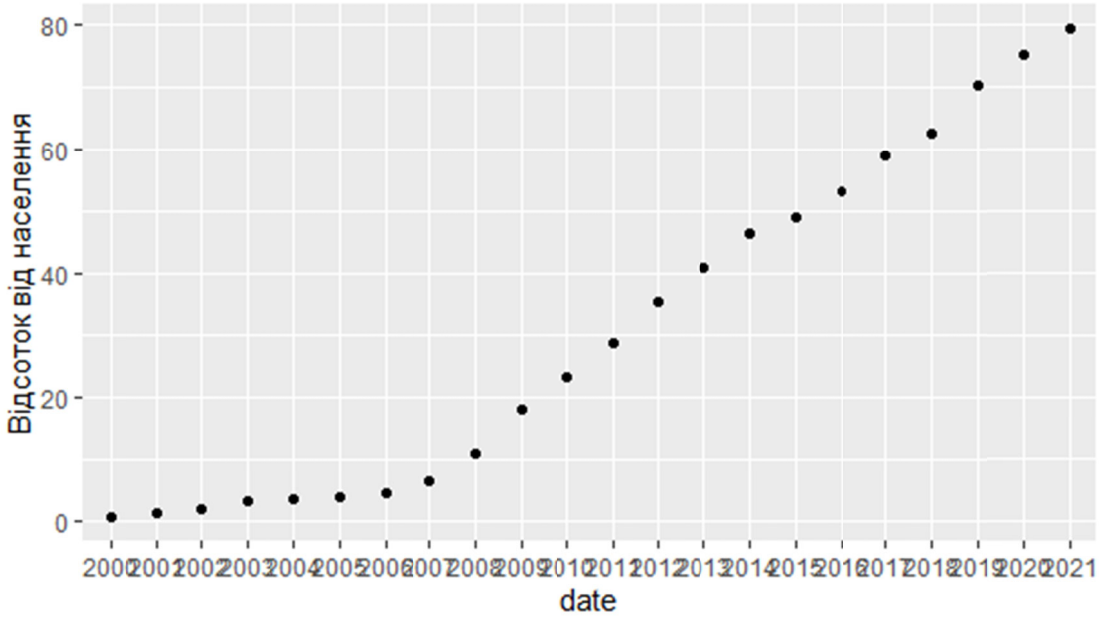

Рисунок 1.4 - Графік розсіювання

Тема 2. Набір даних та їх атрибутів

Мета: ознайомитися з поняттям данні, визначити їх властивості та атрибути.

План

2.1 Дані, набір даних та їх атрибути
2.2 Формати зберігання даних
2.3 Якісний аналіз даних із використанням Data Mining (DM)
2.4 Системи управління базами даних

Основні поняття

Змінна. Значення. Генеральна сукупність. Вибірка. Параметри. Статистики. Гіпотези. Вимірювання. Числові та символьні змінні. Дискретні та неперервні дані.

2.1 Дані, набір даних та їх атрибути

У широкому розумінні **дані – це факти, текст, графіки, картинки, звуки, аналогові або цифрові відео-сегменти**. Дані можуть бути отримані в результаті вимірювань, експериментів, арифметичних і логічних операцій. Дані повинні бути представлені у формі, придатній для зберігання, передачі й обробки. Іншими словами, дані – це необроблений матеріал, що надається постачальниками даних і використовується споживачами для формування інформації на основі даних.

У таблиці 2.1 представлена двовимірна таблиця, що представляє собою набір даних.

Таблиця 2.1 − Двовимірна таблиця «об’єкт-атрибут»

	Атрибути				
Об’єкти	Код клієнта	Вік	Сімейний статус	Прибуток	Клас
	1	19	неодр.	1234	1
	2	23	одр.	1222	1
	3	34	одр.	2700	1
	4	24	неодр.	2343	1
	5	26	одр.	1765	2
	6	32	розл.	2652	1
	7	19	неодр.	1200	2
	8	22	неодр.	1765	2
	9	40	одр.	1998	1
	10	43	розл.	4332	1

По горизонталі таблиці розташовуються атрибути об’єкта або його ознаки. По вертикалі таблиці – об’єкти. Об’єкт описується як набір атрибутів. Об’єкт також відомий як запис, випадок, приклад, рядок таблиці тощо.

Атрибут – властивість, що характеризує об’єкт. Наприклад: колір очей людини, температура води. Атрибут також називають змінною, полем таблиці, виміром, характеристикою.

Змінна (variable) – властивість або характеристика, загальна для всіх досліджуваних об’єктів, прояв якої може змінюватися від об’єкта до об’єкта.

Значення (value) змінної є проявом ознаки.

При аналізі даних, як правило, немає можливості розглянути всю сукупність об’єктів, що нас цікавить. Вивчення дуже великих обсягів даних є дорогим процесом, що вимагає великих затрат часу, а також неминуче призводить до помилок, пов’язаних із людським фактором.

Цілком достатньо розглянути деяку частину всієї сукупності, тобто вибірку, і отримати цікаву для нас інформацію на її підставі.

Однак розмір вибірки повинен залежати від різноманітності об’єктів, представлених у генеральній сукупності. У вибірці повинні бути представлені різні комбінації та елементи генеральної сукупності.

Генеральна сукупність (population) – вся сукупність досліджуваних об’єктів, що цікавить дослідника.

Вибірка (sample) – частина генеральної сукупності, певним способом відібрана з метою дослідження та отримання висновків про властивості та характеристики генеральної сукупності.

Параметри – числові характеристики генеральної сукупності.

Статистики – числові характеристики вибірки. Часто дослідження ґрунтуються на гіпотезах. Гіпотези перевіряються за допомогою даних.

Гіпотеза – припущення щодо параметрів сукупності об’єктів, яке має бути перевірено на її частині. Це частково обґрунтована закономірність знань, що служить або для зв’язку між різними емпіричними фактами, або для пояснення факту групи фактів.

Приклад гіпотези: між показниками тривалості життя та якістю харчування є зв’язок. У цьому випадку метою дослідження може бути пояснення змін конкретної змінної, в даному випадку – тривалості життя. Припустимо, існує гіпотеза, що залежна змінна (тривалість життя) змінюється залежно від деяких причин (якість харчування, спосіб життя, місце проживання тощо), які й є незалежними змінними.

Однак змінна першопочатково не є залежною або незалежною, вона стає такою після формулювання конкретної гіпотези. Залежна змінна в одній гіпотезі може бути незалежною в іншій.

Вимірювання – процес присвоєння чисел характеристикам досліджуваних об’єктів згідно певного правила.

У процесі підготовки даних вимірюється не сам об’єкт, а його характеристики.

Шкала – правило, відповідно до якого об’єктам присвоюються числа.

Багато інструментів Data Mining при імпорті даних з інших джерел пропонують вибрати тип шкали для кожної змінної та/або вибрати тип даних для вхідних і вихідних змінних (символьні, числові, дискретні та безперервні).

Користувачеві такого інструменту необхідно володіти цими поняттями.

Змінні можуть бути числовими даними або символьними.

Числові дані, своєю чергою, можуть бути дискретними і неперервними.

Дискретні дані є значеннями ознаки, загальне число яких скінченне або нескінченне, але може бути підраховане за допомогою натуральних чисел від одного до нескінченності.

Прикладом дискретних даних є тривалість маршруту тролейбуса (кількість варіантів тривалості скінченне): 10, 15, 25 хв.

Неперервні дані – дані, значення яких можуть набувати якого завгодно значення в деякому інтервалі. Вимірювання неперервних даних передбачає велику точність.

Приклад неперервних даних: температура, висота, вага, довжина тощо.

Шкали. Існує п’ять типів шкал вимірювань: номінальна, порядкова, інтервальна, відносна і дихотомічна.

Номінальна шкала (nominal scale) – шкала, яка містить тільки категорії; дані в ній не можуть упорядковуватися, з ними не можуть бути зроблені ніякі арифметичні дії.

Номінальна шкала складається з назв, категорій, імен для класифікації і сортування об’єктів або спостережень за деякою ознакою.

Приклад такої шкали: професії, місто проживання, сімейний стан.

Для цієї шкали застосовні тільки такі операції: дорівнює (=), не дорівнює (≠).

Порядкова шкала (ordinal scale) – шкала, в якій числа присвоюють об’єктам для позначення відносної позиції об’єктів, але не величини відмінностей між ними.

Шкала вимірювань дає можливість ранжувати значення змінних. Вимірювання ж у порядковій шкалі містять інформацію лише про порядок проходження величин, але не дозволяють сказати наскільки одна величина більше іншої, або наскільки вона менше іншої.

Приклад такої шкали: місце (1-ше, 2-ге, 3-є), яке команда отримала на змаганнях, номер студента в рейтингу успішності (1-й, 20-й), при цьому невідомо, наскільки один студент успішніше іншого, відомий лише його номер у рейтингу.

Для цієї шкали застосовуються тільки такі операції: дорівнює (=), не дорівнює (≠), більше (>), менше (<).

Інтервальна шкала (interval scale) **–** шкала, різниці між значеннями якої можуть бути обчислені, проте їх відношення не мають сенсу.

Ця шкала дозволяє знаходити різницю між двома величинами, має властивості номінальної та порядкової шкал, а також дозволяє визначити кількісну зміну ознаки.

Приклад такої шкали: температура води в морі вранці – 19 градусів, ввечері – 24, тобто вечірня на 5 градусів вище, але не можна сказати, що вона в 1,26 разів вище.

Номінальна і порядкова шкали є дискретними, а інтервальна шкала – неперервною. Вона дозволяє здійснювати точні вимірювання ознаки і виробляти арифметичні операції додавання, віднімання, множення, ділення.

Для цієї шкали застосовуються тільки такі операції: дорівнює (=), не дорівнює (≠), більше (>), менше (<), операції додавання (+) і віднімання (-).

Відносна шкала (ratio scale) – шкала, в якій є певна точка відліку і можливі відносини між значеннями шкали.

Приклад такої шкали: вага новонародженої дитини (4 кг і 3 кг). Перша в 1,33 рази важче.

Відносні та інтервальні шкали є числовими.

Для цієї шкали можуть бути застосовані тільки такі операції: дорівнює (=), не дорівнює (≠), більше (>), менше (<), операції додавання (+) і віднімання (-), множення (*) і ділення (/).

Дихотомічна шкала (dichotomous scale) – шкала, яка містить тільки дві категорії.

Приклад такої шкали: стать (чоловіча і жіноча).

Приклад використання різних шкал для вимірювань властивостей різних об’єктів, у даному випадку характеристик людей, наведено в таблиці 2.2.

Таблиця 2.2 − Множина вимірювань властивостей різних об’єктів

Номер об’єкту	**Професія (номінальна шкала)**	**Середній бал (інтервальна шкала)**	**Освіта (порядкова шкала)**
1	Слюсар	22	середня
2	Вчений	55	вища
3	Вчитель	47	вища

Приклад використання різних шкал для вимірювань властивостей температурних умов однієї системи наведено в таблиці 2.3.

Таблиця 2.3 − Множина вимірювань властивостей однієї системи

Дата змінення	**Хмарність (номінальна шкала)**	**Температура о 7 годині (інтервальна шкала)**	**Сила вітру (порядкова шкала)**
3 жовтня	Хмарно	22ºС	Сильний вітер
4 жовтня	Напівхмарно	17ºС	Слабий вітер
5 жовтня	Ясно	23ºС	Дуже сильний вітер

Типи наборів даних. Найбільш часто зустрічаються дані, що складаються із записів (record data).

Приклади таких наборів даних: табличні дані, матричні дані, документальні дані, транзакційні або операційні.

Табличні дані – дані, що складаються із записів, кожен з яких складається з фіксованого набору атрибутів.

Транзакційні дані представляють собою особливий тип даних, де кожен запис, що є транзакцією, включає набір значень.

Приклад транзакційної бази даних, що містить перелік покупок клієнтів магазину, наведено в таблиці 2.4.

Таблиця 2.4 − Приклад транзакційних даних

TID	Покупки
1	хліб, лимонад, молоко
2	пиво, хліб
3	пиво, лимонад, цукерки, молоко
4	пиво, хліб, цукерки, молоко
5	лимонад, цукерки, молоко

Графічні дані. Приклади графічних даних: молекулярні структури; графи (рис. 2.1); карти.

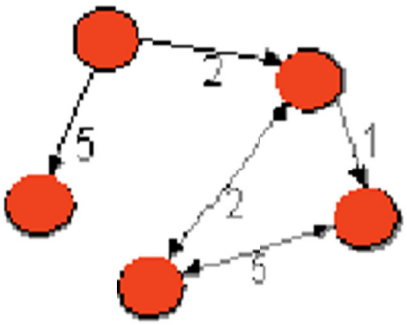

Рисунок 2.1 – Приклад графу

За допомогою карт, наприклад, можна відстежити зміни об’єктів у часі та просторі, визначити характер їх розподілу на площині або в просторі.

Перевагою графічного представлення даних є простота їх сприйняття у порівнянні, наприклад, з табличними даними.

Приклад карти, що є картою Кохонена (моделлю нейронних мереж), представлений на рис. 2.2.

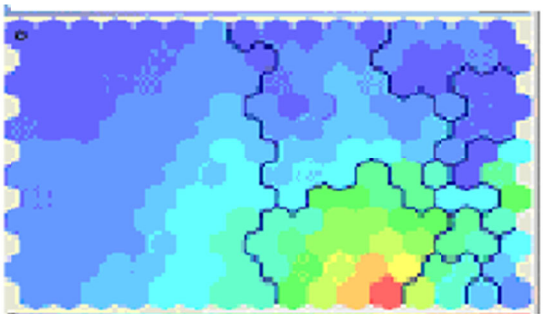

Рисунок 2.2 – Приклад даних типу «Карта Кохонена»

Хімічні дані представляють собою особливий тип даних. Приклад таких даних: молекула бензолу C_6H_6 (рис. 2.3).

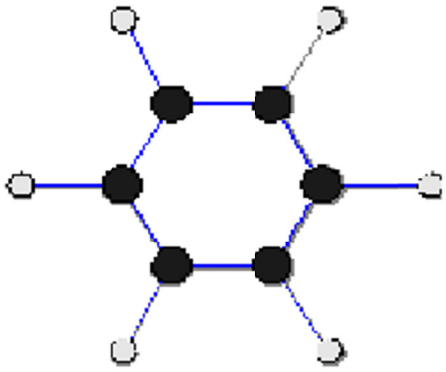

Рисунок 2.3 – Приклад хімічних даних

2.2 Формати зберігання даних

Одна з основних особливостей даних сучасного світу полягає в тому, що їх стає дуже багато.

Можливі чотири аспекти роботи з даними:

- визначення даних;
- обчислення;
- маніпулювання;
- обробка (збір, передача тощо).

При маніпулюванні даними використовується структура даних типу «файл». Файли можуть мати різні формати.

Більшість інструментів Data Mining дозволяють імпортувати дані з різних джерел, а також експортувати результуючі дані в різні формати. Дані для експериментів зручно зберігати в якомусь одному форматі. У деяких інструментах Data Mining ці процедури називаються імпорт / експорт даних, інші дозволяють напряму відкривати різні джерела даних і зберігати результати Data Mining в одному із запропонованих форматів. Найбільш поширеним форматом зберігання даних для Data Mining виступають бази даних.

2.3 Якісний аналіз даних з використанням Data Mining.

Для якісного аналізу будь-яких даних слід дотримуватися загальної схеми використання Data Mining:

1. Висування гіпотез.
2. Збір та систематизація даних.
3. Підбір адекватної моделі.
4. Тестування та інтерпретація отриманих даних.
5. Використання в реальних умовах.

Ця схема не залежить від предметної області та сфери діяльності. Вона є універсальною.

1) Висування гіпотез

Гіпотезою будемо вважати припущення про вплив певних факторів на процес, що досліджується.

Автоматизувати процес висування гіпотез є вкрай складно, тому цю задачу мають розв’язувати експерти – фахівці в предметній області.

Слід довіритися їх досвіду та здоровому глузду, максимально використати ці знання про предмет досліджень і зібрати як найбільше гіпотез/припущень.

Зазвичай, добрі результати надають тактики «круглого столу» або «мозкової атаки». На початку слід зібрати та систематизувати всі ідеї, а оцінювати їх пізніше. У результаті повинен бути складений перелік з описів всіх факторів досліджуваного об’єкту.

Наприклад, для задачі прогнозування попиту товару потрібно скласти перелік факторів, що впливатимуть на об’єкт і експертно оцінити суттєвість кожного з них (табл. 2.5).

Таблиця 2.5 − Вплив кожного фактору (у %) на попит на товар

Сезон	100
День тижня	80
Обсяг продажів за попередні тижні	100
Обсяг продажів за аналогічний період минулого року	95
Рекламна компанія	60
Маркетингові заходи	40
Якість продукції	50
Бренд	25
Коливання ціни від середньоринкової	60
Наявність подібного товару в конкурентів	15

Згодом під час аналізу може з’ясуватися, що фактор, який експерти оцінили як важливий, буде мати незначний вплив на процес і навпаки.

2) Збір та систематизація даних

2.1. Збір даних

Для аналізу потрібно як найбільше даних, бо це надає можливість оцінити вплив максимальної кількості показників. Згодом простіше відхилити певну частину даних, аніж розпочинати новий збір.

Методи збору:

1. Отримання даних із внутрішніх джерел.

Це не складно, бо така інформація зазвичай зберігається в облікових системах у табличній формі, де існують різні механізми отримання звітів та експортування даних.

2. Отримання відомостей із непрямих даних.

Наприклад, потрібно оцінити реальний фінансовий стан мешканців певного регіону. Існує кілька категорій товару (зокрема, авто), що різняться за ціною – для незаможних, середнього класу, заможних. Якщо отримати звіт про продажі товару в цьому районі і проаналізувати пропорції, то дійдемо до висновку: чим більшим є відсоток продажів дорогого товару, тим заможнішими є мешканці.

3. Використання відкритих джерел.

До широкого загалу надаються статистичні збірники, звіти корпорацій, результати маркетингових досліджень, соціологічні опитування.

4. Влаштування власних маркетингових досліджень та подібних заходів по збору даних.

Це зазвичай є дорогим заходом, але доволі ефективним.

5. Наповнення даних згідно експертних оцінок співробітниками організації.

Слід оцінити вартість збору даних, що потрібні для аналізу. Одні дані беруться з публічних інформаційних джерел, інші мають бути оплачені, дані про діяльність конкурентів можуть бути доволі дорогими.

Вартість збору інформації різними методами суттєво різниться за ціною та витраченим часом, тому слід зважати на співвідношення теперішніх витрат із майбутніми результатами.

Від даних, які експерти вважають несуттєвими, певна річ можна відмовитися, але не від значущих даних, бо аналіз базуватиметься в цьому випадку на другорядних факторах і, відповідно, отримана модель буде надавати нестабільні та невірні результати.

2.4 Системи управління базами даних.

Не кожен блок інформації можна вважати базою даних. ***База даних*** – це сукупність даних, яким властива структурованість і взаємопов'язаність, а також незалежність від прикладних програм.

Пояснимо, що означають названі властивості бази даних. Щоб користувач легко міг знаходити потрібну інформацію, остання має бути організована певним чином. Це стосується не лише інформації в комп'ютері, а й будь-якої інформації про об'єкти реального світу. Скажімо, зручно знаходити потрібну книгу в бібліотеці, користуючись каталогом. Легко відшукати в газеті оголошення, що вас цікавлять. Така легкість пошуку можлива завдяки тому, що дані в каталозі або в газеті мають структуру, або, інакше, *структуровані*. Усі книги описані однаково: автор, назва, видавництво, рік видання тощо. Усі оголошення з продажу розміщені по рубриках і також мають визначену структуру: короткий опис товару, ціна, телефон.

Структура бази даних складніша, ніж структура простого каталогу або набору газетних оголошень. Це зумовлено насамперед властивістю *взаємопов'язаності* даних у базі. Пояснимо це на такому прикладі: скажімо, ви хотіли б, крім каталожних карток, що описують кожну книгу, мати картки з інформацією про кожного автора (рік народження, літературний жанр, хобі тощо). Якби такі картки були створені, це був би приклад взаємозалежності даних: відомості про окрему книгу пов'язані з інформацією про автора. Цей зв'язок здійснюється через визначений параметр – прізвище автора.

Нарешті, остання з названих властивостей баз даних – це їхня *незалежність від прикладних програм.* Бази даних складаються таким чином,

щоб із ними можна було працювати в різних програмних середовищах і на різних комп'ютерних платформах.

Щоб оперувати даними, які становлять базу, необхідна окрема програма – система управління базами даних. Керівна програма, призначена для збереження, пошуку й обробки даних у базі, називається ***системою управління базами даних*** (скорочено СУБД).

Система управління базами даних – це прикладна програма, реалізована на електронній обчислювальній машині чи обчислювальному комплексі. За допомогою її можна:

– створювати структуру бази даних, вводити інформацію та зберігати її на зовнішніх носіях;

– виконувати певне коло операцій із даними;

– одержувати результати та зберігати їх на зовнішніх носіях або передавати на віддалені термінали;

–виводити інформацію на термінал у зручній для користувача формі або на друкувальні пристрої;

– давати можливість працювати з базами даних багатьом користувачам.

У цьому визначенні відсутній людський фактор – персонал, який відповідає за дані (адміністратор бази даних), але для розуміння роботи СУБД буде достатньо попереднього визначення.

Сучасні СУБД – це програмні додатки, які дозволяють виконувати різноманітні завдання. Усі існуючі системи задовольняють, як правило, таким вимогам:

– можливості маніпулювання даними (введення, вибір, вставка, відновлення, видалення тощо) – основні операції з даними виконуються під керуванням СУБД;

– можливість пошуку і формування запитів – за допомогою запитів користувач може оперативно одержувати різну інформацію, що зберігається в базі даних;

– забезпечення цілісності (узгодженості) даних – під час використання даних багатьма користувачами важливо забезпечити коректність операцій, щоб запобігти порушенню узгодженості даних (порушення узгодженості даних може призвести до їх невідновної втрати);

– забезпечення захисту і таємності – крім захисту від некоректних дій користувачів, важливо забезпечити захист даних від несанкціонованого доступу і від апаратних збоїв.

Важливими показниками є продуктивність СУБД, витрати на збереження і використання даних, простота звернення до бази даних тощо. Проникнення в базу осіб, які не мають на це права, може спричинити руйнацію даних. Таємність бази даних дозволяє визначити коло осіб, що мають доступ до інформації, і порядок доступу.

Сьогодні існує багато СУБД, що відрізняються архітектурою, внутрішньою мовою програмування, операційною системою, якою вони керуються, а також іншими характеристиками. Найпопулярнішими СУБД, що

встановлюються в невеликих організаціях і орієнтовані на роботу з кінцевими користувачами, є Access, FoxPro, Paradox. До складніших систем належать розподілені СУБД, що призначені для роботи з великими базами даних, розподіленими на кількох серверах (сервери можуть міститися в різних регіонах). Потужними СУБД такого типу є Oracle, Sybase, Informix.

Вимоги до СУБД

СУБД разом із базами даних іноді називають банком даних. У банках даних повинні бути передбачені засоби, що забезпечують захист певних областей даних від несанкціонованого доступу.

Банк даних повинен відповідати таким вимогам:

– мати можливість оновлення, поповнення та розширення баз даних;

– забезпечити високу надійність зберігання інформації;

– видавати повну та вірогідну інформацію на запити;

– мати засоби, що забезпечують захист баз даних від несанкціонованого доступу.

Основні функції СУБД

До основних функцій СУБД належать такі:

– опис бази даних (вказати назви полів, їх довжину, тип та інше);

– введення до бази даних підготовлених даних;

– перевірка правильності введення даних (контроль за типом);

– редагування даних (вилучення, заміна, коректування, вставка, доповнення);

– обробка запитів від користувачів (пошук певної інформації);

– забезпечення одночасної роботи декількох користувачів з однією базою даних;

– захист даних.

Питання для самоконтролю

1. Дайте визначення поняттю вибірка (sample).
2. Дайте визначення поняттю гіпотеза.
3. Наведіть приклади застосування гіпотез.
4. Які види шкал ви знаєте? Чим вони відрізняються?
5. Які типи даних ви знаєте?
6. З яких етапів складається загальна схема використання Data Mining?

ЗМІСТОВИЙ МОДУЛЬ 2. МЕТОДИ, СТАДІЇ ТА ЗАВДАННЯ DATA MINING

Тема 3. Методи та стадії Data Mining

Мета: ознайомитися з класифікацією стадій та методів Data Mining.

План

3.1 Класифікація стадій Data Mining
3.2 Класифікація технологічних методів Data Mining
3.3 Властивості методів Data Mining

Основні поняття

Статистичні методи Data Mining. Кібернетичні методи Data Mining. Властивості методів Data Mining.

3.1 Класифікація стадій Data Mining

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

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

Більшість аналітичних методів, що використовуються в технології Data Mining – це відомі математичні алгоритми і методи. Новим в їх застосуванні є можливість їх використання при розв'язуванні тих чи інших конкретних задач, зумовлена можливостями нових технічних і програмних засобів. Слід зазначити, що більшість методів Data Mining були розроблені в рамках теорії штучного інтелекту.

Метод (method) являє собою норму або правило, певний шлях, спосіб, прийом розв'язання задачі теоретичного, практичного, пізнавального, управлінського характеру.

Поняття алгоритму з'явилося задовго до створення електронних обчислювальних машин. Зараз алгоритми є основою для вирішення багатьох прикладних і теоретичних завдань у різних сферах людської діяльності, у більшості – це завдання, вирішення яких передбачено з використанням комп'ютера.

Алгоритм (algorithm) – точний припис щодо послідовності дій (кроків), що перетворюють вихідні дані в шуканий результат.

Data Mining може складатися з двох або трьох стадій:

Стадія 1. Виявлення закономірностей (**вільний пошук**).

Стадія 2. Використання виявлених закономірностей для передбачення невідомих значень (**прогностичне моделювання**).

На додаток до цих стадій іноді вводять **стадію валідації**, наступну за стадією вільного пошуку. **Мета валідації** – перевірка достовірності знайдених закономірностей. Однак, валідація вважається частиною першої стадії, оскільки в реалізації багатьох методів, зокрема, нейронних мереж і дерев рішень, передбачено поділ загальної множини даних на навчальну і перевірочну, і останнє дозволяє перевіряти достовірність отриманих результатів.

Стадія 3. Аналіз винятків – стадія призначена для виявлення і пояснення аномалій, знайдених у закономірностях.

Отже, процес Data Mining може бути представлений низкою таких послідовних стадій:

ВІЛЬНИЙ ПОШУК (у тому числі ВАЛІДАЦІЯ) -> **ПРОГНОСТИЧНЕ МОДЕЛЮВАННЯ** -> **АНАЛІЗ ВИНЯТКІВ**

1. Вільний пошук (Discovery).

На стадії вільного пошуку здійснюється дослідження набору даних із метою пошуку прихованих закономірностей. Попередні гіпотези щодо виду закономірностей тут не визначаються.

Закономірність (law) – істотний і постійно повторюваний взаємозв'язок, що визначає етапи і форми процесу становлення, розвитку різних явищ або процесів. Система Data Mining на цій стадії визначає шаблони, для отримання яких в системах **OLAP**, наприклад, аналітик повинен обдумувати і створювати множину запитів. Тут же аналітик звільняється від такої роботи – шаблони шукає за нього система. Особливо корисне застосування даного підходу в надвеликих базах даних, де визначити закономірність шляхом створення запитів досить складно, для цього потрібно перепробувати багато різноманітних варіантів.

Вільний пошук представлений такими **діями**:

– виявлення закономірностей умовної логіки (conditional logic);

– виявлення закономірностей асоціативної логіки (associations and affinities);

– виявлення трендів і коливань (trends and variations).

Припустимо, є база даних кадрового агенства з даними про професії, стаж, вік і бажаний рівень винагороди. У разі самостійного задавання запитів аналітик може отримати приблизно такі результати: середній бажаний рівень винагороди фахівців у віці від 25 до 35 років дорівнює 1200 умовних одиниць. У разі вільного пошуку система сама шукає закономірності, необхідно лише задати цільову змінну. У результаті пошуку закономірностей система сформує набір логічних правил " якщо..., то...".

Можуть бути знайдені, наприклад, такі закономірності

«Якщо вік <20 років і бажаний рівень винагороди >700 умовних одиниць, то в 75% випадків здобувач шукає роботу програміста»

або

«Якщо вік >35 років і бажаний рівень винагороди >1200 умовних одиниць, то в 90% випадків здобувач шукає роботу керівника». Цільовою змінною в описаних правилах виступає професія.

Задавши іншу цільову змінну, наприклад, вік, отримуємо такі правила: «Якщо здобувач шукає керівну роботу і його стаж >15 років, то вік здобувача >35 років у 65% випадків».

Описані дії, в рамках стадії вільного пошуку, виконуються за допомогою:

– індукції правил умовної логіки (задачі класифікації та кластеризації, опис у компактній формі близьких або схожих груп об'єктів);

– індукції правил асоціативної логіки (задачі асоціації та послідовності й одержувана за їх допомогою інформація);

– визначення трендів і коливань (вихідний етап задачі прогнозування).

На стадії вільного пошуку також повинна здійснюватися валідація закономірностей, тобто перевірка їх достовірності на частині даних, які не брали участь у формуванні закономірностей. Такий прийом розділення даних на навчальну і перевірочну множину часто використовується в методах нейронних мереж і дерев рішень.

2. Прогностичне моделювання (Predictive Modeling).

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

Прогностичне моделювання включає такі дії:

– передбачення невідомих значень (outcome prediction);

– прогнозування розвитку процесів (forecasting).

У процесі прогностичного моделювання розв'язуються задачі класифікації та прогнозування.

При розв'язанні задачі класифікації результати роботи першої стадії (індукції правил) використовуються для віднесення нового об'єкта, з певною впевненістю, до одного з відомих, визначених класів на підставі відомих значень.

При розв'язанні задачі прогнозування результати першої стадії (визначення тренда або коливань) використовуються для передбачення невідомих (пропущених або ж майбутніх) значень цільової змінної (змінних).

Продовжуючи розглянутий приклад першої стадії, можемо зробити такий висновок.

Знаючи, що здобувач шукає керівну роботу і його стаж >15 років, на 65% можна бути впевненим у тому, що вік здобувача >35 років. Або ж, якщо вік здобувача >35 років і бажаний рівень винагороди >1200 умовних одиниць, на 90% можна бути впевненим у тому, що здобувач шукає керівну роботу.

Порівняємо вільний пошук і прогностичне моделювання з точки зору логіки.

Вільний пошук розкриває загальні закономірності. Він за своєю природою індуктивний. Закономірності, отримані на цій стадії, формуються від особистого до загального. У результаті ми отримуємо деяке загальне знання про деякий клас об'єктів на підставі дослідження окремих представників цього класу.

Правило: «Якщо вік здобувача <20 років і бажаний рівень винагороди >700 умовних одиниць, то в 75% випадків здобувач шукає роботу програміста». На підставі особистого, тобто інформації про деякі властивості класу «вік <20 років» і «бажаний рівень винагороди >700 умовних одиниць», робимо висновок про загальне, а саме: шукають роботу – програмісти.

Прогностичне моделювання, навпаки, дедуктивне. Закономірності, отримані на цій стадії, формуються від загального до особистого і одиничного. Тут ми отримуємо нове знання про деякий об'єкт або ж групи об'єктів на підставі:

– знання класу, до якого належать досліджувані об'єкти;

– знання загального правила, діючого в межах даного класу об'єктів.

Знаємо, що претендент шукає керівну роботу і його стаж >15 років, на 65% можна бути впевненим у тому, що вік здобувача >35 років.

На підставі деяких загальних правил, а саме: мета здобувача – керівна робота і його стаж >15 років, ми робимо висновок про одиничне – вік здобувача >35 років.

Слід зазначити, що отримані закономірності, а точніше, їх конструкції, можуть бути прозорими, тобто допускають тлумачення аналітика (розглянуті вище правила), і непрозорими, так званими «чорними ящиками». Типовий приклад останньої конструкції – нейронна мережа.

3. Аналіз винятків (forensic analysis).

На третій стадії Data Mining аналізуються виключення або аномалії, виявлені в знайдених закономірностях.

Дія, що виконується на цій стадії, це виявлення відхилень (deviation detection). Для виявлення відхилень необхідно визначити норму, яка розраховується на стадії вільного пошуку.

Повернемося до одного з прикладів, розглянутих вище.

Знайдено правило «Якщо вік >35 років і бажаний рівень винагороди >1200 умовних одиниць, то в 90% випадків здобувач шукає керівну роботу». Виникає питання – до чого віднести решту 10% випадків? Тут можливі два варіанти. Перший з них – існує деяке логічне пояснення, яке також може бути оформлено у вигляді правила. Другий варіант для решти 10% – це помилки вихідних даних. У цьому випадку стадія аналізу винятків може бути використана для очищення даних.

3.2 Класифікація технологічних методів Data Mining

Усі методи Data Mining поділяються на дві великі групи за принципом роботи з вихідними навчальними даними. У цій класифікації верхній рівень

визначається на підставі того, зберігаються дані після Data Mining чи вони дистилюються для подальшого використання.

1. Безпосереднє використання даних, або збереження даних.

У цьому випадку вихідні дані зберігаються в явному деталізованому вигляді і безпосередньо використовуються на стадіях прогностичного моделювання та/або аналізу винятків. Проблема цієї групи методів – при їх використанні можуть виникнути складності аналізу надвеликих баз даних.

Методи цієї групи: **кластерний аналіз, метод найближчого сусіда, метод *k*-найближчого сусіда, міркування за аналогією.**

2. Виявлення і використання формалізованих закономірностей, або дистиляція шаблонів.

При технології дистиляції шаблонів один зразок (шаблон) інформації витягується з вихідних даних і перетворюється в якісь формальні конструкції, вигляд яких залежить від використовуваного методу Data Mining. Цей процес виконується на стадії вільного пошуку, у першій же групі методів ця стадія в принципі відсутня. На стадіях прогностичного моделювання та аналізу винятків використовуються результати стадії вільного пошуку, вони значно компактніше самих баз даних. Нагадаємо, що конструкції цих моделей можуть бути трактовані аналітиком або не трактовані («чорні ящики»).

Методи цієї групи: **логічні методи, методи візуалізації; методи крос-табуляції; методи на основі рівнянь.**

Логічні методи, або методи логічної індукції, включають:

– нечіткі запити і аналізи;

– символьні правила;

–дерева рішень;

– генетичні алгоритми.

Методи цієї групи є такими, що найкраще інтерпретуються – вони оформляють знайдені закономірності, в більшості випадків у досить прозорому вигляді з точки зору користувача. Отримані правила можуть включати безперервні і дискретні змінні. Слід зауважити, що дерева рішень можуть бути легко перетворені в набори символьних правил шляхом генерації одного правила по шляху від кореня дерева до його термінальної вершини. Дерева рішень і правила фактично є різними способами розв’язання однієї задачі і відрізняються лише за своїми можливостями. Крім того, реалізація правил здійснюється більш повільними алгоритмами, ніж індукція дерев рішень.

Методи крос-табуляції: агенти, байєсовські (довірчі) мережі, крос-таблична візуалізація. Останній метод не зовсім відповідає одній з властивостей Data Mining – самостійного пошуку закономірностей аналітичною системою. Однак надання інформації у вигляді крос-таблиць забезпечує реалізацію основного завдання Data Mining – пошук шаблонів, тому цей метод можна також вважати одним із методів Data Mining.

Методи на основі рівнянь висловлюють виявлені закономірності у вигляді математичних виразів – рівнянь. Отже, вони можуть працювати лише з чисельними змінними, і змінні інших типів повинні бути закодовані

відповідним чином. Це дещо обмежує застосування методів цієї групи, проте вони широко використовуються при вирішенні різних завдань, особливо завдань прогнозування.

Основні методи цієї групи: **статистичні методи і нейронні мережі.**

Статистичні методи найбільш часто застосовуються для розв'язання задач прогнозування. Існує багато методів статистичного аналізу даних, серед них, наприклад, кореляційно-регресійний аналіз, кореляція рядів динаміки, виявлення тенденцій динамічних рядів, гармонійний аналіз.

Інша класифікація поділяє все різноманіття методів Data Mining на дві групи: **статистичні** та **кібернетичні** методи. Ця схема поділу заснована на різних підходах до навчання математичних моделей.

Слід зазначити, що існує два підходи віднесення статистичних методів до Data Mining. Перший з них протиставляє статистичні методи і Data Mining, його прихильники вважають класичні статистичні методи окремим напрямом аналізу даних. Відповідно до другого підходу, статистичні методи аналізу є частиною математичного інструментарію Data Mining. Більшість авторитетних джерел дотримується другого підходу.

У цій класифікації розрізняють дві групи методів:

• *статистичні методи*, засновані на використанні усередненого накопиченого досвіду, який відображений у ретроспективних даних;

• *кібернетичні методи*, що включають множину різнорідних математичних підходів.

Недолік такої класифікації: і статистичні, і кібернетичні алгоритми тим чи іншим чином спираються на зіставлення статистичного досвіду з результатами моніторингу поточної ситуації.

Перевагою такої класифікації є її зручність для інтерпретації – вона використовується при описі математичних засобів сучасного підходу до вилучення знань із масивів вихідних спостережень (оперативних і ретроспективних), тобто в задачах Data Mining.

Статистичні методи Data mining являють собою чотири взаємопов'язаних розділи:

– попередній аналіз природи статистичних даних (перевірка гіпотез стаціонарності, нормальності, незалежності, однорідності, оцінка виду функції розподілу, її параметрів тощо);

– виявлення зв'язків і закономірностей (лінійний і нелінійний регресійний аналіз, кореляційний аналіз та ін.);

– багатовимірний статистичний аналіз (лінійний і нелінійний дискримінантний аналіз, кластерний аналіз, компонентний аналіз, факторний аналіз та ін.);

– динамічні моделі і прогноз на основі часових рядів.

Кібернетичні методи Data Mining відносяться **до і**ншого напрямку Data Mining – це множина підходів, об'єднаних ідеєю комп'ютерної математики та використання теорії штучного інтелекту.

До цієї групи відносяться такі методи:

– штучні нейронні мережі (розпізнавання, кластеризація, прогнозування);

– еволюційне програмування (у т.ч. алгоритми методу групового обліку аргументів);

– генетичні алгоритми (оптимізація);

– асоціативна пам’ять (пошук аналогів, прототипів);

– нечітка логіка;

– дерева рішень;

– системи обробки експертних знань.

Методи Data Mining також можна класифікувати за задачами Data Mining. Відповідно до такої класифікації виділяємо дві групи. Перша з них – це поділ методів Data Mining на вирішальні завдання сегментації (тобто задачі класифікації та кластеризації) і завдання прогнозування.

У відповідності до другої класифікації за задачами методи Data Mining можуть бути спрямовані на отримання описових і прогнозуючих результатів.

Описові методи служать для знаходження шаблонів або зразків, що описують дані, які піддаються інтерпретації з точки зору аналітика.

До методів, спрямованих на отримання описових результатів, відносяться ітеративні методи кластерного аналізу, в тому числі: алгоритм *k*-середніх, *k*-медіани, ієрархічні методи кластерного аналізу, карти Кохонена, методи крос-табличної візуалізації, різні методи візуалізації та ін.

Прогнозуючі методи використовують значення одних змінних для передбачення / прогнозування невідомих (пропущених) або майбутніх значень інших (цільових) змінних.

До методів, спрямованих на отримання прогнозуючих результатів, відносяться такі методи: нейронні мережі, дерева рішень, лінійна регресія, метод найближчого сусіда, метод опорних векторів тощо.

3.3 Властивості методів Data Mining

Різні методи Data Mining характеризуються певними властивостями, які можуть бути визначальними при виборі методу аналізу даних. Методи можна порівнювати між собою, оцінюючи характеристики їх властивостей.

Серед основних властивостей і характеристик методів Data Mining розглянемо такі: точність, масштабованість, інтерпретованість, здатність до перевірки, трудомісткість, гнучкість, швидкість і популярність.

Масштабованість – властивість обчислювальної системи, яка забезпечує передбачуваний зріст системних характеристик, наприклад, швидкості реакції, загальної продуктивності тощо, при додаванні до неї обчислювальних ресурсів.

Більшість інструментів Data Mining, пропонованих зараз на ринку програмного забезпечення, реалізують відразу кілька методів, наприклад, дерева рішень, індукцію правил і візуалізацію, або ж нейронні мережі, карти Кохонена та візуалізацію. В універсальних прикладних статистичних пакетах (наприклад, SPSS, SAS, STATGRAPHICS, Statistica) реалізується широкий

спектр найрізноманітніших методів (як статистичних, так і кібернетичних). Слід враховувати, що для можливості їх використання, а також для інтерпретації результатів роботи статистичних методів (кореляційного, регресійного, факторного, дисперсійного аналізу) потрібні спеціальні знання в галузі статистики.

Універсальність того чи іншого інструмента часто накладає певні обмеження на його можливості. Перевагою використання таких універсальних пакетів є можливість відносно легко порівнювати результати побудованих моделей, отримані різними методами. Така можливість реалізована, наприклад, в пакеті Statistica, де порівняння засноване на так званій «конкурентній оцінці моделей». Ця оцінка полягає в застосуванні різних моделей до одного і того ж набору даних і в наступному порівнянні їх характеристик для вибору найкращої з них.

Високим рівнем масштабує мості володіють класичні методи (лінійна регресія) та дерева рішень. Класичні методи (лінійна регресія), методи візуалізації та дерева рішень мають високу інтерпретованість. Методи візуалізації мають дуже високу трудомісткість. Високою швидкодією володіють класичні методи (лінійна регресія), дерева рішень та алгоритм *k*-найближчого сусіда.

Питання для самоконтролю

1. Дайте визначення методу та алгоритму.
2. Які ви знаєте методи і алгоритми Data Mining?
3. Із яких стадій складається процес Data Mining?
4. Які ви знаєте види класифікацій технологічних методів Data Mining?
5. Приведіть приклади застосування інструменту Data Mining асоціації (або відношення)?
6. Для чого застосовуються інструментарій Data Mining дерева рішень (наведіть приклад)?

Лабораторне заняття №2

Тема: Оцінка статистичних характеристик на мові R

Мета роботи: Навчитися розраховувати статистичні характеристики та будувати гістограми на мові R.

Завдання.

1. Підключіть необхідні для роботи бібліотеки (dplyr, readr та ggplot2).
2. Збережіть із системи Moodle та відкрийте у R файл macro.csv (команда read.csv()).
3. Порівняйте розподіл обсягів ВВП, рівня безробіття, обсягів торгівлі по країнах, використовуючи коробчасті діаграми (діаграми розмістіть горізонтально).

4. Для обраної країни, використовуючи команду filter(), побудуйте стовпчикові діаграми обсягів ВВП, рівня безробіття та обсягів торгівлі.

5. Для обраної країни, використовуючи команду filter(), побудуйте гістограми обсягів ВВП, рівня безробіття та обсягів торгівлі.

6. Для обраної країни, використовуючи команду filter(), побудуйте графік розсіювання обсягів ВВП, рівня безробіття та обсягів торгівлі.

7. Для обраної країни розрахуйте медіанні значення обсягів ВВП, рівня безробіття та обсягів торгівлі.

8. Для обраної країни розрахуйте середнє значення обсягів ВВП, рівня безробіття та обсягів торгівлі.

9. Для обраної країни розрахуйте значення середньоквадратичного відхилення обсягів ВВП, рівня безробіття та обсягів торгівлі.

10. Згенеруйте нормальний розподіл, який має середнє значення та середньоквадратичне відхилення обсягу ВВП для обраної країни. Для того, щоб послідовність, яка генерується була сталою, при кожному виконанні коду, встановіть параметр set.seed. Додайте це значення до таблиці. Побудуйте гістограму для симуляції. Перевірте, чи є розподіл, нормальним за допомогою функції qqplot. Також перевірте на нормальність розподілу обсяг ВВП для обраної країни.

Хід роботи.

1. Підключіть потрібні для виконання завдання роботи бібліотеки:

```
library(readr)
library(dplyr)
library(ggplot2)
```

2. Запишіть у пам’ять дані файлу macro.csv

```
macro<-read.csv(file="macro.csv", sep=",", header=TRUE)
```

3. Порівняйте розподіл обсягів ВВП, рівня безробіття, обсягів торгівлі по країнах, використовуючи коробчасті діаграми (діаграми розмістіть горізонтально)

Для створення коробчастої діаграми використовується функція:

ggplot(data, aes(x,y))+geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha color, fill, group, linetype, shape, size, weight

Щоб розмістити коробчасті діаграми горизонтально використаємо параметр *coord_flip()*.

Наприклад, коробчаста діаграма для розподілу обсягів ВВП (рис.3.1):

```
ggplot(macro, aes(country, gdp))+geom_boxplot()+coord_flip()
```

Аналогічно будуються коробчасті діаграми для рівня безробіття та обсягів торгівлі по країнах.

4. Для обраної країни, використовуючи команду filter(), побудуйте стовпчикові діаграми обсягів ВВП, рівня безробіття та обсягів торгівлі.

Для створення стовпчикової діаграми використовується функція:

ggplot(data, aes(x))+geom_bar(), x, alpha color, fill, linetype, size, weight

Для підпису вертикальної вісі використовується функція *ylab('Підпис')*

Наприклад, стовпчикова діаграма для розподілу обсягів ВВП для Австрії (рис 3.2)):

ggplot(filter(macro,macro$country=="Austria"),aes(x=gdp))+geom_bar()
+ylab('Кількість')

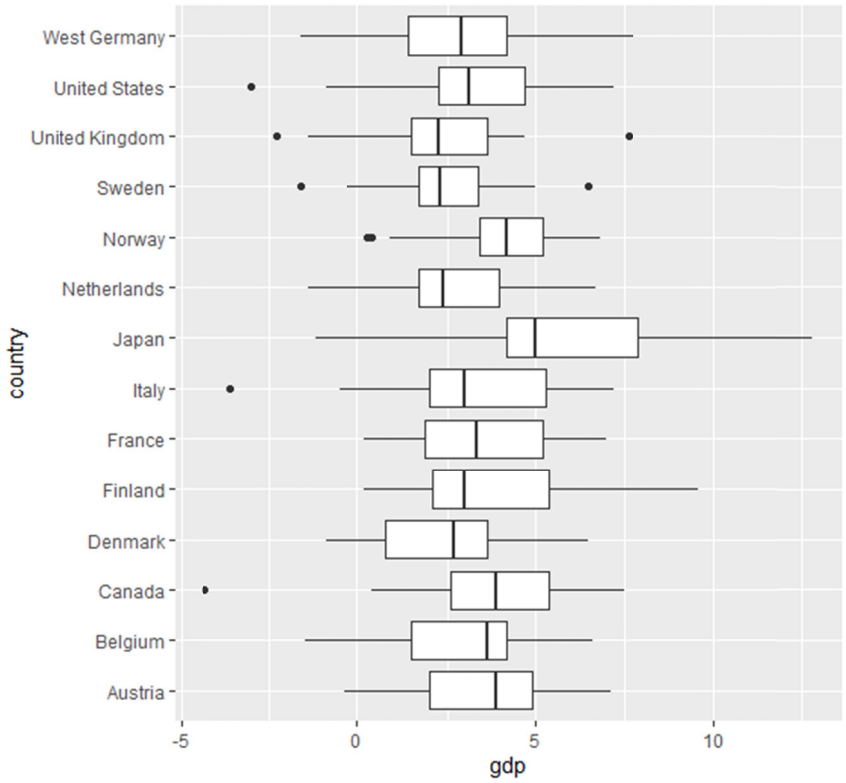

Рисунок 3.1 – Розподіл обсягів ВВП

Аналогічно будуються стовпчикові діаграми для рівня безробіття та обсягів торгівлі по обраній країні.

5. Для обраної країни, використовуючи команду filter(), побудуйте гістограми обсягів ВВП, рівня безробіття та обсягів торгівлі.

Для створення гістограми використовується функція:

ggplot(data, aes(x))+geom_histogram(), x, alpha color, fill, linetype, size, weight

Для підпису вертикальної вісі використовується функція *ylab('Підпис')*

Наприклад, гістограма для розподілу обсягів ВВП для Австрії (рис 3.3):

ggplot(macro_Austria, aes(x=gdp))+geom_histogram(bins=10, color="blue", fill="green")+ylab('Кількість')

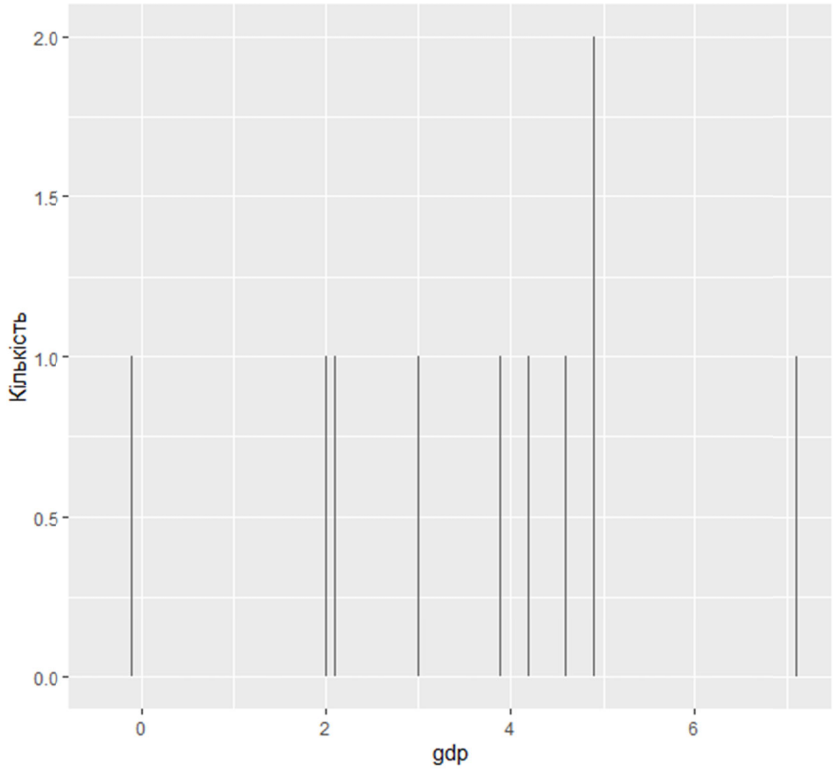

Рисунок 3.2 – Стовпчикова діаграма ВВП для Австрії

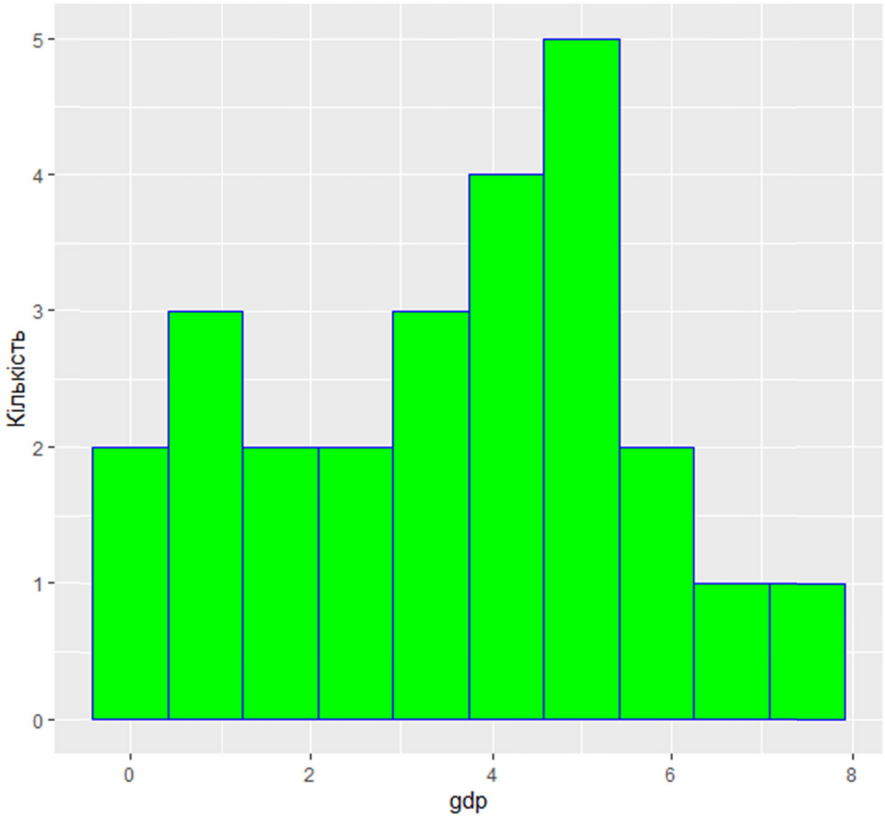

Рисунок 3.3 – Гістограма ВВП для Австрії

Аналогічно будуються гістограми для рівня безробіття та обсягів торгівлі по обраній країні.

6. Для обраної країни, використовуючи команду filter(), побудуйте графік розсіювання обсягів ВВП, рівня безробіття та обсягів торгівлі.

Для створення графіка розсіювання використовується функція:

ggplot(data, aes(x, y))+geom_point(), x, alpha color, fill, shape, size, stroke

Наприклад, графік розсіювання для розподілу обсягів ВВП для Австрії (рис 3.4):

ggplot(filter(macro,macro$country=="Austria"),aes(x=year,y=gdp))+geom_point()

Аналогічно будуються графіки розсіювання для рівня безробіття та обсягів торгівлі по обраній країні.

Для розрахунку арифметичної середньої, медіани, дисперсії, стандартного відхилення, а також мінімального та максимального значень у R використовують функції *mean(), median(), var(), sd(), min() и max()* відповідно.

7. Медіанне значення обсягу ВВП для обраної країни (Австрія) розраховується так:

– присвоїмо змінній *macro_Austria* дані про обрану країну

macro_Austria<-filter(macro, macro$country=="Austria")

– розрахуємо медіанне значення ВВП для обраної країни

median(macro_Austria$gdp)
[1] 3.9

Аналогічно розраховуються медіанні значення для рівня безробіття та обсягів торгівлі по обраній країні.

8. Для обраної країни розрахуйте середнє значення обсягів ВВП, рівня безробіття та обсягів торгівлі

Середнє значення обсягу ВВП для обраної країни (Австрія) розраховується так:

Austria_gdp_mean=mean(macro_Austria$gdp)
Austria_gdp_mean
[1] 3.478027

Аналогічно розраховуються середні значення для рівня безробіття та обсягів торгівлі по обраній країні.

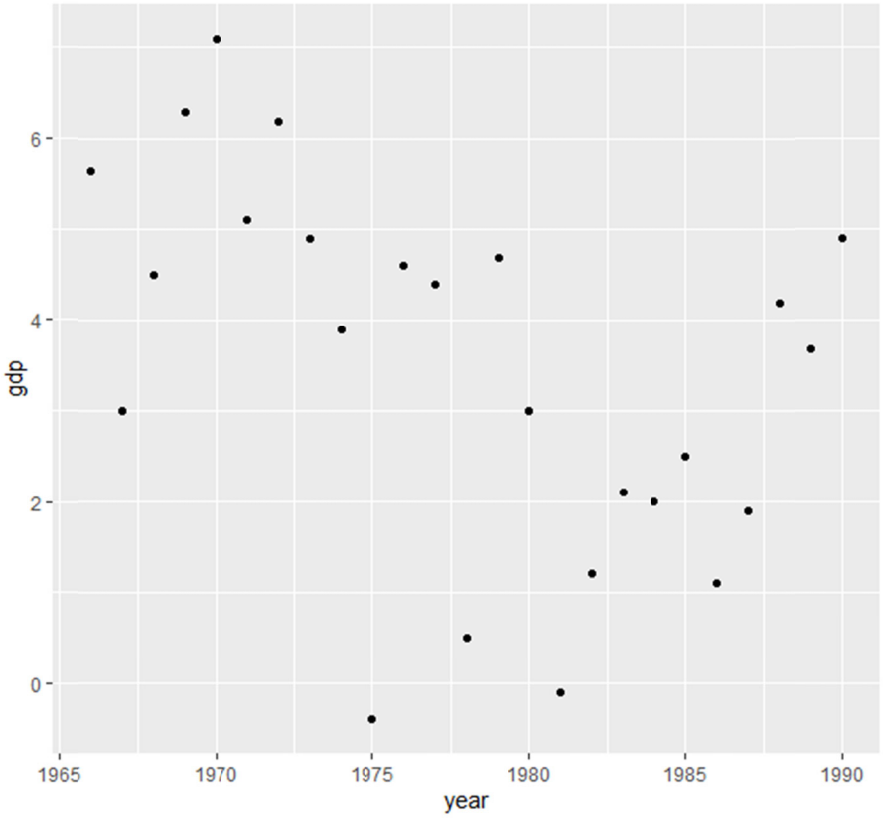

Рисунок 3.4 – Графік розсіювання ВВП для Австрії

9. Для обраної країни розрахуйте значення середньоквадратичного відхилення обсягів ВВП, рівня безробіття та обсягів торгівлі

Середньоквадратичне відхилення обсягу ВВП для обраної країни (Австрія) розраховується так:

```
Austria_gdp_sd=sd(macro_Austria$gdp)
Austria_gdp_sd
[1] 2.049215
```

10. Згенеруйте нормальний розподіл, який має середнє значення та середньоквадратичне відхилення обсягу ВВП для обраної країни. Для того, щоб послідовність, яка генерується була сталою, при кожному виконанні коду, встановіть параметр *set.seed*. Додайте це значення до таблиці. Побудуйте гістограму для симуляції. Перевірте, чи є розподіл нормальним, за допомогою функції *qqplot*. Також перевірте на нормальність розподілу обсяг ВВП для обраної країни

Згенеруємо нормальний розподіл, який має середнє значення та середньоквадратичне відхилення обсягу ВВП для Австрії:

```
set.seed(100)
gdp_simulation<-rnorm(n=nrow(macro_Austria), mean= Austria_gdp_mean, sd=
Austria_gdp_sd)
```

Додамо це значення до таблиці:

macro_Austria$simulation<-gdp_simulation

Перевіремо оновлену структуру таблиці:

```
str(macro_Austria)
'data.frame':    25 obs. of  7 variables:
 $ X         : int  176 177 178 179 180 181 182 183 184 185 ...
 $ country   : chr  "Austria" "Austria" "Austria" "Austria" ...
 $ year      : int  1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 ...
 $ gdp       : num  5.64 3.01 4.5 6.3 7.1 ...
 $ unem      : num  1.7 1.8 2 1.8 1.4 1.2 1 1 1.1 1.7 ...
 $ trade     : num  50.8 51.5 50.9 51.6 55.5 ...
 $ simulation: num  2.45 3.75 3.32 5.3 3.72 ...
```

Побудуємо гістограму для симуляції (рис. 3.5):

```
ggplot(macro_Austria, aes(x=gdp_simulation))+geom_histogram(bins=10,
color="blue", fill="green")
```

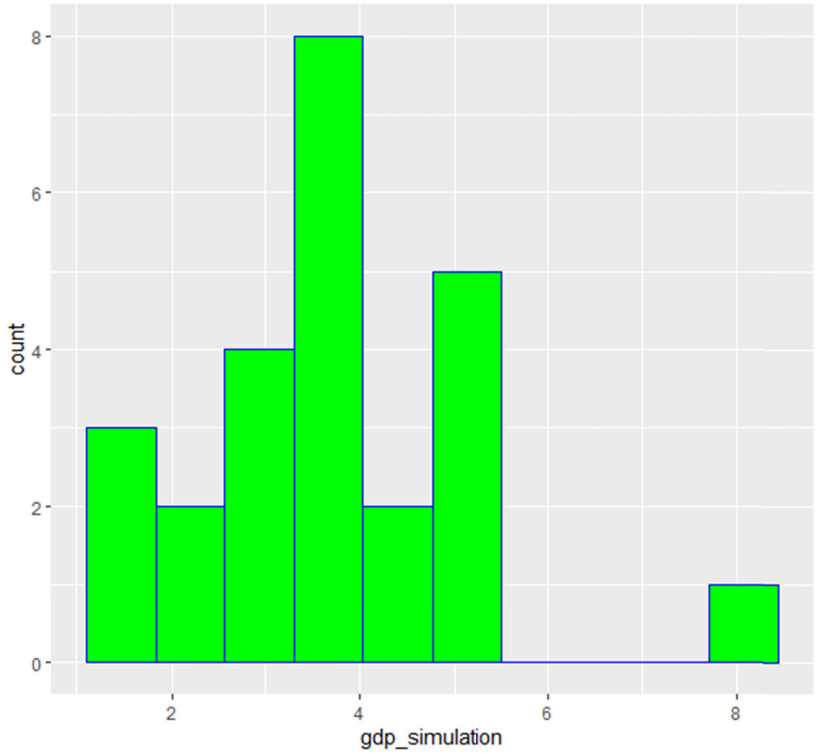

Рисунок 3.5 – Гістограму симуляції

Перевіремо, чи є розподіл нормальним(рис.3.6):

ggplot(macro_Austria, aes(sample=gdp_simulation))+stat_qq()

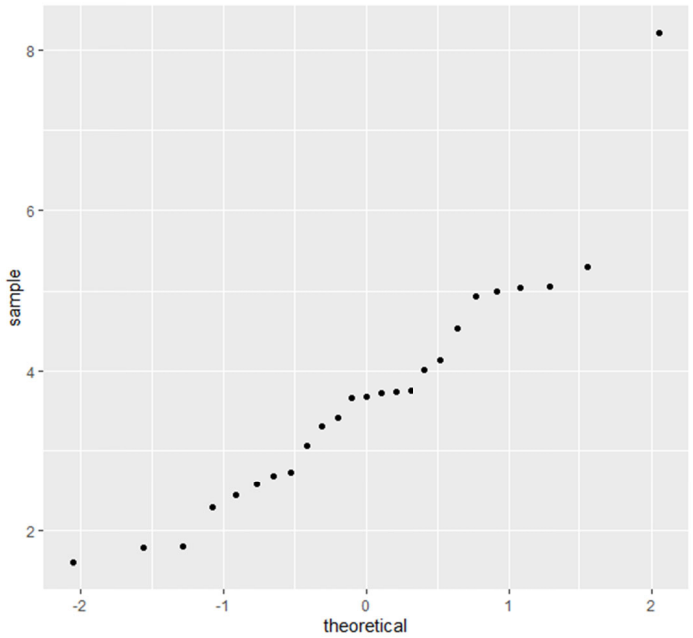

Рисунок 3.6 – Графік для перевірки закону розподіла

Перевіремо на нормальність вихідний ряд значень ВВП (рис.3.7):

ggplot(macro_Austria, aes(sample=gdp))+stat_qq()

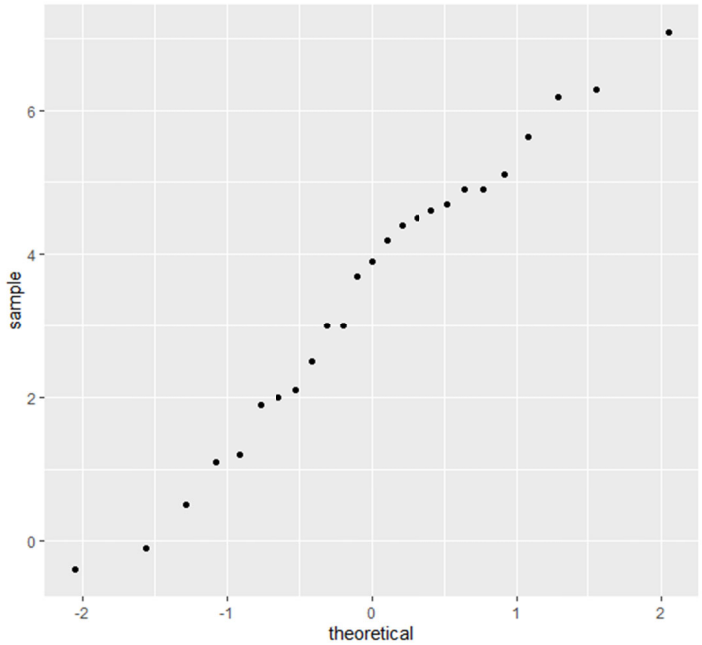

Рисунок 3.7 – Перевірка вихідного ряду значень ВВП

Тема 4. Завдання Data Mining. Інформація та знання

Мета: ознайомитися з класифікацією завдань Data mining в залежності від моделей, що використовуються.

План

4.1 Завдання Data Mining

4.2 Класифікація завдань інтелектуального аналізу даних

4.3 Інформація. Властивості інформації

Основні поняття

Класифікація. Кластеризація. Асоціація. послідовна асоціація. Прогнозування. Оцінювання. Аналіз зв'язків. Візуалізація.

4.1 Завдання Data Mining

В основу технології Data Mining покладена концепція шаблонів, що представляють собою закономірності. У результаті виявлення цих, прихованих від неозброєного ока закономірностей вирішуються завдання інтелектуального аналізу даних

Задачі (tasks) Data Mining іноді називають закономірностями (regularity) або техніками (techniques).

Єдиної думки щодо того, які задачі слід відносити до Data Mining, немає.

Більшість авторитетних джерел перераховують такі задачі: класифікація, кластеризація, прогнозування, асоціація, візуалізація, аналіз і виявлення відхилень, оцінювання, аналіз зв'язків, підведення підсумків.

Класифікація (Classification) є найбільш простою і поширеною задачею Data Mining. У результаті розв'язання задачі класифікації виявляються ознаки, які характеризують групи об'єктів досліджуваного набору даних – класи; за цими ознаками новий об'єкт можна віднести до того чи іншого класу.

Для розв'язання задачі класифікації можуть використовуватися методи: найближчого сусіда (Nearest Neighbor), *k*-найближчого сусіда (*k*-Nearest Neighbor); байєсовські мережі (Bayesian Networks); індукція дерев рішень; нейронні мережі (neural networks).

Кластеризація (Clustering) є логічним продовженням ідеї класифікації. Ця задача більш складна. Особливість кластеризації полягає в тому, що класи об'єктів спочатку не визначені. Результатом кластеризації є розбиття об'єктів на групи.

Приклад методу розв'язання задачі кластеризації: навчання «без вчителя» особливого виду нейронних мереж – самоорганізованих карт Кохонена.

Асоціація (Associations) полягає у тому, що у ході розв'язання задачі пошуку асоціативних правил відшукуються закономірності між пов'язаними подіями в наборі даних.

Відмінність асоціації від двох попередніх задач Data Mining полягає в тому, що пошук закономірностей здійснюється не на основі властивостей аналізованого об'єкта, а між кількома подіями, які відбуваються одночасно.

Найбільш відомий алгоритм розв'язання задачі пошуку асоціативних правил – алгоритм Apriori.

Послідовність (Sequence), або послідовна асоціація (sequential association) дозволяє знайти тимчасові закономірності між транзакціями. Задача послідовності подібна асоціації, але її метою є встановлення закономірностей не між подіями, що наступають одночасно, а між подіями, які пов'язаними в часі (тобто відбуваються з деяким певним інтервалом у часі). Іншими словами послідовність визначається високою ймовірністю ланцюжка пов'язаних у часі подій.

Фактично, асоціація є окремим випадком послідовності з тимчасовим лагом, що дорівнює нулю. Цю задачу Data Mining також називають задачею знаходження послідовних шаблонів (sequential pattern).

Правило послідовності: після події X через певний час відбудеться подія Y.

Приклад. Після покупки квартири мешканці в 60% випадків протягом двох тижнів купують холодильник, а протягом двох місяців в 50% випадків купується телевізор. Розв'язок цієї задачі широко застосовується в маркетингу і менеджменті, наприклад, при управлінні циклом роботи з клієнтом (управління життєвим циклом клієнта).

У результаті розв'язання задачі ***прогнозування (Forecasting)*** на основі особливостей історичних даних оцінюються пропущені або ж майбутні значення цільових чисельних показників.

Для розв'язання таких задач широко застосовуються методи математичної статистики, нейронні мережі тощо.

Визначення відхилень або викидів (Deviation Detection) полягає у виявленні та аналізі даних, що найбільш відрізняються від загальної множини даних, виявленні так званих нехарактерних шаблонів.

Задача ***оцінювання*** зводиться до передбачення неперервних значень ознаки.

Аналіз зв'язків (Link Analysis) – задача знаходження залежностей в наборі даних.

У результаті ***візуалізації (Visualization, Graph Mining)*** створюється графічний образ аналізованих даних. Для розв'язання задачі візуалізації використовуються графічні методи, що показують наявність закономірностей у даних.

Приклад методу візуалізації – подання даних у 2-D і 3-D вимірах.

Підведення підсумків (Summarization) – задача, метою якої є опис конкретних груп об'єктів з аналізованого набору даних.

4.2 Класифікація завдань інтелектуального аналізу даних

Згідно класифікації за стратегіями, завдання Data Mining поділяються на такі групи:

– навчання з учителем;

– навчання без вчителя;

– інші.

Категорія «навчання з учителем» представлена такими завданнями Data Mining: класифікація, оцінка, прогнозування.

Категорія «навчання без вчителя» представлена завданням кластеризації.

До категорії «інші» належать завдання, не включені в попередні дві стратегії.

Завдання інтелектуального аналізу даних, залежно від використовуваних моделей, можуть бути **дескриптивними** і **прогнозуючими.**

Відповідно до цієї класифікації, **завдання Data Mining** представлені **групами описових і прогнозуючих задач.**

У результаті розв'язання описових (descriptive) завдань аналітик отримує шаблони, що описують дані, які піддаються інтерпретації.

Ці завдання описують загальну концепцію аналізованих даних, визначають інформативні, підсумкові, відмінні особливості даних. Концепція описових завдань передбачає характеристику і порівняння наборів даних.

Характеристика набору даних забезпечує короткий і стислий опис деякого набору даних.

Порівняння забезпечує порівняльний опис двох або більше наборів даних.

Прогнозуючі (predictive) завдання ґрунтуються на аналізі даних, створенні моделі, передбаченні тенденцій або властивостей нових або невідомих даних.

Досить близьким до вищезгаданої класифікації є розділення завдань Data Mining на такі:

– дослідження та відкриття;

– прогнозування та класифікація;

– пояснення й опис.

Автоматичне дослідження і відкриття (вільний пошук). *Приклад задачі: виявлення нових сегментів ринку.*

Для розв'язання цього класу завдань використовуються методи кластерного аналізу прогнозування та класифікація.

Приклад завдання: передбачення зростання обсягів продажів на основі поточних значень.

Методи: регресія, нейронні мережі, генетичні алгоритми, дерева рішень.

Завдання **класифікації та прогнозування** становлять групу так званого індуктивного моделювання, в результаті якого забезпечується вивчення аналізованого об'єкта або системи. У процесі вирішення цих завдань на основі набору даних розробляється загальна модель або гіпотеза.

Пояснення й опис. *Приклад завдання: характеристика клієнтів за демографічними даними і історіями покупок.*

Методи: дерева рішень, системи правил, правила асоціації, аналіз зв'язків.

Якщо дохід клієнта більше, ніж 50 умовних одиниць, і його вік – понад 30 років, тоді клас клієнта – перший.

В інтерпретації узагальненої моделі аналітик отримує нове знання. Групування об’єктів відбувається на основі їх подібності.

Нагадаємо, що головна цінність Data Mining – це практична спрямованість даної технології, шлях від сирих даних до конкретного знання, від постановки завдання до готового додатку, за підтримки якого можна приймати рішення.

Велика кількість понять, які об’єдналися в Data Mining, а також різноманітність методів, що підтримують дану технологію, аналітику-початківцю можуть нагадати мозаїку, частини якої мало пов’язані між собою.

Як же ми можемо зв’язати в одне ціле задачі, методи, дії, закономірності, додатки, дані, інформацію, рішення?

Розглянемо два потоки:

1. Дані – інформація – знання і рішення.

2. Завдання – дії і методи розв’язання – програми.

Ці потоки є «двома сторонами однієї медалі», відображенням одного процесу, результатом якого має бути знання і прийняття рішення.

Поняття «дані», «інформація» і «рішення», пов’язані між собою циклічним процесом. Прийняття рішень потребує інформації, яка заснована на даних. Дані забезпечують інформацію, яка підтримує рішення тощо.

Розглянуті поняття є складовою частиною так званої інформаційної піраміди, в основі якої знаходяться дані, наступний рівень – це інформація, потім йде рішення, завершує піраміду рівень знання. У міру просування вгору інформаційною пірамідою обсяги даних переходять у цінність рішень, тобто цінність для бізнесу. А, як відомо, метою Business Intelligence є перетворення обсягів даних у цінність бізнесу.

З іншого боку, три рівні аналізу (дані, інформація, знання) практично відповідають етапам еволюції аналізу даних, яка відбувалася протягом останніх років.

Верхній рівень – рівень додатків – є рівнем бізнесу (якщо ми маємо справу із завданням бізнесу), на ньому менеджери приймають рішення. Приклади додатків: перехресні продажі, контроль якості, утримування клієнтів.

Середній рівень – рівень дій – за своєю суттю є рівнем інформації, саме на ньому виконуються дії Data Mining; на рисунку наведені такі дії: прогностичне моделювання, аналіз зв’язків, сегментація даних та інші.

Нижній рівень – рівень визначення завдання інтелектуального аналізу даних, яке необхідно розв’язати стосовно даних, що є в наявності. Прикладами таких завдань є: завдання передбачення числових значень, класифікація, кластеризація, асоціація.

Нагадаємо, що для розв’язання задачі класифікації результати роботи першої стадії (індукції правил) використовуються для віднесення нового об’єкта з певною впевненістю до одного з відомих, визначених класів на підставі відомих значень.

Розглянемо завдання утримання клієнтів (визначення надійності клієнтів фірми).

Першим рівнем є дані – база даних за клієнтами. Є дані про клієнта (вік, стать, професія, дохід). Певна частина клієнтів, скориставшись продуктом фірми, залишилася їй вірною; інші клієнти більше не купували продукти фірми. На цьому рівні визначаємо тип задачі – це задача класифікації.

На другому рівні визначаємо дію – прогностичне моделювання. За допомогою прогностичного моделювання ми з певною частиною впевненості можемо віднести новий об'єкт, у цьому випадку, нового клієнта, до одного з відомих класів – постійний клієнт, або це, швидше за все, його разова покупка.

На третьому рівні ми можемо скористатися додатком для прийняття рішення. У результаті придбання знань, фірма може істотно знизити витрати, наприклад, на рекламу, знаючи заздалегідь, яким із клієнтів слід активно розсилати рекламні матеріали.

4.3 Інформація. Властивості інформації

Інформація (лат. informātiō) –

1) будь-які повідомлення про що-небудь;

2) відомості, що є об'єктом зберігання, переробки і передачі (наприклад, генетична інформація);

3) у математиці (кібернетиці) – кількісна міра усунення невизначеності (ентропія), міра організації системи; в теорії інформації – розділ кібернетики, що вивчає кількісні закономірності, які пов'язані зі збором, передачею, перетворенням і обчисленням інформації.

Інформація – будь-які, невідомі раніше відомості про якусь подію, сутності, процеси тощо, є об'єктом деяких операцій, для яких існує змістовна інтерпретація.

Під операціями тут мається на увазі сприйняття, передача, перетворення, зберігання і використання. Для сприйняття інформації необхідна деяка сприймаюча система, яка може інтерпретувати її, перетворювати, визначати відповідність певним правилам та інше. Отже, поняття інформації слід розглядати тільки при наявності джерела і одержувача інформації, а також каналу зв'язку між ними.

Властивості інформації:

– повнота інформації;
– достовірність інформації;
– цінність інформації;
– адекватність інформації;
– актуальність інформації;
– ясність інформації;
– доступність інформації;
– суб'єктивність інформації.

Властивість повноти інформації характеризує якість інформації і визначає достатність даних для прийняття рішень, тобто інформація повинна містити весь необхідний набір даних.

Приклад: «Продажі товару *А* почнуть скорочуватися». Ця інформація неповна, оскільки невідомо, коли саме вони почнуть скорочуватися.

Приклад повної інформації: «Починаючи з першого кварталу, продажі товару *А* почнуть скорочуватися». Цієї інформації достатньо для прийняття рішень;

Інформація може бути достовірною і недостовірною. У недостовірній інформації присутній інформаційний шум, і чим він вищий, тим нижче достовірність інформації;

Цінність інформації не може бути абстрактною. Інформація повинна бути корисною і цінною для певної категорії користувачів;

Властивість адекватності інформації характеризує ступінь відповідності інформації реальному об'єктивному стану. Адекватна інформація – це повна і достовірна інформація;

Інформація повинна бути актуальною, тобто НЕ застарілою. Ця властивість інформації характеризує ступінь відповідність інформації справжньому моменту часу;

Інформація повинна бути зрозуміла для того кола осіб, для якого вона призначена;

Доступність характеризує міру можливості отримати певну інформацію. На цю властивість інформації впливають одночасно доступність даних і доступність адекватних методів;

Інформація носить суб'єктивний характер, вона визначається ступенем сприйняття суб'єкта (одержувача інформації).

Вимоги, що пред'являються до інформації:

– динамічний характер інформації: інформація існує тільки в момент взаємодії даних і методів, тобто в момент інформаційного процесу, решту часу вона перебуває в стані даних;

– адекватність використовуваних методів: **і**нформація витягується з даних, проте в результаті використання одних і тих даних може з'являтися різна інформація (це залежить від адекватності вибраних методів обробки вихідних даних).

Дані, за своєю суттю, є об'єктивними. ***Методи*** є суб'єктивними, в основі методів лежать алгоритми, суб'єктивно складені та підготовлені. Отже, інформація виникає та існує в момент діалектичної взаємодії об'єктивних даних і суб'єктивних методів.

Для бізнесу інформація є вихідною складової прийняття рішень.

Всю інформацію, що виникає в процесі функціонування бізнесу та управління ним, можна класифікувати певним чином. Залежно від джерела одержання, інформацію поділяють на внутрішню і зовнішню (наприклад, інформація, що описує явища, які відбуваються за межами фірми, але мають до неї безпосереднє відношення).

Також інформація може бути класифікована на фактичну і прогнозну. До фактичної інформації про бізнес відноситься інформація, що характеризує доконаний факт, вона є точною. Прогнозна інформація розраховується або

передбачається, тому її не можна вважати точною, вона може мати певну похибку.

Знання **–** сукупність фактів, закономірностей і евристичних правил, за допомогою яких вирішується поставлене завдання.

Отже, формування інформації відбувається в процесі збору та передачі, тобто обробки даних. Яким же чином із інформації отримують знання?

Усе частіше істинні знання утворюються на основі розподілених взаємозв'язків різнорідної інформації. Коли інформація зібрана і передана для отримання явно не визначеного заздалегідь результату, то ви отримуєте знання. Сама по собі інформація в чистому вигляді безглузда. Звідси випливає висновок, що інформація – це чиєсь тактичне знання, передане у вигляді символів і за допомогою будь-яких прикладних засобів.

За визначенням Денхема Грея, «знання – це абсолютне використання інформації і даних, спільно з потенціалом практичного досвіду людей, здібностями, ідеями, інтуїцією, переконаністю і мотиваціями».

Властивостями, які відрізняють знання від інформації є:

– структурованість;

– зручність доступу і засвоєння;

– лаконічність;

– несуперечливість;

– процедури обробки.

Структурованість знань полягає у тому, що знання повинні бути «розкладені по поличках».

Зручність доступу та засвоєння для людини означає здатність швидко зрозуміти і запам'ятати або, навпаки, згадати, для комп'ютерних знань – засоби доступу до знань.

Лаконічність дозволяє швидко освоювати і переробляти знання і підвищує «коефіцієнт корисного змісту». У цей список лаконічність була додана через усім відому проблему шуму і сміттєвих документів, характерних саме для комп'ютерної інформації – Інтернету та електронного документообігу.

Знання не повинні суперечити одне одному.

Знання потрібні для того, щоб їх використовувати. Одна з головних властивостей знань – можливість їх передачі іншим і здатність робити висновки на їх основі. Для цього повинні існувати процедури обробки знань. Здатність робити висновки означає для машини наявність процедур обробки та виведення і підготовленість структур даних для такої обробки, тобто наявність спеціальних форматів знань.

Для того, щоб впевнено оперувати поняттями «інформація», «дані», «знання», необхідно не тільки розуміти суть цих понять, а й відчути відмінності між ними. Однак, однієї інтуїтивної інтерпретації цих понять тут недостатньо. Складність розуміння відмінностей вищезазначених понять – в їх уявній синонімічності. Згадаймо, що поняття Data Mining переводиться на українську мову за допомогою цих же трьох понять: як видобуток даних, вилучення інформації, розкопування знань.

Для початку зробимо спробу розібратися в цих термінах на простих прикладах:

1. Студент, який здає іспит, потребує даних.
2. Студент, який здає іспит, потребує інформації.
3. Студент, який здає іспит, потребує знань.

При розгляді першого варіанту – студент потребує даних – виникає думка, що студенту потрібні дані, наприклад, для обчислень. Інформацією в другому варіанті може виступати конспект або підручник. У результаті їх використання студент отримує лише інформацію, яка в певних випадках може перейти у знання. Третій варіант звучить найбільш логічно.

Поняття «інформація» і «знання», з філософської точки зору, є поняттями більш високого рівня, ніж «дані», яке виникло відносно недавно.

Поняття «інформації» безпосередньо пов'язано із сутністю процесів усередині інформаційної системи, тоді як поняття «знання» швидше орієнтоване на якість процесів. Поняття «знання» тісно пов'язане з процесом прийняття рішень.

Незважаючи на відмінності, розглянуті поняття, як уже зазначалося раніше, не є розрізненими і непов'язаними. Вони є частиною одного потоку: біля витоку його знаходяться дані, у процесі передачі яких виникає інформація, і в результаті використання інформації, за певних умов, виникають знання.

У процесі руху вгору в інформаційній піраміді обсяги даних переходять у цінність знань. Однак великі обсяги даних зовсім не означають і, тим більше, не гарантують отримання знань. Існує певна залежність цінності отриманих знань від якості та потужності процедур обробки даних. Типовим прикладом інформації, яку не можна перетворити в знання, є текст іноземною мовою. За відсутності словника і перекладача ця інформація взагалі не має цінності, вона не може перейти в знання. За наявності словника процес переходу від інформації до знання можливий, але тривалий і трудомісткий. За наявності перекладача інформація дійсно переходить в знання.

Таким чином, для отримання цінних знань необхідні якісні процедури обробки. Процес переходу від даних до знань займає багато часу і коштує дорого. Тому очевидно, що технологія Data Mining з її потужними і різноманітними алгоритмами є інструментом, за допомогою якого, просуваючись вгору інформаційною піраміді, ми можемо отримувати дійсно якісні та цінні знання.

Питання для самоконтролю

1. Дайте визначення класифікації (Classification).
2. Дайте визначення кластеризації (Clustering).
3. На які групи поділяють задачі Data Mining?
4. Який зв'язок між поняттями «дані», «інформація» і «рішення»? Намалюйте схему їх зв'язку.
5. Які рівні аналізу ви знаєте?
6. Які властивості мають знання?

ЗМІСТОВИЙ МОДУЛЬ 3. МЕТОД ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ

Тема 5. Метод пошуку асоціативних правил

Мета: ознайомитися з правилами та напрямками застосування асоціативних правил.

План

5.1 Визначення асоціативних правил
5.2 Алгоритми пошуку асоціативних правил
5.3 Методи пошуку асоціативних правил

Основні поняття

Транзакційна база даних. Характеристики асоціативних правил. Алгоритм Apriori.

5.1 Визначення асоціативних правил

Асоціація **–** одна із задач Data Mining. Метою пошуку асоціативних правил (association rule) є знаходження закономірностей між зв'язаними подіями в базах даних.

Дуже часто покупці купують не один товар, а декілька. У більшості випадків між цими товарами існує взаємозв'язок. Так, наприклад, покупець, що купує ноутбук, швидше за все, захоче придбати також сумку. Ця інформація може бути використана для розміщення товару на прилавках.

Асоціативні правила, часто знаходять застосування:

– у аналізі Web-блогів;

– у роздрібній торгівлі: визначення товарів, які варто просувати спільно; вибір місця розташування товару в магазині; аналіз споживчого кошика; прогнозування попиту;

– у перехресних продажах: якщо є інформація про те, що клієнти придбали продукти А, Б і В, то які з них найімовірніше куплять продукт Г;

– у маркетингу: пошук ринкових сегментів, тенденцій купівельної поведінки;

– для сегментації клієнтів: виявлення загальних характеристик клієнтів компанії, виявлення груп покупців;

– під час оформлення каталогів, при аналізі збутових кампаній фірми, визначенні послідовностей покупок клієнтів (яка покупка піде за покупкою товару А).

Простим прикладом асоціативного правила є: покупець, що купує ноутбук, придбає до нього мишку з імовірністю 50%.

Уперше задача пошуку асоціативних правил (association rule mining) була запропонована для знаходження типових шаблонів покупок, здійснених у супермаркетах, тому іноді її ще називають аналізом ринкового кошика (market basket analysis).

Ринковий кошик – це набір товарів, придбаних покупцем у рамках однієї окремо взятої транзакції.

Транзакції є досить характерними операціями, ними, наприклад, можуть описуватися результати відвідувань різних магазинів.

Транзакція – це множина подій, які відбулися одночасно.

Реєструючи всі бізнес-операції протягом усього часу своєї діяльності, торговельні компанії накопичують величезні кількості транзакцій. Кожна така транзакція являє собою набір товарів, куплених покупцем за один візит.

Отримані в результаті аналізу шаблони включають перелік товарів і число транзакцій, які містять дані набори.

Транзакційна або операційна база даних (Transaction database) являє собою двовимірну таблицю, яка складається з номера транзакції (TID) і переліку покупок, придбаних під час цієї транзакції.

TID – унікальний ідентифікатор, що визначає кожну угоду або транзакцію.

Приклад транзакційної бази даних, що складається з купівельних транзакцій, наведено в таблиці 5.1. У таблиці перший стовпчик (TID) визначає номер транзакції, у другому стовпчику таблиці наведені товари, придбані під час певної транзакції**.**

На основі наявної бази даних нам потрібно знайти закономірності між подіями, тобто покупками.

Таблиця 5.1 − Транзакційна база даних

TID	Покупки
100	Карта пам’яті, DVD-диск, USB-подовжувач
200	DVD-диск, USB-подовжувач, WEB-камера
300	Комп’ютерна миша, DVD-диск, WEB-камера
400	USB-подовжувач, DVD-диск, Карта пам’яті, Комп’ютерна миша
500	DVD-диск, Карта пам’яті
600	VGA-кабель

Розглянемо шаблони, що часто зустрічаються, або зразки. Допустимо, є транзакційна база даних D.

Присвоємо значенням товарів змінні (таблиця 5.2).

Карта пам’яті = *a*

DVD-диск = *b*

USB-подовжувач = *c*

Комп’ютерна миша = *d*

WEB-камера = *e*

VGA-кабель = *f*

Розглянемо набір товарів (Itemset), що включає, наприклад, (флешка, DVD-диск, USB-подовжувач). Виразимо цей набір за допомогою змінних:

$$abc=\{a,b,c\}$$

Цей набір товарів зустрічається в нашій базі даних три рази, тобто підтримка цього набору товарів рівна 3:

$$SUP(abc)=3.$$

Таблиця 5.2 − Набори товарів, що часто зустрічаються

TID	Покупки	TID	Покупки
100	Флешка, DVD-диск, USB-подовжувач	100	*a, b, c*
200	DVD-диск, USB-подовжувач, WEB-камера	200	*b, c, e*
300	Комп'ютерна миша, DVD-диск, WEB-камера	300	*d, b, e*
400	USB- подовжувач, DVD-диск, Флешка, Комп'ютерна миша	400	*c, b, a, d*
500	DVD-диск, Флешка, USB-подовжувач	500	*b, a, c*
600	VGA-кабель	600	*f*

При мінімальному рівні підтримки, що дорівнює трьом, набір товарів *abc* є шаблоном, що часто зустрічається.

min_sup=3, {Карта пам'яті, DVD-диск, USB-подовжувач} – частий шаблон, що зустрічається.

Підтримкою називають кількість або відсоток транзакцій, що містять певний набір даних.

Для даного набору товарів підтримка, виражена у відсотковому відношенні, дорівнює 50%.

$$SUP(abc)=(3/6)*100\%=50\%$$

Підтримку іноді також називають забезпеченням набору.

Отже, набір становить інтерес, якщо його підтримка вище заданого користувачем мінімального значення (min support). Ці набори називають такими, що часто зустрічаються (frequent**).**

5.2 Алгоритми пошуку асоціативних правил

Асоціативне правило має вигляд: «з події А випливає подія В». У результаті такого виду аналізу встановлюємо закономірність такого виду: «Якщо в транзакції зустрівся набір товарів (або набір елементів) А, то можна зробити висновок, що в цій же транзакції повинен з'явитися набір елементів В)». Встановлення таких закономірностей дає можливість знаходити дуже прості й зрозумілі правила, які називають асоціативними.

Основними характеристиками асоціативного правила є підтримка й вірогідність правила.

Розглянемо правило «з покупки флешки випливає покупка USB-подовжувача» для бази даних, яка була наведена вище в таблиці 5.1. Поняття підтримки набору вже розглянуто. Існує поняття підтримки правила.

Правило має підтримку *S*, якщо *s*% транзакцій із усього набору містять одночасно набори елементів A і B або, інакше кажучи, містять обидва товари.

Флешка – це товар A, USB-подовжувач – це товар B. Підтримка правила «з покупки флешки випливає покупка USB-подовжувача» рівна 3, або 50%.

Вірогідність правила показує, яка ймовірність того, що з події A випливає подія B.

Правило «з A випливає B» справедливе з вірогідністю *C*, якщо *c*% транзакцій з усієї множини, що містить набір елементів A, також містять набір елементів B.

Якщо кількість транзакцій, що містять USB-подовжувач, дорівнює чотирьом, а кількість транзакцій, що містять також і флешку, дорівнює трьом, то вірогідність правила дорівнює (3/4)*100%, тобто 75%.

Вірогідність правила «з покупки USB-подовжувача випливає покупка флешки» 75%, тобто 75% транзакцій, що містять товар A, також містять товар B.

За допомогою використання алгоритмів пошуку асоціативних правил аналітик може одержати всі можливі правила вигляду «з A випливає B», з різними значеннями підтримки й вірогідності. Однак у більшості випадків, кількість правил необхідно обмежувати заздалегідь установленими мінімальними й максимальними значеннями підтримки й вірогідності.

Якщо значення підтримки правила занадто велике, то в результаті роботи алгоритму будуть знайдені правила очевидні й добре відомі. Занадто низьке значення підтримки призведе до знаходження дуже великої кількості правил, які, можливо, будуть у більшості необґрунтованими, але не відомими й не очевидними для аналітика. Отже, необхідно визначити такий інтервал («золоту середину»), який з одного боку забезпечить знаходження неочевидних правил, а з іншого – їх обґрунтованість.

Якщо рівень вірогідності занадто малий, то цінність правила викликає серйозні сумніви. Наприклад, правило з вірогідністю в 3% тільки умовно можна назвати правилом.

5.3 Методи пошуку асоціативних правил

Перший алгоритм пошуку асоціативних правил, що називався ***AIS***, (запропонований Agrawal, Imielinski and Swami) був розроблений співробітниками дослідного центру IBM Almaden у 1993 році. Із цієї роботи виник інтерес до асоціативних правил; на середину 90-х років минулого століття припадає пік дослідницьких робіт у цій області, і з того часу щороку з’являється кілька нових алгоритмів.

В алгоритмі AIS кандидати множини наборів генеруються й підраховуються «на льоту», під час сканування бази даних.

Створення алгоритму ***SETM*** алгоритму було мотивовано бажанням використовувати мову SQL для обчислення наборів товарів, що часто зустрічаються. Як і алгоритм AIS, SETM також формує кандидатів «на льоту»,

ґрунтуючись на перетвореннях бази даних. Щоб використовувати стандартну операцію об'єднання мови SQL для формування кандидата, SETM відокремлює формування кандидата від їхнього підрахунку.

Незручність алгоритмів AIS і SETM – надмірне генерування й підрахунок занадто багатьох кандидатів, які в результаті не є такими, що часто зустрічаються. Для поліпшення їх роботи був запропонований алгоритм ***Apriori***.

Робота даного алгоритму складається з декількох етапів, кожен з яких складається з таких кроків:

– формування кандидатів;

– підрахунок кандидатів.

Формування кандидатів (candidate generation) – етап, на якому алгоритм, скануючи базу даних, створює множину *i*-елементних кандидатів (*i* – номер етапу). На цьому етапі підтримка кандидатів не розраховується.

Підрахунок кандидатів (candidate counting) – етап, на якому обчислюється підтримка кожного *i*-елементного кандидата. Тут же здійснюється відсікання кандидатів, підтримка яких менша мінімуму, установленого користувачем (min_sup).

Решту *i*-елементних наборів називаємо такими, що часто зустрічаються.

Розглянемо роботу алгоритму Apriori на прикладі бази даних D. Ілюстрація роботи алгоритму наведена на рис. 5.1 Мінімальний рівень підтримки рівний 3.

На першому етапі відбувається формування одноелементних кандидатів. Далі алгоритм підраховує підтримку одноелементних наборів. Набори з рівнем підтримки менше встановленого, тобто 3, відкидаються. У цьому прикладі це набори *e* і *f*, які мають підтримку, що дорівнює 1. Набори товарів, що залишилися, вважаються одноелементними наборами товарів, що часто зустрічаються, – це набори *a, b, c, d*.

Далі відбувається формування двоелементних кандидатів, підрахунок їх підтримки й відсікання наборів з рівнем підтримки, меншим 3. Двоелементні набори товарів, що залишилися, вважаються двоелементними наборами, що часто зустрічаються, *ab, ac, bd*, беруть участь у подальшій роботі алгоритму.

Якщо дивитися на роботу алгоритму прямолінійно, на останньому етапі алгоритм формує трьохелементні набори товарів: *abc, abd, bcd, acd*, підраховує їхню підтримку й відтинає набори з рівнем підтримки, меншим 3. Набір товарів *abc* може бути названий таким, що часто зустрічається.

Однак алгоритм Apriori зменшує кількість кандидатів, відсікаючи – apriori – тих, які свідомо не можуть стати такими, що часто зустрічаються, на основі інформації про відсічених кандидатів на попередніх етапах роботи алгоритму.

Відсікання кандидатів відбувається на основі припущення про те, що в наборі товарів, що часто зустрічаються, усі підмножини повинні бути такими, що часто зустрічаються. Якщо в наборі наявна підмножина, яку на

попередньому етапі було визначено такою, що нечасто зустрічається, цей кандидат уже не включається у формування й підрахунок кандидатів.

Так набори товарів *ad, bc, cd* були відкинуті як такі, що нечасто зустрічаються, алгоритм не розглядав набори товарів *abd, bcd, acd*.

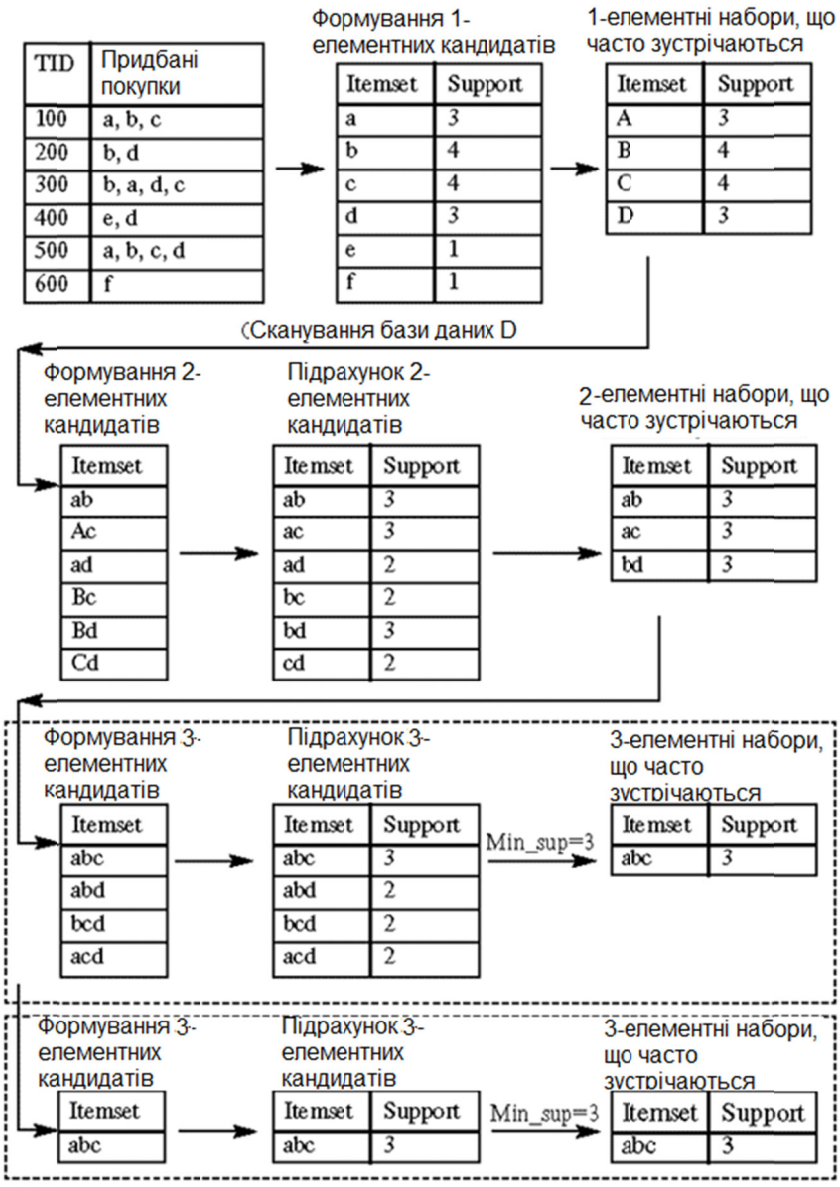

Рисунок 5.1 – Алгоритм Apriori

При розгляді цих наборів формування трьохелементних кандидатів відбувалося б за схемою, наведеною у верхньому пунктирному прямокутнику. Оскільки алгоритм апріорі відкинув набори, що свідомо нечасто зустрічаються, останній етап алгоритму відразу визначив набір *abc* як єдиний триелементний набір, що часто зустрічається (етап наведений у нижньому пунктирному прямокутнику).

Алгоритм Apriori розраховує також підтримку наборів, які не можуть бути відсічені апріорі. Це так звана негативна область (negative border), до неї належать набори-кандидати, які зустрічаються рідко, їх самих не можна

віднести до таких, що часто зустрічаються, але всі підмножини даних наборів є такими, що часто зустрічаються.

Залежно від розміру найдовшого набору, що часто зустрічається, алгоритм Apriori сканує базу даних певну кількість разів. Різновиди алгоритму Apriori, що є його оптимізацією, запропоновані для скорочення кількості сканувань бази даних, кількості наборів-кандидатів або того й іншого. Були запропоновані такі різновиди алгоритму Apriori: Aprioritid і Apriorihybrid.

Цікава особливість алгоритму ***Aprioritid*** – те, що база даних D не використовується для підрахунку підтримки кандидатів набору товарів після першого проходу. Із цією метою використовується кодування кандидатів, виконане на попередніх проходах. У наступних проходах розмір закодованих наборів може бути набагато меншим, ніж база даних, і в такий спосіб заощаджуються значні ресурси.

Аналіз часу роботи алгоритмів Apriori і Aprioritid показує, що в більш ранніх проходах Apriori досягає більшого успіху, ніж Aprioritid; однак Aprioritid працює краще Apriori у більш пізніх проходах. Крім того, вони використовують ту саму процедуру формування наборів-кандидатів. Заснований на цьому спостереженні алгоритм ***Apriorihybrid*** запропонований, щоб об'єднати кращі властивості алгоритмів Apriori і Aprioritid. Apriorihybrid використовує алгоритм Apriori у початкових проходах і переходить до алгоритму Aprioritid, коли очікується, що закодований набір первісної множини наприкінці проходу буде відповідати можливостям пам'яті. Однак перемикання від Apriori до Aprioritid вимагає залучення додаткових ресурсів.

Алгоритм ***DHP***, або алгоритм хешування (J. Park, M. Chen and P. Yu) було запропоновано у 1995 році. В основі його роботи – імовірнісний підрахунок наборів-кандидатів, що здійснюється для скорочення кількості підрахованих кандидатів на кожному етапі виконання алгоритму Apriori. Скорочення забезпечується за рахунок того, що кожний з k-елементних наборів-кандидатів крім кроку скорочення проходить крок хешування. В алгоритмі на k-1 етапі під час вибору кандидата створюється так звана хеш-таблиця. Кожний запис хеш-таблиці є лічильником усіх підтримок k-елементних наборів, які відповідають цьому запису в хеш-таблиці. Алгоритм використовує цю інформацію на етапі k для скорочення множини k-елементних наборів-кандидатів. Після скорочення підмножини, як це відбувається в Apriori, алгоритм може вилучити набір-кандидат, якщо його значення в хеш-таблиці менше граничного значення, установленого для забезпечення.

До інших удосконалених алгоритмів відносяться: ***PARTITION***, ***DIC***.

Алгоритм PARTITION (A. Savasere, E. Omiecinski and S. Navathe) розроблено у 1995 році. Цей алгоритм розбивки (поділу) полягає в скануванні транзакційної бази даних шляхом поділу її на розділи, які не перетинаються, кожний з яких може вміститися в оперативній пам'яті. На першому кроці в кожному з розділів за допомогою алгоритму Apriori визначаються «локальні» набори даних, що часто зустрічаються. На другому підраховується підтримка

кожного такого набору щодо всієї бази даних. Отже, на другому етапі визначається множина усіх потенційних наборів даних, що зустрічаються.

Алгоритм DIC, Dynamic Itemset Counting (S. Brin R. Motwani, J. Ullman and S. Tsur) запропоновано у 1997 році. Алгоритм розбиває базу даних на кілька блоків, кожний з яких відзначається так званими «початковими точками» (start point), і потім циклічно сканує базу даних.

Питання для самоконтролю

1. В якому випадку, на Ваш погляд, виникають асоціації?
2. Який зміст вкладається в поняття «вірогідності асоціативного правила»?
3. Для розв'язання яких задач можна застосовувати інструмент Data Mining асоціативні правила?
4. Які існують методи пошуку асоціативних правил?
5. Визначте сутність алгоритму Apriori.

Лабораторне заняття №3

Тема: Пошук асоціативних правил на мові R

Мета роботи: набути навички використання алгоритму apriori на мові R.

Завдання. Виконати пункти 1-8 зазначених в Хід роботи.

Хід роботи.

1. Підключіть необхідні для роботи бібліотеки (arules, arulesViz).

2. Завантажте з Moodle до R дані з інформацією про "ринковий кошик" з файлу groceries.csv (команда read.transaction()) та створіть об'єкт itemMatrix, зверніть увагу на особливості структури файлу з транзакціями.

3. Отримайте інформацію про сформований масив транзакцій (команди inspect() та summary()).

4. Побудуйте графік розподілу частот зустрічальності ознак (*itemFrequencyPlot(ім'я змінної, support=__ ,cex.names=__*), оберіть значення підтримки та розміру підписів такими, що найкраще візуалізують дані.

5. Формування правил здійснюється функцією apriori() з зазначенням порогових значень підтримки та достовірності:

rules<- apriori(ім'я змінної, parameter=list(support=__, confidence=__)).

Функція summary() забезпечує частотний аналіз правил за їх довжиною та досягнутими мірами якості.

6. Функція plot() з пакету arulesViz дозволяє отримувати різні форми візуалізації синтезованих правил, у тому числі, аналіз мінливості їх мір якості:

```
library(arulesViz)
plot(rules, measure=c("support", "lift"), shading="confidence")
```

7. Для вирішення завдання виявлення характерних особливостей груп товарів, нас цікавлять, у першу чергу, високоякісні правила, що мають відповідну ознаку групи у правій частині. Тоді товар можна буде легко впізнати за його оточенням:

rules_назва товару<-subset(rules, subset=rhs %in% "назва товару" & lift>задане значення параметра)
inspect(head(rules_назва товару, n=кількість правил, які необхідно вивести, by="support"))
plot(head(sort(rules_назва товару, by="support", кількість правил), method ="paracoord")

Графік у паралельних координатах (*method ="paracoord"*) показує, як формуються комбінації ознак правої частини при зростанні її розміру, а товщина ліній відповідає рівню підтримки.

8. Метод "graph" функції plot() показує правила та складові їх ознаки у вигляді графа, розмір вузлів якого пропорційний рівню підтримки кожного наявного правила:

plot(head(sort(rules_назва товару, by="support", кількість правил), method ="graph ", control=list(nodeCol=grey.color(10), edgeCol=grey(0.7), alpha=1))

ЗМІСТОВИЙ МОДУЛЬ 4. МЕТОДИ КЛАСИФІКАЦІЇ ТА КЛАСТЕРИЗАЦІЇ

Тема 6. Метод кластерного аналізу

Мета: ознайомитися з методами кластерного аналізу та місцем їх застосування.

План

6.1 Кластерний аналіз
6.2 Методи кластерного аналізу
6.3 Ієрархічний кластерний аналіз
6.4 Алгоритми неієрархічної кластеризації
6.5 Ітеративні методи кластеризації
6.6 Порівняльний аналіз ієрархічних і неієрархічних методів кластеризації

Основні поняття

Кластеризація. Квадрат евклідової відстані. Манхеттінська відстань. Відстань Чебишева. Алгоритм k-середніх.

6.1 Кластерний аналіз

Термін кластерний аналіз, уперше введений Тріоном (Tryon) у 1939 році, містить більш 100 різних алгоритмів.

На відміну від задач класифікації, кластерний аналіз не вимагає апріорних припущень про набір даних, не накладає обмеження на показ досліджуваних об'єктів, дозволяє аналізувати показники різних типів даних (інтервальні дані, частоти, бінарні дані). При цьому необхідно пам'ятати, що змінні повинні вимірюватися в порівнюваних шкалах.

Кластерний аналіз дозволяє скорочувати розмірність даних, робити їх наглядними. Кластерний аналіз може застосовуватися до сукупностей тимчасових рядів, тут можуть виділятися періоди схожості деяких показників і визначатися групи тимчасових рядів зі схожою динамікою.

Кластерний аналіз паралельно розбудовувався в декількох напрямках, таких як біологія, психологія тощо, тому більшість методів мають по дві й більш назв. Це суттєво ускладнює роботу при використанні кластерного аналізу.

Задачі кластерного аналізу можна об'єднати в такі групи:

1. Розробка типології або класифікації.
2. Дослідження корисних концептуальних схем групування об'єктів.
3. Представлення гіпотез на основі дослідження даних.
4. Перевірка гіпотез або досліджень для визначення, чи дійсно типи (групи), виділені тим або іншим способом, присутні в наявних даних.

Як правило, при практичному використанні кластерного аналізу одночасно розв'язуються декілька із зазначених задач.

Розглянемо приклад процедури кластерного аналізу.

Допустимо, маємо набір даних A, що складається з 14-ти прикладів, у яких є по дві ознаки *X* і *Y*. Дані в табличній формі не носять інформативний характер. Представимо змінні *X* і *Y* у вигляді діаграми розсіювання (рис. 6.1).

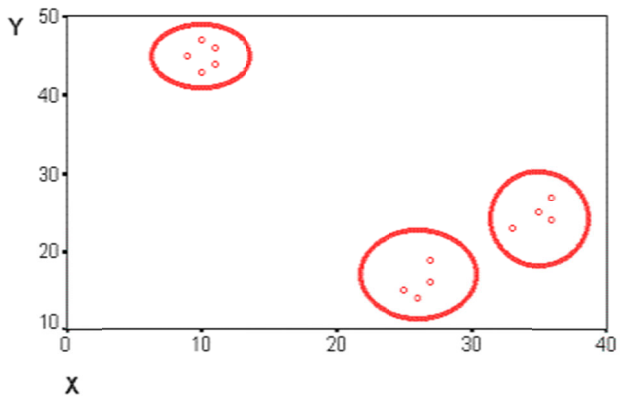

Рисунок 6.1 – Діаграма розсіювання змінних X і Y

На рисунку бачимо кілька груп «схожих» прикладів. Приклади (об'єкти), які за значеннями *X* і *Y* «схожі» один на одного, належать до однієї групи (кластеру); об'єкти з різних кластерів не схожі один на одного.

Критерієм для визначення схожості й відмінності кластерів є відстань між точками на діаграмі розсіювання. Цю подібність можна «виміряти», вона дорівнює відстані між точками на графіку. Способів визначення міри відстані між кластерами, яку називають ще мірою близькості, існує небагато. Найпоширеніший спосіб – обчислення евклідової відстані між двома точками *i* та *j* на площині, коли відомі їхні координати *X* і *Y*:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{6.1}$$

Якщо нам потрібно знайти відстань між двома точками в просторі трьох вимірів (рис. 6.2), формула (6.1) набуває вигляду:

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}, \tag{6.2}$$

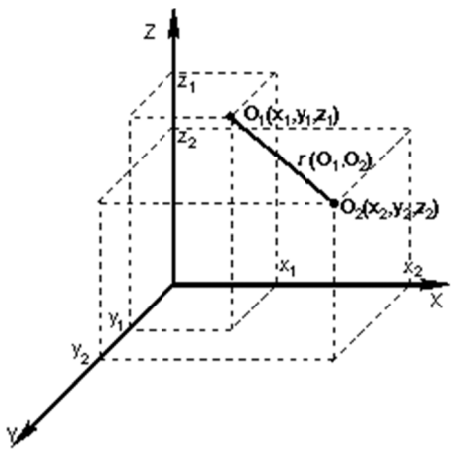

Рисунок 6.2 – Відстань між двома точками в просторі трьох вимірів

Кластер має такі математичні характеристики: центр, радіус, середньоквадратичне відхилення, розмір кластера.

Центр кластера – це середнє геометричне місце точок у просторі змінних.

Радіус кластера – максимальна відстань точок від центру кластера.

У випадку перекриття кластерів неможливо за допомогою математичних процедур однозначно віднести об'єкт до одного з двох кластерів. Такі об'єкти називають спірними.

Спірний об'єкт – це об'єкт, який у міру подібності може бути віднесений до декількох кластерів.

Розмір кластера може бути визначений або за радіусом кластера, або за середньоквадратичним відхиленням об'єктів для цього кластера. Об'єкт належить до кластера, якщо відстань від об'єкта до центру кластера менше радіуса кластера. Якщо ця умова виконується для двох і більш кластерів, об'єкт є спірним. Неоднозначність може бути усунута експертом або аналітиком.

Робота кластерного аналізу опирається на два припущення. Перше припущення – розглянуті ознаки об'єкта в принципі допускають бажане розбиття сукупності об'єктів на кластери. Друге припущення – правильність вибору масштабу або одиниці вимірювання ознак.

Вибір масштабу в кластерному аналізі має велике значення. Розглянемо приклад. Уявимо собі, що дані ознаки x у наборі даних A на два порядки більші даних ознаки y: значення змінної x перебувають в діапазоні від 100 до 700, а значення змінної y – у діапазоні від 0 до 1.

Тоді, при розрахунках величини відстані між точками, що відображають положення об'єктів у просторі їх властивостей, змінна, що має більші значення, тобто змінна x, буде практично повністю домінувати над змінною з малими значеннями, тобто змінної y. У такий спосіб через неоднорідність одиниць виміру ознак стає неможливим коректно розрахувати відстані між точками.

Ця проблема вирішується за допомогою попередньої стандартизації змінних. Стандартизація (standardization) або нормування (normalization) приводить значення всіх перетворених змінних до єдиного діапазону значень шляхом вираження через відношення цих значень до якоїсь величини, що відображає певні властивості конкретної ознаки. Існують різні способи нормування вихідних даних.

Два найпоширеніші способи:

– розподіл вихідних даних на середньоквадратичне відхилення відповідних змінних;

– обчислення Z-внеску або стандартизованого внеску.

Поряд зі стандартизацією змінних, існує варіант додавання до кожної з них певного коефіцієнта важливості, або ваги, яка би відображала значимість відповідної змінної. За ваги можуть виступати експертні оцінки, отримані в ході опитування експертів – фахівців предметної області. Отримані добутки нормованих змінних на відповідні ваги дозволяють одержувати відстані між точками в багатомірному просторі з урахуванням неоднакової ваги змінних.

У ході експериментів можливе порівняння результатів, отриманих з урахуванням експертних оцінок і без них, і вибір якіснішого з них.

6.2 Методи кластерного аналізу

Методи кластерного аналізу можна розділити на дві групи:

– ієрархічні;

– неієрархічні.

Кожна із груп включає багато підходів і алгоритмів.Використовуючи різні методи кластерного аналізу, аналітик може одержати різні розв'язки для тих самих даних. Це вважається нормальним явищем.

Розглянемо ієрархічні й неієрархічні методи докладно.

Ієрархічні методи кластерного аналізу. Суть ієрархічної кластеризації полягає в послідовному об'єднанні менших кластерів у більші або поділі більших кластерів на менші.

Ієрархічні агломеративні методи (Agglomerative Nesting, AGNES). Ця група методів характеризується послідовним об'єднанням вихідних елементів і відповідним зменшенням числа кластерів.

На початку роботи алгоритму всі об'єкти є окремими кластерами. На першому кроці найбільш схожі об'єкти поєднуються в кластер. На наступних кроках об'єднання триває доти, поки всі об'єкти не будуть становити один кластер.

Ієрархічні дивизимні (ділені) методи (Divisive Analysis, DIANA). Ці методи є логічною протилежністю агломеративним методам. На початку роботи алгоритму всі об'єкти належать одному кластеру, який на наступних кроках ділиться на менші кластери, у результаті утворюється послідовність груп, що розщеплюються. Принцип роботи описаних вище груп методів у вигляді дендрограми показаний на рис. 6.3.

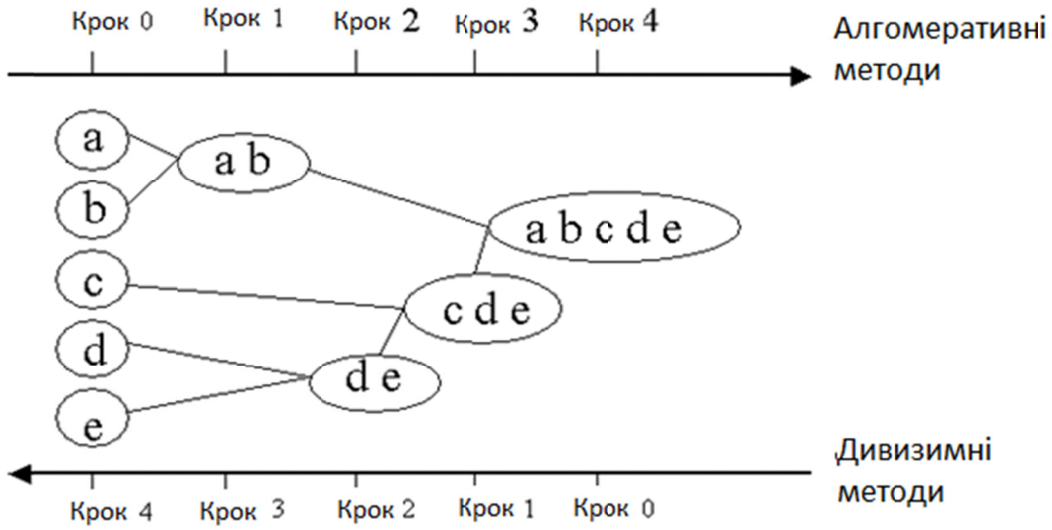

Рисунок 6.3 – Дендрограма агломеративних і дивизимних методів

Програмна реалізація алгоритмів кластерного аналізу широко представлена в різних інструментах Data Mining, які дозволяють вирішувати завдання досить великої розмірності. Наприклад, агломеративні методи реалізовані в пакеті SPSS, дивизимні методи – у пакеті Statgraf.

Ієрархічні методи кластеризації різняться правилами побудови кластерів. За правила виступають критерії, які використовуються при вирішенні питання про «схожість» об'єктів при об'єднанні їх в групу (агломеративні методи) або поділу на групи (дивизимні методи).

Ієрархічні методи кластерного аналізу використовуються при невеликих обсягах наборів даних.

Перевагою ієрархічних методів кластеризації є їхня наочність.

Ієрархічні алгоритми пов'язані з побудовою дендрограм (від грецького dendron – «дерево»), які є результатом ієрархічного кластерного аналізу.

Дендрограма описує близькість окремих точок і кластерів один до одного, представляє в графічному вигляді послідовність об'єднання (поділу) кластерів.

Дендрограма (dendrogram) – деревоподібна діаграма, що містить *n* рівнів, кожний з яких відповідає одному з кроків процесу послідовного укрупнення кластерів. Дендрограму також називають деревоподібною схемою, деревом об'єднання кластерів, деревом ієрархічної структури.

Дендрограма являє собою вкладене угруповання об'єктів, яке змінюється на різних рівнях ієрархії.

Існує багато способів побудови дендограмм. У дендограмі об'єкти можуть розташовуватися вертикально або горизонтально. Приклад вертикальної дендрограми наведений на рис. 6.4.

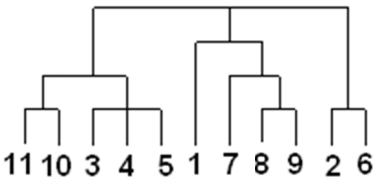

Рисунок 6.4 – Приклад дендрограмми

Числа 11, 10, 3 і т.д. відповідають номерам об'єктів або спостережень вихідної вибірки. Бачимо, що на першому кроці кожне спостереження представляє один кластер (вертикальна лінія), на другому кроці спостерігаємо об'єднання таких спостережень: 11 і 10; 3, 4 і 5; 8 і 9; 2 і 6. На другому кроці триває об'єднання в кластери: спостереження 11, 10, 3, 4, 5 і 7, 8, 9. Цей процес триває доти, поки всі спостереження не об'єднаються в один кластер.

Міри подібності. Для обчислення відстані між об'єктами використовуються різні міри подібності, їх називають також метриками або функціями відстаней. На початку теми розглянуто евклідову відстань, це найбільш популярна міра подібності.

Квадрат евклідової відстані. Для надання більшої ваги більш віддаленим один від одного об'єктів можемо скористатися квадратом евклідової відстані шляхом піднесення у квадрат стандартної евклідової відстані.

Манхеттенска відстань (відстань міських кварталів), також називається «хемінговою» або «сіті-блок» відстанню. Ця відстань розраховується як середня різниць по координатах. У більшості випадків ця міра відстані приводить до результатів, подібних розрахункам відстані евкліда. Однак, для цієї міри вплив окремих викидів менший, ніж при використанні евклидової відстані, оскільки тут координати не підносяться до квадрату.

Відстань Чебишева. Цю відстань варто використовувати, коли необхідно визначити два об'єкти як «різні», якщо вони відрізняються за якимось одним виміром.

Відсоток незгоди. Ця відстань обчислюється, якщо дані є категоріальними.

Методи об'єднання або зв'язки. Коли кожний об'єкт являє собою окремий кластер, відстані між цими об'єктами визначаються обраною мірою. Виникає таке питання – як визначити відстані між кластерами? Існують різні правила – методи об'єднання або зв'язки для двох кластерів.

Метод найближчого сусіда або одиночний зв'язок. Тут відстань між двома кластерами визначається відстанню між двома найбільш близькими об'єктами (найближчими сусідами) у різних кластерах. Цей метод дозволяє виділяти кластери як завгодно складної форми за умови, що різні частини таких кластерів з'єднані ланцюжками близьких один до одного елементів. У результаті роботи цього методу кластери представляються довгими «ланцюжками» або «волокнистими» кластерами, «зчепленими разом» тільки окремими елементами, які випадково виявилися ближче інших один до одного.

Метод найбільш віддалених сусідів або повний зв'язок. Тут відстані між кластерами визначаються найбільшою відстанню між будь-якими двома об'єктами в різних кластерах (тобто «найбільш віддаленими сусідами»). Метод добре використовувати, коли об'єкти дійсно походять із різних «ділянок». Якщо ж кластери мають до певної міри подовжену форму або їх природній тип є «ланцюговим», то цей метод не слід використовувати.

Метод Варда (Ward's method). За відстань між кластерами береться приріст суми квадратів відстаней об'єктів до центрів кластерів, одержуваний у результаті їх об'єднання (Ward, 1963). На відміну від інших методів кластерного аналізу для оцінки відстаней між кластерами, тут використовуються методи дисперсійного аналізу. На кожному кроці алгоритму поєднуються такі два кластери, які приводять до мінімального збільшення цільової функції, тобто внутрішньо групової суми квадратів. Цей метод спрямований на об'єднання близько розташованих кластерів і «прагне» створювати кластери малого розміру.

Метод незваженого попарного середнього (метод незваженого попарного арифметичного середнього – unweighted pair-group method using arithmetic averages, UPGMA (Sneath, Sokal, 1973)). За відстань між двома кластерами

береться середня відстань між усіма парами об’єктів у них. Цей метод слід використовувати, якщо об’єкти дійсно походять із різних «ділянок», у випадках присутності кластерів «ланцюгового» типу, при припущенні нерівних розмірів кластерів.

Метод зваженого попарного середнього (метод зваженого попарного арифметичного середнього – weighted pair-group method using arithmetic averages, WPGMA (Sneath, Sokal, 1973)). Цей метод схожий на метод незваженого попарного середнього, різниця полягає лише в тому, що тут як ваговий коефіцієнт використовується розмір кластера (число об’єктів, що втримуються в кластері). Рекомендується використовувати саме при наявності припущення про кластери різних розмірів.

Незважений центроїдний метод (метод незваженого попарного центроїдного усереднення – unweighted pair-group method using the centroid average (Sneath and Sokal, 1973)). За відстань між двома кластерами в цьому методі береться відстань між їхніми центрами ваги.

Зважений центроїдний метод (метод зваженого попарного центроїдного усереднення – weighted pair-group method using the centroid average, WPGMC (Sneath, Sokal 1973)). Цей метод схожий на попередній, різниця полягає в тому, що для обліку різниці між розмірами кластерів (числа об’єктів у них), використовуються ваги. Використовують переважно у випадках, якщо є припущення щодо істотних відмінностей у розмірах кластерів.

6.3 Ієрархічний кластерний аналіз

Розглянемо процедуру ієрархічного кластерного аналізу, яка передбачає угруповання як об’єктів (рядків матриці даних), так і змінних (стовпців). Можна вважати, що в останньому випадку роль об’єктів відіграють змінні, а роль змінних – стовпці.

У цьому методі реалізується ієрархічний агломеративний алгоритм, зміст якого полягає в такому. Перед початком кластеризації всі об’єкти вважаються окремими кластерами, у ході алгоритму вони поєднуються. Спочатку вибирається пара найближчих кластерів, які поєднуються в один кластер. У результаті кількість кластерів стає рівним *N-1*. Процедура повторюється, поки всі класи не об’єднаються. На будь-якому етапі об’єднання можна перервати, одержавши потрібне число кластерів. Отже, результат роботи алгоритму агрегування залежить від способів обчислення відстані між об’єктами й визначення близькості між кластерами.

Для визначення відстані між парою кластерів можуть бути сформульовані різні підходи. У програмних продуктах для кластерного аналізу передбачені такі методи:

– середня відстань між кластерами (Between-groups linkage), установлюється за замовчуванням;

– середня відстань між усіма об’єктами пари кластерів з урахуванням відстаней усередині кластерів (Within-groups linkage);

– відстань між найближчими сусідами – найближчими об’єктами кластерів (Nearest neighbor);

– відстань між самими далекими сусідами (Furthest neighbor);

– відстань між центрами кластерів (Centroid clustering) або центроидный метод: недоліком цього методу є те, що центр об’єднаного кластера обчислюється як середнє центрів поєднуваних кластерів, без обліку їх обсягу;

– метод Варда;

– метод медіан – той же центроїдний метод, але центр об’єднаного кластера обчислюється як середнє всіх об’єктів (Median clustering).

Процедура стандартизації використовується для виключення ймовірності того, що класифікацію будуть визначати зміни, що мають найбільший розкид значень. У програмних продуктах застосовуються такі види стандартизації:

– Z-Шкали (Z-Scores): зі значень змінних віднімається їхнє середнє, і ці значення діляться на стандартне відхилення;

– розкид від -1 до 1: лінійним перетворенням змінних домагаються розкиду значень від -1 до 1;

– розкид від 0 до 1: лінійним перетворенням змінних домагаються розкиду значень від 0 до 1;

–максимум 1: значення змінних діляться на їхній максимум;

– середнє 1: значення змінних діляться на їхнє середнє;

– стандартне відхилення 1: значення змінних діляться на стандартне відхилення.

Крім того, можливі перетворення самих відстаней, зокрема, можна відстані замінити їхніми абсолютними значеннями, це актуально для коефіцієнтів кореляції. Можна також усі відстані перетворити так, щоб вони змінювалися від 0 до 1.

Існує проблема визначення числа кластерів. Іноді можна апріорно визначити це число. Однак у більшості випадків число кластерів визначається в процесі агломерації/поділу множини об’єктів.

Процесу угруповання об’єктів в ієрархічному кластерному аналізі відповідає поступове зростання коефіцієнта, який називається критерієм Е. Стрибкоподібне збільшення значення критерію Е можна визначити як характеристику числа кластерів, які дійсно існують у досліджуваному наборі даних. Отже, цей спосіб зводиться до визначення стрибкоподібного збільшення деякого коефіцієнта, який характеризує перехід від сильно зв’язаного до слабко зв’язаного стану об’єктів.

6.4 Алгоритми неієрархічної кластеризації

При великій кількості спостережень ієрархічні методи кластерного аналізу непридатні. У таких випадках використовують **неієрархічні методи**, засновані на поділі, які являють собою **ітеративні методи дроблення вихідної сукупності**. У процесі розподілу нові кластери формуються доти, поки не буде виконане **правило зупинки**.

Така неієрархічна кластеризація полягає в поділі набору даних на певну кількість окремих кластерів. **Існує два підходи**. Перший полягає у визначенні границь кластерів як найбільш щільних ділянок у багатомірному просторі вихідних даних, тобто визначення кластера там, де є велике «згущення точок». Другий підхід полягає в мінімізації міри відмінності об'єктів.

Найпоширеніший серед неієрархічних методів є ***алгоритм k-середніх***, який також називають ***швидким кластерним аналізом***. Повний опис алгоритму можна знайти в роботі Хартігана і Вонга (Hartigan and Wong, 1978). На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо числа кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш імовірну кількість кластерів.

Алгоритм *k*-середніх будує *k* кластерів, розташованих на максимально можливо великих відстанях один від одного. Основний тип задач, які вирішує алгоритм *k*-середніх, – наявність припущень (гіпотез) щодо числа кластерів, при цьому вони повинні бути різні настільки, наскільки це можливо. Вибір числа *k* може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Загальна ідея алгоритму: задане фіксоване число *k* кластерів спостереження зіставляється кластерам так, що середні в кластері (для всіх змінних) максимально можливо відрізняються одна від одної.

Опис алгоритму:

1. Первісний розподіл об'єктів по кластерах.

Обирається число *k*, і на першому кроці ці точки вважаються «центрами» кластерів. Кожному кластеру відповідає один центр.

Вибір початкових центроїдів може здійснюватися в такий спосіб:

– вибір спостережень для максимізації початкової відстані;

– випадковий вибір *k*-спостережень;

– вибір перших *k*-спостережень.

У результаті кожний об'єкт призначений певному кластеру.

2. Ітеративний процес.

Обчислюються центри кластерів, якими потім і далі вважаються покоординатні середні кластерів. Об'єкти знову перерозподіляються.

Процес обчислення центрів і перерозподілу об'єктів триває доти, поки не виконана одна з умов:

– кластерні центри стабілізувалися, тобто всі спостереження належать кластеру, якому належали до поточної ітерації;

– число ітерацій дорівнює максимальному числу ітерацій.

На рис. 6.5 наведений приклад роботи алгоритму *k*-середніх для *k* дорівнює двом.

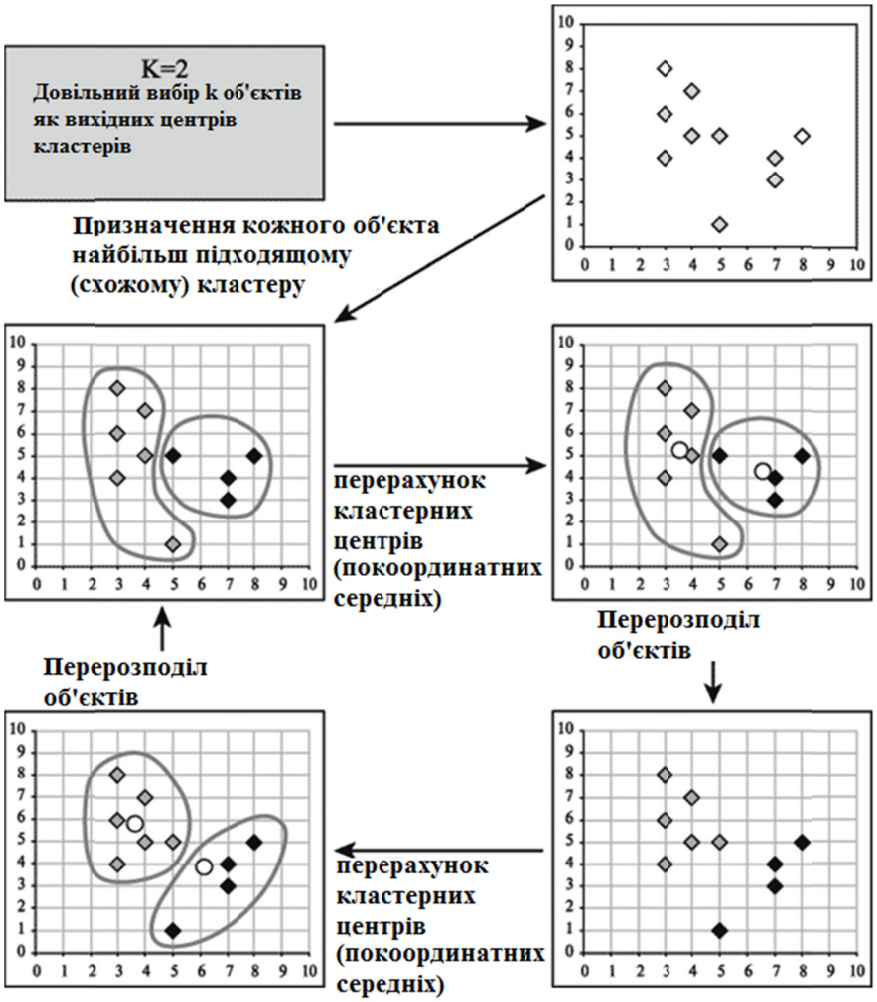

Рисунок 6.5 – Приклад роботи алгоритму *k*-середніх (*k*=2)

Вибір числа кластерів є складним питанням. Якщо немає припущень щодо цього числа, рекомендують створити 2 кластера, потім 3, 4, 5 і т.д., порівнюючи отримані результати.

Перевірка якості кластеризації. Після одержання результатів кластерного аналізу методом *k*-середніх слід перевірити правильність кластеризації (тобто оцінити, наскільки кластери відрізняються один від одного). Для цього розраховуються середні значення для кожного кластера. При гарній кластеризації повинні бути отримані дуже відмінні середні для всіх вимірів або хоча б більшої їхньої частини.

Переваги алгоритму k-середніх:

– простота використання;

– швидкість використання;

– зрозумілість і прозорість алгоритму.

Недоліки алгоритму k-середніх:

– алгоритм занадто чутливий до викидів, які можуть спотворювати середнє, можливим вирішенням цієї проблеми є використання модифікації алгоритму – алгоритм *k*-медіани;

– алгоритм може повільно працювати на великих базах даних, можливим вирішенням даної проблеми є використання вибірки даних.

Алгоритм PAM (partitioning around Medoids) є модифікацією алгоритму *k*-середніх алгоритмом *k*-медіани (k-medoids). Алгоритм менш чутливий до шумів і викидів даних, ніж алгоритм *k*-means, оскільки медіана менше піддається впливам викидів. PAM ефективний для невеликих баз даних, але його не слід використовувати для великих наборів даних.

Розглянемо приклад. Є база даних клієнтів фірми, яких слід розбити на однорідні групи. Кожний клієнт описується за допомогою 25 змінних.

Використання такого великого числа змінних призводить до виділення кластерів нечіткої структури. У результаті аналітикові досить складно інтерпретувати отримані кластери.

Більш зрозумілі й прозорі результати кластеризації можуть бути отримані, якщо замість множини вихідних змінних використовувати якісь узагальнені змінні або критерії, що містять у стислому вигляді інформацію про зв’язки між змінними. Тобто виникає задача зниження розмірності даних. Вона може вирішуватися за допомогою різних методів; один із найпоширеніших – факторний аналіз.

6.5 Порівняльний аналіз ієрархічних і неієрархічних методів кластеризації

Перед проведенням кластеризації в аналітика може виникнути питання, якій групі методів кластерного аналізу віддати перевагу. Вибираючи між ієрархічними й неієрархічними методами, необхідно враховувати такі їхні особливості.

Неієрархічні методи виявляють більш високу стабільність стосовно шумів і викидів, некоректного вибору метрики, включення незначущих змінних у набір, що брав участь у кластеризації. Ціною, яку доводиться платити за ці переваги методу, є слово «апріорі». Аналітик повинен заздалегідь визначити кількість кластерів, кількість ітерацій або правило зупинки, а також деякі інші параметри кластеризації. Це особливо складно починаючим фахівцям.

Якщо немає припущень щодо числа кластерів, рекомендують використовувати ієрархічні алгоритми. Однак, якщо обсяг вибірки не дозволяє це зробити, можливий шлях – проведення низки експериментів із різною кількістю кластерів, наприклад, почати розбивку сукупності даних із двох груп і, поступово збільшуючи їх кількість, порівнювати результати. За рахунок такого «варіювання» результатів досягається значно більша гнучкість кластеризації.

Ієрархічні методи, на відміну від неієрархічних, відмовляються від визначення кількості кластерів, а будують повне дерево вкладених кластерів.

Складності ієрархічних методів кластеризації: обмеження обсягу набору даних; вибір міри близькості; негнучкість отриманих класифікацій.

Перевага цієї групи методів у порівнянні з неієрархічними методами – їх наочність і можливість одержати детальне представлення про структуру даних.

При використанні ієрархічних методів існує можливість досить легко ідентифікувати викиди в наборі даних і, як результат, підвищити якість даних. Ця процедура лежить в основі двокрокового алгоритму кластеризації. Такий набір даних надалі може бути використаний для проведення неієрархічної кластеризації.

Існує ще один аспект – питання кластеризації всієї сукупності даних або ж її вибірки. Названий аспект вагомий для обох розглянутих груп методів, однак він більш критичний для ієрархічних методів. Ієрархічні методи не можуть працювати з більшими наборами даних, а використання деякої вибірки, тобто частини даних, могло б дозволити застосовувати ці методи.

Результати кластеризації можуть не мати достатнього статистичного обґрунтування. З іншого боку, при розв'язку задач кластеризації припустима нестатистична інтерпретація отриманих результатів, а також досить велика різноманітність варіантів поняття кластера. Така нестатистична інтерпретація дає можливість аналітикові одержати задовольняючі його результати кластеризації, що при використанні інших методів часто буває скрутним.

До останнього часу основним критерієм, по якому оцінювався алгоритм кластеризації, була якість кластеризації: вважалося, щоб увесь набір даних вміщався в оперативній пам'яті. Однак зараз, у зв'язку з появою надвеликих баз даних, з'явилися нові вимоги, яким повинен задовольняти алгоритм кластеризації. Основна з них – це масштабованість алгоритму.

Відзначимо також інші властивості, яким повинен задовольняти алгоритм кластеризації: незалежність результатів від порядку вхідних даних; незалежність параметрів алгоритму від вхідних даних.

Останнім часом ведуться активні розробки нових алгоритмів кластеризації, здатних обробляти надвеликі бази даних. У них основна увага приділяється масштабованості. До таких алгоритмів відноситься узагальнене представлення кластерів (summarized cluster representation), а також вибірка й використання структур даних, підтримуваних СУБД.

Розроблені алгоритми, у яких методи ієрархічної кластеризації інтегровані з іншими методами. До таких алгоритмів відносяться: BIRCH, CURE, CHAMELEON, ROCK.

Алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) запропонований Тьян Зангом і його колегами. Завдяки узагальненим представленням кластерів, швидкість кластеризації збільшується, алгоритм при цьому має більше масштабування.

У цьому алгоритмі реалізований двоетапний процес кластеризації. У ході першого етапу формується попередній набір кластерів. На другому етапі до виявлених кластерів застосовуються інші алгоритми кластеризації – придатні для роботи в оперативній пам'яті.

Розглянемо аналогію, що описує цей алгоритм. Якщо кожний елемент даних уявити собі як бусинку, що лежить на поверхні стола, то кластери бусин можна «замінити» тенісними кульками й перейти до більш детального вивчення кластерів тенісних кульок. Кількість бусин може виявитися досить

велике, однак діаметр тенісних кульок можна підібрати таким чином, щоб на другому етапі можна було, застосувавши традиційні алгоритми кластеризації, визначити дійсну складну форму кластерів.

Алгоритм Wavecluster являє собою алгоритм кластеризації на основі хвильових перетворень. На початку роботи алгоритму дані узагальнюються шляхом накладення на простір даних багатомірних ґрат. На подальших кроках алгоритму аналізуються не окремі точки, а узагальнені характеристики точок, що потрапили в одне гніздо ґрат. У результаті такого узагальнення необхідна інформація розміщується в оперативній пам'яті. На наступних кроках для визначення кластерів алгоритм застосовує хвильове перетворення до узагальнених даних.

Головні особливості Wavecluster:

1) складність реалізації;
2) алгоритм може виявляти кластери довільних форм;
3) алгоритм не чутливий до шумів;
4) алгоритм застосовується тільки до даних низкої розмірності.

Алгоритм CLARA **(Clustering Large Applications)** був розроблений Kaufmann і Rousseeuw у 1990 році для кластеризації даних у великих базах даних. Даний алгоритм будується в статистичних аналітичних пакетах, наприклад, таких як S+.

Алгоритм CLARA витягає множину зразків із бази даних. Кластеризація застосовується до кожного із зразків, на виході алгоритму пропонується краща кластеризація.

Для великих баз даних цей алгоритм ефективніший, ніж алгоритм PAM. Ефективність алгоритму залежить від обраного за зразок набору даних. Гарна кластеризація на обраному наборі може не дати гарну кластеризацію на всій множині даних.

Алгоритми Clarans, CURE, Dbscan формулюють задачу кластеризації як випадковий пошук у графові. У результаті роботи цих алгоритмів сукупність вузлів графа являє собою розбивку множини даних на число кластерів, визначене користувачем. «Якість» отриманих кластерів визначається за допомогою критеріальної функції. Алгоритм Clarans сортує всі можливі розбивки множини даних у пошуках прийнятного розв'язку. Пошук розв'язку зупиняється в тому вузлі, де досягається мінімум серед визначеного числа локальних мінімумів.

Серед нових масштабованих алгоритмів також можна відзначити алгоритм CURE – алгоритм ієрархічної кластеризації, і алгоритм Dbscan, де поняття кластера формулюється з використанням концепції щільності (density).

Основним недоліком алгоритмів BIRCH, Clarans, CURE, Dbscan є та обставина, що вони вимагають задання деяких порогів щільності точок, а це не завжди прийнятне. Ці обмеження зумовлені тим, що описані алгоритми орієнтовані на надвеликі бази даних і не можуть користуватися великими обчислювальними ресурсами.

Над масштабованими методами зараз активно працюють багато дослідників, основне завдання яких – подолати недоліки алгоритмів, що існують на сьогодні.

Питання для самоконтролю

1. Які існують групи задач кластерного аналізу?
2. Як вимірюється відстань між двома точками? Які функції відстані ви знаєте?
3. Дайте визначення математичним характеристикам кластеру: центр, радіус, середньоквадратичне відхилення, розмір кластера?
4. Які найпоширеніші способи нормування (normalization) змінних?
5. Які існують групи кластерного аналізу? Чим вони відрізняються?
6. Як відбувається визначення кількості кластерів?
7. За яких умов використовуються неієрархічні методи кластерного аналізу?
8. Наведіть опис алгоритму *k*-середніх? Приведіть приклад при *k*=2.
9. Як відбувається перевірка якості кластеризації?
10. Які переваги алгоритму *k*-середніх?
11. Які існують недоліки алгоритму *k*-середніх?

Лабораторне заняття №4

Тема: Кластерний аналіз на мові R

Мета роботи: опанувати навичкі проведення кластерного аналізу.

Завдання. Проведіть кластерний аналіз на мові R для даних фінансової стійкості банків за 4 квартал 2018 р з файлу banks_2018_4.csv.

Хід роботи.

Файли banks_2018_3.csv та banks_2018_4.csv містять інформацію про показники фінансової стійкості банків, що працюють у м. Запоріжжя, за 3 та 4 квартали 2018р. відповідно.

Наведені показники фінансової стійкості відносяться до групи показників фінансової стійкості банку, яка заснована на достатності банківського капіталу.

Коефіцієнт надійності ($К_{над}$) є відношенням власного капіталу до зобов'язань і розраховується за формулою:

$$К_{над} = \frac{Капітал}{Зобов'язання}. \quad (6.3)$$

Якщо в чисельнику балансовий капітал (капітал-брутто), то передбачається, що капітал повинен на 25-30% покривати зобов'язання, а якщо чисельник містить чистий капітал (регулятивний), то значення цього коефіцієнта має бути більше 5% (а для деяких банків 10%). Динаміка показника свідчить про фінансовий стан банку: при зростанні показника стійкість підвищується, а при падінні – знижується. Аритмія показника свідчить про

ризик втрат по формуванню стійкої ресурсної бази або є свідченням можливих проблем з поточною ліквідністю.

Коефіцієнт фінансового важеля ($К_{фв}$) є оберненим показником до коефіцієнта надійності, він розкриває здатність банку залучати кошти на фінансовому ринку:

$$К_{фв} = \frac{Зобов'язання}{Капітал}. \quad (6.4)$$

Збільшення цього показника свідчить про підвищення ділової активності банку, але його фінансова активність знижується. Значення цього коефіцієнта повинно бути близько 20:1.

Загальний рівень фінансування активів за рахунок власного капіталу, тобто скільки грошей припадає на 1 грн. активів показує коефіцієнт участі власного капіталу у формуванні активів ($К_{ук}$):

$$К_{ук} = \frac{Капітал}{Загальні\ активи}. \quad (6.5)$$

Значення цього коефіцієнта має бути не менше 4%.

1. Підключіть необхідні для роботи бібліотеки (dplyr, factoextra).

2. Завантажте з Moodle до R дані з інформацією про показники фінансової стійкості банків з файлів banks_2018_3.csv та banks_2018_4.csv.

Наприклад, завантажимо дані за 3 квартал 2018р. до змінної banks_3:

```
banks_3<-read.csv("banks_3_2018.csv", header=T, sep=",")
```

Подивіться на дані за допомогою команд Veiw() та head().

3. Для проведення ієрархічного кластерного аналізу оберемо змінні, на основі яких будемо кластеризувати банки. Це 2-4 стовпці, які містять інформацію про показники фінансової стійкості.

```
library(dplyr)
to_clust<-banks_3 %>% select(2,3,4)
```

Логіка використання оператора %>% бібліотеки dplyr така: взяти те, що зліва, та подати цей об’єкт на вхід функції, яка стоїть справа від оператора. В нашому випадку ми беремо базу banks_3 та подаємо її на вхід функції select для обирання стовпців. Всередині дужок у select перелічуємо через кому необхідні для аналізу стовпці. Обрані стовпці зберігаються до нової маленької бази to_clust.

Назвемо рядки за назвою банків для зручності розшифрування дендрограми^

rownames(to_clust)<-banks_3$Банк

Стандартизуємо дані (scale) та створимо матрицю відстаней m:

m <- dist(scale(to_clust))

Проведемо кластерний аналіз, використовуючи метод Уорда як метод агрегування.

Зауваження: замість необхідної матриці з квадратами евклідової відстані візьмемо матрицю зі звичайною евклідовою відстанню, щоби можна було використати цю ж матрицю для іншого методу агрегування:

hc <- hclust(m, method = "ward.D")

Примітка: зверніть увагу, що код hclust(m^2, method = "ward.D")та код hclust(m, method = "ward.D2") повинні дати однакові результати.

Подивимось на дендрограму (рис.6.6):

plot(hc, cex = 0.9)

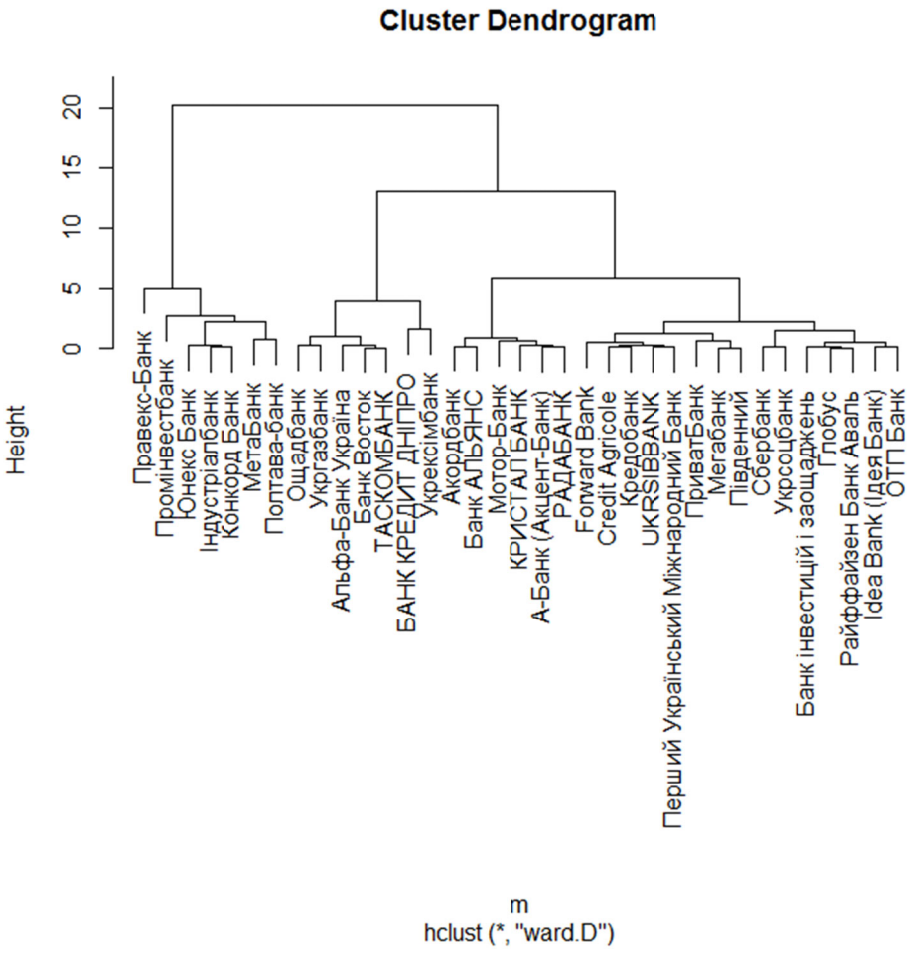

Рисунок 6.6 – Дендрограма банків м. Запоріжжя

Виділемо 3 кластери (рис.6.7).
plot(hc, cex = 0.9)
rect.hclust(hc, k = 3)

Витягнемо з отриманої кластеризації мітки кластерів. Додамо їх окремим стовпцем до бази даних, попередньо зробивши ці мітки факторними (тобто не числами, а умовними позначками)

groups3 <- cutree(hc, k = 3)
banks_3$groups3 <- factor(groups3)

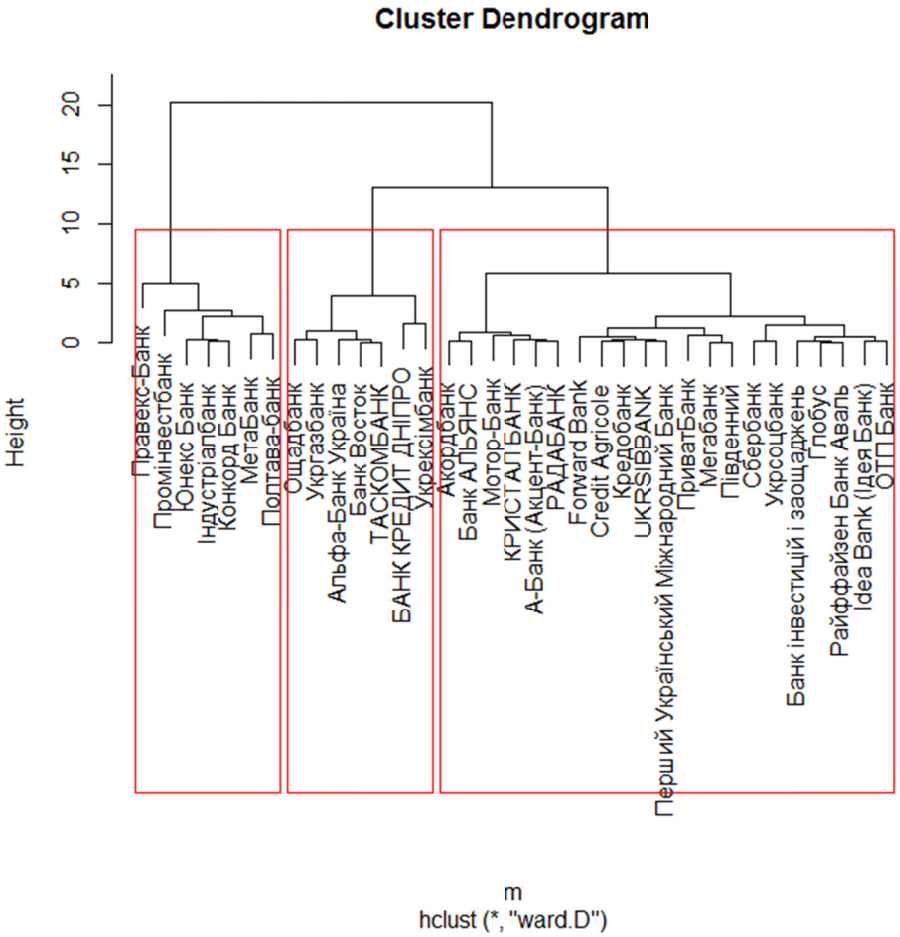

Рисунок 6.7 –Виділення кластерів

4. Розглянемо кожен кластер окремо. Будемо обирати з бази рядки, де значення groups5 дорівнює 1, 2 або 3, а потім оцінювати змістовно отримані кластери. Скористаємося функцією filter() бібліотеки dplyr:

banks_3 %>% filter(groups3 == 1) %>% View
banks_3 %>% filter(groups3 == 2) %>% View
banks_3 %>% filter(groups3 == 3) %>% View

5. Реалізуємо метод кластеризації k-середніх також для k=3 для стандартизованих даних:

banks_3_clusters<-kmeans(m, 3)

Результат кластеризації міститься у об'єкті banks_3_clusters. Перегляньте його.

6. Перевірте оптимальну кількість кластерів. Для цього встановіть та підключіть бібліотеку factoextra.

Elbow method ("метод зігнутого коліна"). Побудуємо графік, де за віссю абсцис відмічено кількість кластерів k, а за віссю ординат – значення функції W(k), яка визначає внутрішньогруповий розкид в залежності від кількості кластерів (рис.6.8)

```
fviz_nbclust(to_clust, kmeans, method = "wss") +
labs(subtitle = "Elbow method") +
geom_vline(xintercept = 4, linetype = 2)
```

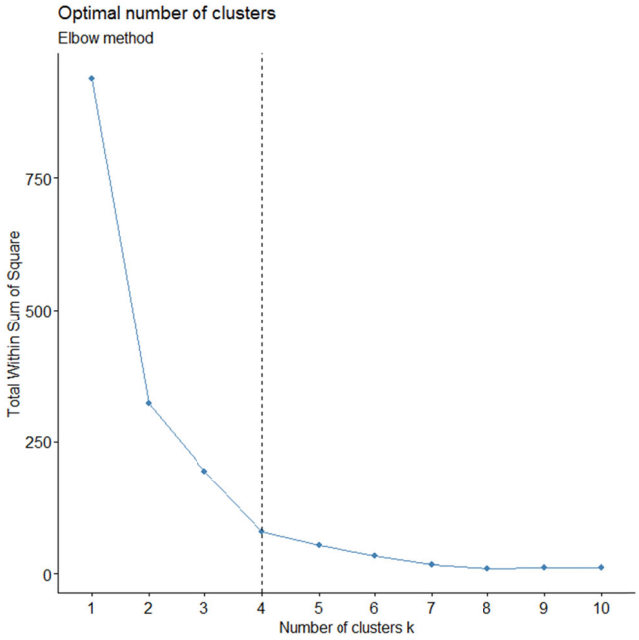

Рисунок 6.8 – Внутрішньогруповий розкид в залежності від кількості кластерів

Спробуємо інший метод - Silhouette method ("силуетний метод"):

```
fviz_nbclust(to_clust, kmeans, method = "silhouette") +
labs(subtitle = "Silhouette method")
```

Таким чином, кількість кластерів має бути від 2 до 4.

7. Проведіть аналогічний аналіз для даних фінансової стійкості банків за 4 квартал 2018 р.

Тема 7. Метод дерева рішень

Мета: ознайомитися із застосуванням методу дерева рішень для завдань класифікації та прогнозування.

План

7.1 Метод дерев рішень
7.2 Переваги дерев рішень
7.3 Алгоритми

Основні поняття

Дерево рішень. Критерій розщеплення. Зупинка побудови дерева. Скорочення дерева або відсікання гілок.

7.1 Метод дерев рішень

Метод дерев рішень (decision trees) є одним із найбільш популярних методів розв'язання задач класифікації й прогнозування. Іноді цей метод Data Mining також називають деревами вирішальних правил, деревами класифікації і регресії.

Як видно з останньої назви, за допомогою даного методу розв'язуються задачі класифікації й прогнозування.

Якщо залежна, тобто цільова змінна приймає дискретні значення, за допомогою методу дерева рішень розв'язується задача класифікації.

Якщо ж залежна змінна приймає безперервні значення, то дерево рішень установлює залежність цієї змінної від незалежних змінних, тобто розв'язує задачу чисельного прогнозування.

У найбільш простому вигляді дерево рішень – це спосіб показу правил в ієрархічній, послідовній структурі. Основа такої структури – відповіді «Так» або «Ні» на низку питань.

Наведемо приклад дерева рішень, задача якого – відповістити на запитання: «Чи грати в гольф?» Щоб розв'язати задачу, тобто прийняти рішення, чи грати в гольф, слід віднести поточну ситуацію до одного з відомих класів (у цьому випадку – «грати» або «не грати»). Для цього потрібно відповісти на низку питань, які є у вузлах цього дерева, починаючи з його кореня.

Перший вузол нашого дерева «Сонячно?» є вузлом перевірки, тобто умовою. При позитивній відповіді на запитання здійснюється перехід до лівої частини дерева, що називається лівою гілкою, при негативному – до правої частини дерева. Отже, внутрішній вузол дерева є вузлом перевірки певної умови. Далі йде наступне питання і т.д., поки не буде досягнутий кінцевий вузол дерева, що є вузлом розв'язку. Для нашого дерева існує два типи кінцевого вузла: «грати» і «не грати» у гольф.

У результаті проходження від кореня дерева (іноді називається кореневою вершиною) до його вершини розв'язується задача класифікації, тобто вибирається один із класів – «грати» чи «не грати» у гольф**.**

Метою побудови дерева рішень в нашому випадку є визначення значення категоріальної залежної змінної.

Отже, для нашої задачі основними елементами дерева рішень є:

– корінь дерева: «Сонячно?»

–внутрішній вузол дерева або вузол перевірки: «Температура повітря висока?», «Чи йде дощ?»

– листок, кінцевий вузол дерева, вузол розв'язку або вершина: «Грати», «Не грати»;

– гілки дерева (варіанти відповіді): «Так», «Ні».

У розглянутому прикладі розв'язується задача бінарної класифікації, тобто створюється дихотомічна класифікаційна модель. Приклад демонструє роботу так званих бінарних дерев.

У вузлах бінарних дерев розгалуження може відбуватися тільки у двох напрямках, тобто існує можливість тільки двох відповідей на поставлене питання («так» і «ні»).

Бінарні дерева є найпростішим, частковим випадком дерев рішень. В інших випадках, відповідей і, відповідно, гілок дерева, що виходять із його внутрішнього вузла, може бути більше двох.

Розглянемо більш складний приклад.

База даних, на основі якої повинне здійснюватися прогнозування, містить такі ретроспективні дані про клієнтів банку, що є її атрибутами:

– вік,

– наявність нерухомості,

– освіта,

– середньомісячний дохід,

– чи повернув клієнт вчасно кредит.

Задача полягає в тому, щоб на підставі перерахованих вище даних (крім останнього атрибута) визначити, чи варто видавати кредит новому клієнтові.

Така задача розв'язується у два етапи:

– побудова класифікаційної моделі

– її використання.

На етапі побудови моделі, власне, і будується дерево класифікації або створюється набір якихось правил.

На етапі використання моделі побудоване дерево, або шлях від його кореня до однієї з вершин, що є набором правил для конкретного клієнта, використовується для відповіді на поставлене питання «Чи видавати кредит?».

Правилом є логічна конструкція, представлена у вигляді «якщо : то :».

Наведемо приклад дерева класифікації, за допомогою якого розв'язується задача «Чи видавати кредит клієнтові?». Вона є типовою задачею класифікації, і за допомогою дерев рішень одержують досить хороші варіанти її розв'язку.

Як ми бачимо, внутрішні вузли дерева (вік, наявність нерухомості, дохід і освіта) є атрибутами описаної вище бази даних.

Ці атрибути називають прогнозуючими, або атрибутами розщеплення (splitting attribute). Кінцеві вузли дерева, або листки, іменуються мітками класу,

що є значеннями залежної категоріальної змінної «видавати» або «не видавати» кредит.

Кожна гілка дерева, що йде від внутрішнього вузла, відзначена предикатом розщеплення. Останній може відноситися лише до одного атрибуту розщеплення даного вузла.

Характерна риса предикатів розщеплення: кожний запис використовує унікальний шлях від кореня дерева тільки до одного вузла-розв'язку. Об'єднана інформація про атрибути розщеплення й предикати розщеплення у вузлі називається критерієм розщеплення (splitting criterion).

Наприклад, критерій розщеплення «Яка освіта?», міг би мати два предикати розщеплення й виглядати інакше: освіта «вища» і «не вища». Тоді дерево рішень мало б інший вигляд.

Отже, для цієї задачі (як і для будь-якої іншої) може бути побудовано множина дерев рішень різної якості, з різною прогнозуючою точністю.

Якість побудованого дерева рішень досить сильно залежить від правильного вибору критерію розщеплення. Над розробкою й удосконаленням критеріїв працюють багато дослідників.

Метод дерев рішень часто називають «наївним» підходом. Але завдяки певній низці переваг, цей метод є одним із найбільш популярних для розв'язання задач класифікації.

7.2 Переваги дерев рішень

Класифікаційна модель, представлена у вигляді дерева рішень, є інтуїтивною і спрощує розуміння розв'язуваної задачі.

Результат роботи алгоритмів конструювання дерев рішень, *на відміну, наприклад, від нейронних мереж, що представляють собою «чорні ящики»,* легко інтерпретується користувачем. Ця властивість дерев рішень не тільки важлива при віднесенні до певного класу нового об'єкта, але й корисна при інтерпретації моделі класифікації в цілому. Дерево рішень дозволяє зрозуміти й пояснити, чому конкретний об'єкт відноситься до того або іншого класу.

Дерева рішень дають можливість витягати правила з бази даних звичайною мовою. Приклад правила: «Якщо Вік >35 і Дохід >200, то видати кредит».

Дерева рішень дозволяють створювати класифікаційні моделі в тих сферах, де аналітикові досить складно формалізувати знання.

Алгоритм конструювання дерева рішень не вимагає від користувача вибору вхідних атрибутів (незалежних змінних). На вхід алгоритму можна подавати всі існуючі атрибути, алгоритм сам вибере найбільш значимі серед них, і тільки вони будуть використані для побудови дерева. У порівнянні, наприклад, з нейронними мережами, це значно полегшує користувачеві роботу, оскільки в нейронних мережах вибір кількості вхідних атрибутів суттєво впливає на час навчання.

Точність моделей, створених за допомогою дерев рішень, вища порівняно з іншими методами побудови класифікаційних моделей (статистичні методи, нейронні мережі).

Розроблений ряд масштабованих алгоритмів, які можуть бути використані для побудови дерев рішень на надвеликих базах даних. Масштабованість тут означає, що із зростанням кількості прикладів або записів бази даних час, затрачуваний на навчання, тобто побудову дерев рішень, зростає лінійно. Приклади таких алгоритмів: SLIQ, SPRINT.

На побудову класифікаційних моделей за допомогою алгоритмів конструювання дерев рішень потрібно значно менше часу, ніж, наприклад, на навчання нейронних мереж.

Більшість алгоритмів конструювання дерев рішень мають можливість спеціальної обробки пропущених значень.

Багато класичних статистичних методів, за допомогою яких розв'язуються задачі класифікації, можуть працювати тільки із числовими даними, у той час як дерева рішень працюють і з числовими, і з категоріальними типами даних.

Багато статистичних методів є параметричними, і користувач повинен заздалегідь володіти певною інформацією, наприклад, знати вид моделі, мати гіпотезу про вид залежності між змінними, припускати, який вид розподілу мають дані. Дерева рішень, на відміну від таких методів, будують непараметричні моделі. Отже, дерева рішень здатні розв'язувати такі задачі Data Mining, у яких відсутня апріорна інформація про вид залежності між досліджуваними даними.

Процес конструювання дерева рішень. Задача класифікації, що розглядається, відноситься до стратегії навчання з учителем, яке іноді називається індуктивним навчанням. У цих випадках усі об'єкти тренувального набору даних заздалегідь віднесені до одного з визначених класів.

Алгоритми конструювання дерев рішень складається з етапів «побудова» або «створення» дерева (tree building) і «скорочення» дерева (tree pruning). У ході створення дерева вирішуються питання вибору критерію розщеплення й зупинки навчання (якщо це передбачено алгоритмом). У ході етапу скорочення дерева вирішується питання відсікання деяких його гілок.

Процес створення дерева відбувається зверху вниз, тобто є спадним. У ході процесу алгоритм повинен знайти такий ***критерій розщеплення***, іноді також називається критерієм розбивки, щоб розбити множину на підмножини, які б асоціювалися з даним вузлом перевірки. Кожний вузол перевірки повинен бути позначений певним атрибутом. Існує правило вибору атрибута: він повинен розбивати вихідну множину даних таким чином, щоб об'єкти підмножин, що одержуються в результаті цієї розбивки, були представниками одного класу або ж були максимально наближені до такої розбивки. Остання фраза означає, що кількість об'єктів з інших класів, так званих «домішок», у кожному класі прагнула до мінімуму.

Існують різні критерії розщеплення. Найбільш відомі – міра ентропії й індекс Gini.

У деяких методах для вибору атрибута розщеплення використовується так звана міра інформативності підпросторів атрибутів, яка ґрунтується на ентропійному підході й відома за назвою «міра інформаційного виграшу» (information gain measure) або міра ентропії.

Інший критерій розщеплення, запропонований Брейманом (Breiman) та ін., реалізований в алгоритмі CART і називається індексом Gini. За допомогою цього індексу атрибут вибирається на підставі відстаней між розподілами класів.

Якщо дана множина *T*, що включає приклади з *n* класів, індекс Gini, тобто *gini(T)*, визначається за формулою:

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2, \tag{9.1}$$

де *T* – поточний вузол, *pj* – імовірність класу *j* у вузлі *T*, *n* – кількість класів.

Чим більше окремих випадків описано в дереві рішень, тим менша кількість об’єктів потрапляє в кожний окремий випадок. Такі дерева називають «гіллястими» або «кущистими», вони складаються з невиправдано великої кількості вузлів і гілок, вихідна множина розбивається на велику кількість підмножин, що складаються із дуже малого числа об’єктів. У результаті «переповнення» таких дерев їх здатність до узагальнення зменшується, і побудовані моделі не можуть давати вірні відповіді.

У процесі побудови дерева, щоб його розміри не стали надмірно великими, використовують спеціальні процедури, які дозволяють створювати оптимальні дерева, так звані дерева «вдалих розмірів» (Breiman, 1984).

Який розмір дерева може вважатися оптимальним? Дерево повинно бути досить складним, щоб ураховувати інформацію з досліджуваного набору даних, але одночасно воно повинно бути досить простим. Інакше кажучи, дерево повинно використовувати інформацію, що поліпшує якість моделі, і ігнорувати ту інформацію, яка її не поліпшує.

Отут існує дві можливі стратегії. Перша полягає в нарощуванні дерева до певного розміру відповідно до параметрів, заданих користувачем. Визначення цих параметрів може ґрунтуватися на досвіді й інтуїції аналітика, а також на деяких «діагностичних повідомленнях» системи, що конструюють дерево рішень.

Друга стратегія полягає у використанні набору процедур, що визначають «вдалий розмір» дерева, вони розроблені Бріманом, Куїлендом та ін. в 1984 році. Однак, як відзначають автори, не можна сказати, що ці процедури доступні починаючому користувачеві.

Процедури, які використовують для запобігання створення надмірно великих дерев, включають: скорочення дерева шляхом відсікання гілок; використання правил зупинки навчання.

Слід зазначити, що не всі алгоритми при конструюванні дерева працюють за однією схемою. Деякі алгоритми включають два окремі послідовні етапи: побудова дерева і його скорочення; інші чергують ці етапи в процесі своєї роботи для запобігання нарощування внутрішніх вузлів.

Розглянемо ***правило зупинки***. Воно повинне визначити, чи є розглянутий вузол внутрішнім вузлом (при цьому він буде розбиватися далі) або ж він є кінцевим вузлом, тобто вузлом розв'язком.

Зупинка – такий момент у процесі побудови дерева, коли слід припинити подальші розгалуження.

Один із варіантів правил зупинки – «рання зупинка» (prepruning), вона визначає доцільність розбивки вузла. Перевага використання такого варіанта – зменшення часу на навчання моделі. Однак тут виникає ризик зниження точності класифікації. Тому рекомендується «замість зупинки використовувати відсікання» (Breiman, 1984).

Другий варіант зупинки навчання – обмеження глибини дерева. У цьому випадку побудова закінчується, якщо досягнута задана глибина.

Ще один варіант зупинки – задання мінімальної кількості прикладів, які будуть утримуватися в кінцевих вузлах дерева. При цьому варіанті розгалуження тривають до того моменту, поки всі кінцеві вузли дерева не будуть чистими або будуть містити не більш ніж задане число об'єктів.

Існує ще ряд правил, але слід зазначити, що жодне з них не має великої практичної цінності, а деякі можуть бути застосовні лише в окремих випадках.

Вирішенням проблеми занадто гіллястого дерева є його ***скорочення шляхом відсікання (pruning) деяких гілок***.

Якість класифікаційної моделі, побудованої за допомогою дерева рішень, характеризується двома основними ознаками: точністю розпізнавання й помилкою.

Точність розпізнавання розраховується як відношення об'єктів, правильно класифікованих у процесі навчання, до загальної кількості об'єктів набору даних, які брали участь у навчанні.

Помилка розраховується як відношення об'єктів, неправильно класифікованих у процесі навчання, до загальної кількості об'єктів набору даних, які брали участь у навчанні.

Відсікання гілок або заміну деяких гілок піддеревом слід проводити там, де ця процедура не призводить до зростання помилки. Процес проходить знизу вгору, тобто є висхідним. Це більш популярна процедура, ніж використання правил зупинки. Дерева, одержувані після відсікання деяких гілок, називають усіченими.

Якщо таке усічене дерево усе ще не є інтуїтивним і складне для розуміння, використовують витяг правил, які поєднують у набори для опису класів.

Кожний шлях від кореня дерева до його вершини або листка дає одне правило. Умовами правила є перевірки на внутрішніх вузлах дерева.

7.3 Алгоритми, що реалізують дерева рішень

Сьогодні існує велика кількість алгоритмів, що реалізують дерева рішень: CART, C4.5, CHAID, CN2, Newid, Itrule і інші. Атрибути набору даних можуть мати як дискретне, так і числове значення.

Алгоритм CART (Classification and Regression Tree), як видно з назви, розв'язує задачу класифікації й регресії. Він розроблений в 1974-1984 роках чотирма професорами статистики – Leo Breiman (Berkeley), Jerry Friedman (Stanford), Charles Stone (Berkeley) і Richard Olshen (Stanford).

Алгоритм CART призначений для побудови бінарного дерева рішень. Бінарні дерева також називають двійковими. Приклад такого дерева розглядався на початку теми.

Інші особливості алгоритму CART:

– функція оцінки якості розбивки;

– механізм відсікання дерева;

–алгоритм обробки пропущених значень;

– побудова дерев регресії.

Кожний вузол бінарного дерева при розбивці має тільки двох нащадків, що називаються дочірніми галузями. Подальший поділ гілок залежить від того, чи багато вихідних даних описує дана гілка. На кожному кроці побудови дерева правило, формоване у вузлі, ділить задану множину прикладів на дві частини. Права його частина (гілка right) – це та частина множини, у якій правило виконується; ліва (гілка left) – та, для якої правило не виконується.

Функція оцінки якості розбивки, яка використовується для вибору оптимального правила, – індекс Gini – був описаний вище. Відзначимо, що дана оціночна функція заснована на ідеї зменшення невизначеності у вузлі. Допустимо, є вузол, і він розбитий на два класи. Максимальна невизначеність у вузлі буде досягнута при розбивці його на дві підмножини по 50 прикладів, а максимальна визначеність – при розбивці на 100 і 0 прикладів.

Нагадаємо, що алгоритм CART працює із числовими й категоріальними атрибутами. У кожному вузлі розбивка може йти тільки по одному атрибуту. Якщо атрибут є числовим, то у внутрішньому вузлі формується правило виду $x_i \leq c$, Значення c у більшості випадків вибирається як середнє арифметичне двох сусідніх впорядкованих значень змінної x_i навчального набору даних. Якщо ж атрибут відноситься до категоріального типу, то у внутрішньому вузлі формується правило $x_i \ V(x_i)$, де $V(x_i)$ – деяка непорожня підмножина множин значень змінної x_i у навчальному наборі даних.

Механізмом відсікання, що має назву minimal cost-complexity tree pruning, алгоритм CART принципово відрізняється від інших алгоритмів конструювання дерев рішень. У розглянутому алгоритмі відсікання – це деякий компроміс між одержанням дерева «підходящого розміру» і одержанням найбільш точної оцінки класифікації. Метод полягає в одержанні послідовності зменшуваних дерев, але дерева розглядаються не всі, а тільки «кращі представники».

Перехресна перевірка (V-fold cross-validation) є найбільш складною й одночасно оригінальною частиною алгоритму CART. Вона являє собою шлях

вибору остаточного дерева, за умови, що набір даних має невеликий обсяг або ж записи набору даних настільки специфічні, що розділити набір на навчальну й тестову вибірку не представляється можливим.

Отже, основні характеристики алгоритму CART: бінарне розщеплення, критерій розщеплення – індекс Gini, алгоритми minimal cost-complexity tree pruning і V-fold cross-validation, принцип «виростити дерево, а потім скоротити», висока швидкість побудови, обробка пропущених значень.

Алгоритм C4.5 будує дерево рішень з необмеженою кількістю гілок у вузлах. Даний алгоритм може працювати тільки з дискретним залежним атрибутом і тому може розв'язувати тільки задачу класифікації. C4.5 вважається одним із найвідоміших і широко використовуваних алгоритмів побудови дерев класифікації.

Для роботи алгоритму C4.5 необхідне дотримання таких вимог:

– кожний запис набору даних повинен бути асоційованим з одним із визначених класів, тобто один з атрибутів набору даних повинен бути міткою класу;

– класи повинні бути дискретними, кожний приклад повинен однозначно відноситися до одного із класів;

– кількість класів повинна бути значно менше кількості записів у досліджуваному наборі даних.

Остання версія алгоритму – алгоритм C4.8 – реалізована в інструменті Weka як J4.8 (Java). Комерційна реалізація методу: C5.0, розробник Rulequest, Австралія.

Алгоритм C4.5 повільно працює на надвеликих й зашумлених наборах даних.

Обидва алгоритми, CART та C4.5, є робастними, тобто стійкими до шумів і викидів даних.

Алгоритми побудови дерев рішень відрізняються такими характеристиками:

– вид розщеплення – бінарне (binary), множинне (multi-way);

– критерії розщеплення – ентропія, gini, інші;

– можливість обробки пропущених значень;

– процедура скорочення гілок або відсікання;

– можливості витягування правил з дерев.

Жоден алгоритм побудови дерева не можна апріорі вважати найкращим або досконалим, підтвердження доцільності використання конкретного алгоритму повинно бути перевірене й підтверджене експериментом.

Найбільш серйозна вимога, яка зараз пред'являється до алгоритмів конструювання дерев рішень – це масштабованість, тобто алгоритм повинен мати масштабований метод доступу до даних.

Розроблений ряд нових масштабованих алгоритмів, серед них – алгоритм Sprint, запропонований Джоном Боярином і його колегами. Sprint, що є масштабованим варіантом розглянутого вище алгоритму CART, висуває мінімальні вимоги до об'єму оперативної пам'яті.

Питання для самоконтролю

1. Дайте визначення методу дерев рішень (decision trees)? Для чого він використовується?
2. Як використовуються бінарні дерева? Наведіть приклад.
3. Які існують переваги методу дерева рішень?
4. Які існують критерії розщеплення дерева рішень?
5. Які існують варіанти зупинки навчання дерева рішень?
6. Назвіть відомі алгоритми, що реалізують дерева рішень.

Лабораторне заняття №5

Тема: Метод дерева рішень на мові R

Мета роботи: здобути навчички використання дерев рішень.

Завдання. Побудувати моделі дерева рішень на набірі даних audit та пперевірити їх на стійкість.

Хід роботи.

Набір даних audit входить до бібліотеки *rattle.data*. набір даних аудиту – це штучно створений набір даних, що володіє певними характеристиками справжнього набору даних фінансового аудиту для моделювання продуктивних та непродуктивних аудитів фінансового звіту людини. Продуктивний аудит – це такий, що виявляє помилки та неточності у інформації, наданої клієнтом. Непродуктивний аудит – це зазвичай аудит, який виявив, що вся надана інформація у порядку.

Набір даних аудиту використовується для ілюстрації бінарної класифікації. Цільова змінна ідентифікується як TARGET_Adjusted. Набір даних не великий й складається лише з 2000 об’єктів. Його основна мета – проілюструвати моделювання у Rattle, тому підходить набір даних мінімального розміру. Сам набір отримано з загальнодоступних даних (які не мають нічого спільного з аудитами).

Структура набору даних:

ID – унікальний індикатор кожної людини;

Age – вік;

Employment – тип зайнятості;

Education – найвищий рівень освіти людини;

Marital – поточний родинний стан;

Occupation – тип заняття;

Income – заявлена сума доходу;

Gender – стать;

Deductions – загальна сума витрат, які людина відображає у своїй фінансовій звітності;

Hours – середній час роботи на тиждень, години;

IGNORE_Accounts – основна країна, у якій людина зберігає більшу частину своїх грошей (зверніть увагу, що перед ім'ям змінної стоїть IGNORE. Це визнається Rattle як роль за замовченням для цієї змінної);

RISK_Adjustment – ця змінна записує грошову суму будь-якого корегування фінансових претензій особи у результаті продуктивного аудиту, тобто, ця змінна розглядається не як вхідна змінна, а як міра ризику, пов'язаного з людиною;

TARGET_Adjusted –цільова змінна для моделювання (зазвичай для класифікаційного моделювання), це числове поле класу integer, але обмежене 0 та 1, що вказує на непродуктивний та продуктивний аудити відповідно. Продуктивні аудити – це ті, що призводять до корегування фінансової звітності клієнта.

1. Побудова моделі дерева.

Підготовка даних.

Підключіть бібліотеку *rattle.data* та подивіться структуру набору даних audit:

```
library(rattle.data)
str(audit)
Розділіть дані на тренувальну та тестову множини:
library(caTools)
set.seed(3000)
split<-sample.split(audit$TARGET_Adjusted, SplitRatio=0.7)
train<-subset(audit, split==T)
test<-subset(audit, split==F)
```

Загрузіть пакети *rpart, rpart.plot, rattle, RColorBrewer*.

Побудуйте модель.

Наприклад,

```
audittree<-rpart(TARGET_Adjusted~Age+Occupation+Income, data=train, method="class", control=rpart.control(minbucket=25))
```

Зобразіть дерево (рис.7.1):

```
prp(audittree)
```

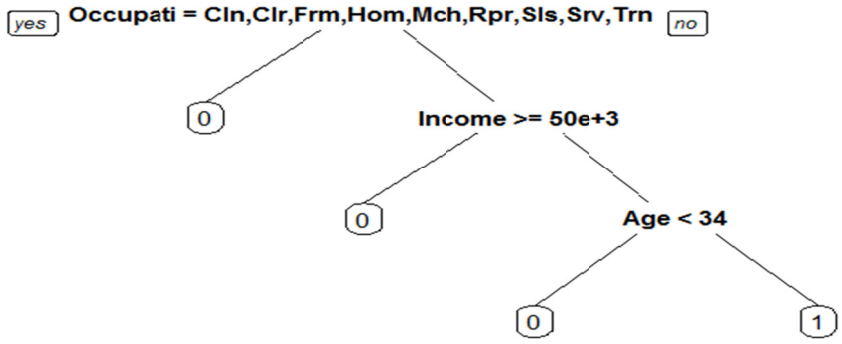

Рисунок 7.1 – Дерево рішень

Або у більш наглядному вигляді (рис.7.2):

fancyRpartPlot(audittree)

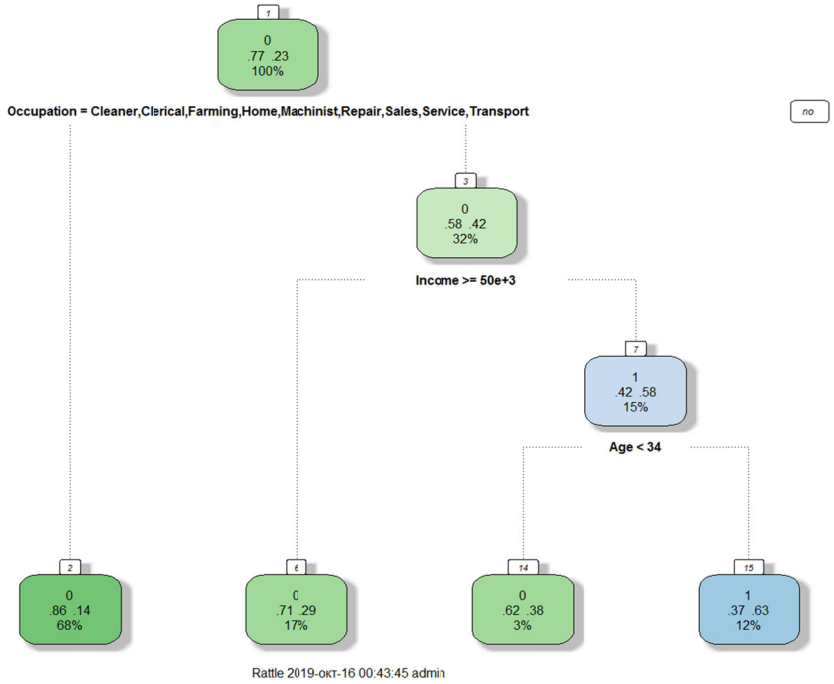

Рисунок 7.2 – Дерево рішень(наглядний вигляд)

Отримайте передбачення:

PredictCART<-predict(audittree, newdata=test, type="class")

Визначте точність:

```
conf.matr<-table(test$TARGET_Adjusted, PredictCART)
conf.matr
   PredictCART
      0   1
  0 440  21
  1  90  49
accuracy<-(conf.matr[‘0’,’0’]+conf.matr[‘1’,’1’])/sum(conf.matr[])
paste("Точність:", accuracy)
[1] "Точність: 0.815"
```

Побудова кривої похибок або ROC-кривої (рис.7.3):

```
library(ROCR)
predictROC<-predict(audittree, newdata=test)
pref<-prediction(predictROC[,2],test$TARGET_Adjusted)
```

perf<-performance(pref, "tpr", "fpr")
plot(perf)

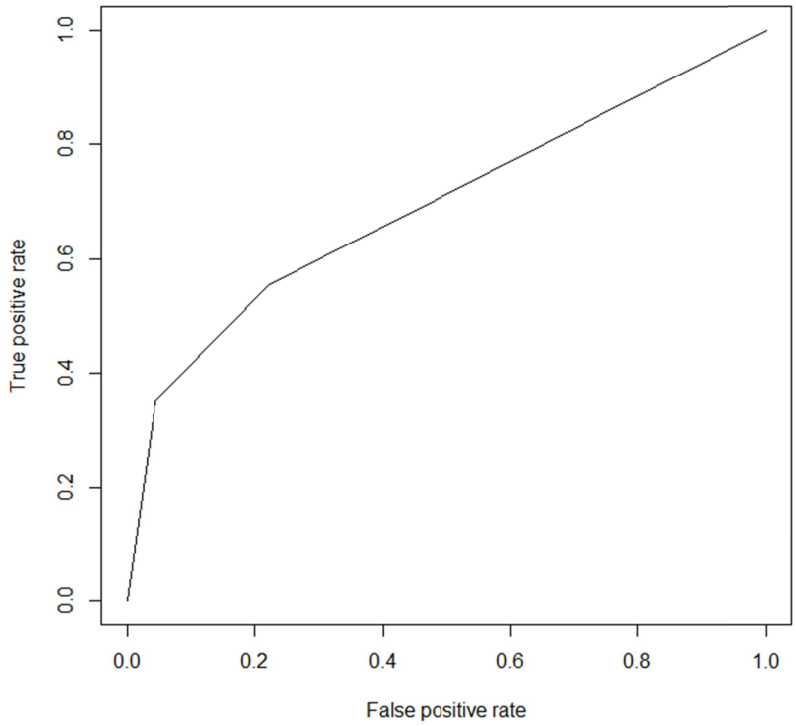

Рисунок 7.3 – Крива похибок

2. Метод крос-валідації
Встановіть необхідні пакети *caret, e1071.*
Розбийте тренувальні дані на частини (folds).
Наприклад,

fitControl<-trainControl(method="cv", number=30)

Тут, method="cv" означає, що використовувати крос-валідацію, а number=30 – що використовувати 30 частин.
Задайте значення ср.
Наприклад від 0,001 до 2

*cartGrid<-expand.grid(.cp=(1:200)*0.001)*

Увага: залежна змінна повинна мати тип factor

train(as.factor(TARGET_Adjusted) ~ Age+Occupation+Income, data = train, method = "rpart", trControl = fitControl, tuneGrid = cartGrid, na.action=na.exclude)

Результатом буде таблиця, яка містить різні значення точності для різних cp. Потрібно значення cp, яке максимізує точність, воно вказано в кінці виведення функції:

cp Accuracy Kappa
0.001 0.7397389 0.1932924176
0.002 0.7533247 0.2098812703
0.003 0.7555974 0.1980251487
0.004 0.7646065 0.2148368580
0.005 0.7721486 0.2411609837
0.006 0.7668624 0.2104484664
0.007 0.7690839 0.2226016774
0.008 0.7713566 0.2301501457
0.009 0.7758852 0.2497323309
0.010 0.7766596 0.2570468149
..0.198 0.7654972 0.0000000000
0.199 0.7654972 0.0000000000
0.200 0.7654972 0.0000000000

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.036.

Отримане cp необхідно використовувати у функції rpart як параметр control=rpart.control(cp=)

audittree_1<-rpart(TARGET_Adjusted~Age+Occupation+Income, data=train, method="class", control=rpart.control(cp=0.036))

Розгляньте отримане дерево(рис.7.4):

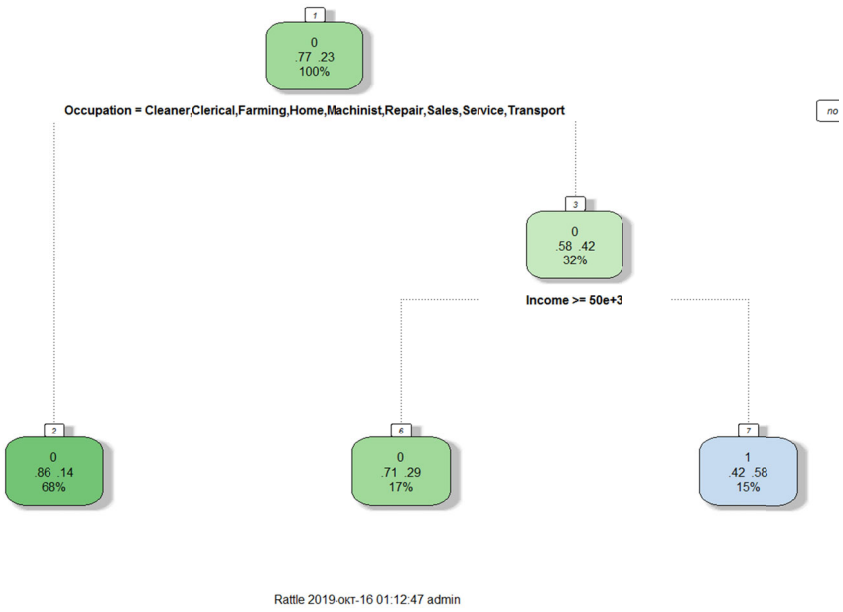

Рисунок 7.5 – Отримане дерево

Отримайте передбачення та визначте точність:

```
PredictCART<-predict(audittree_1, newdata=test, type="class")
conf.matr<-table(test$TARGET_Adjusted, PredictCART)
conf.matr
   PredictCART
      0   1
  0 425  36
  1  85  54
accuracy<-(conf.matr[‘0’,’0’]+conf.matr[‘1’,’1’])/sum(conf.matr[])
paste("Точність:", accuracy)
[1] "Точність: 0.798333333333333"
```

Побудуйте криву похибок (рис.7.6):

```
predictROC1<-predict(audittree_1, newdata=test)
pref1<-prediction(predictROC1[,2],test$TARGET_Adjusted)
perf1<-performance(pref1, "tpr", "fpr")
plot(perf1)
```

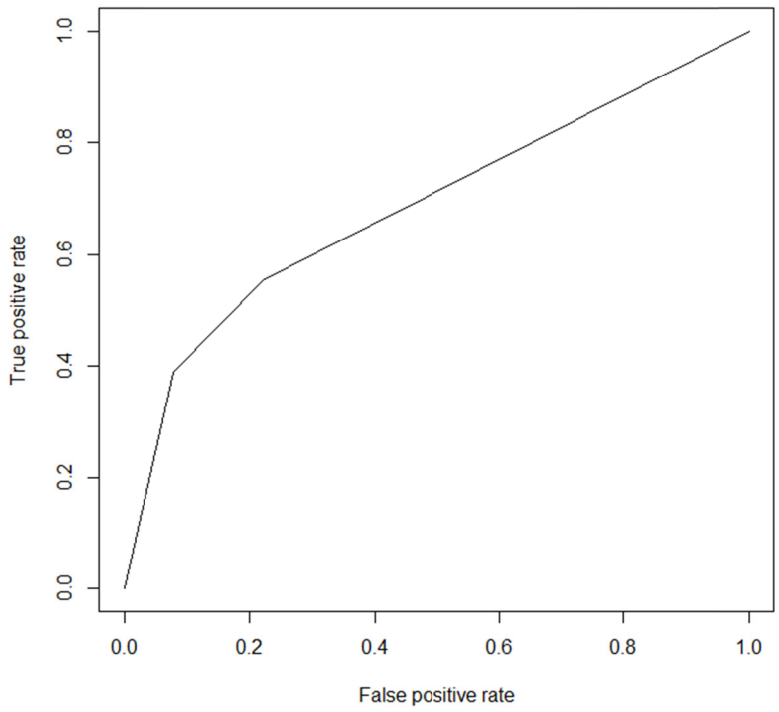

Рисунок 7.6 – Крива похибок

Порівняйте криві похибок після застосування двох методів (рис.7.7):

```
plot(perf, col="red")
par(new=T)
plot(perf1, col="green")
```

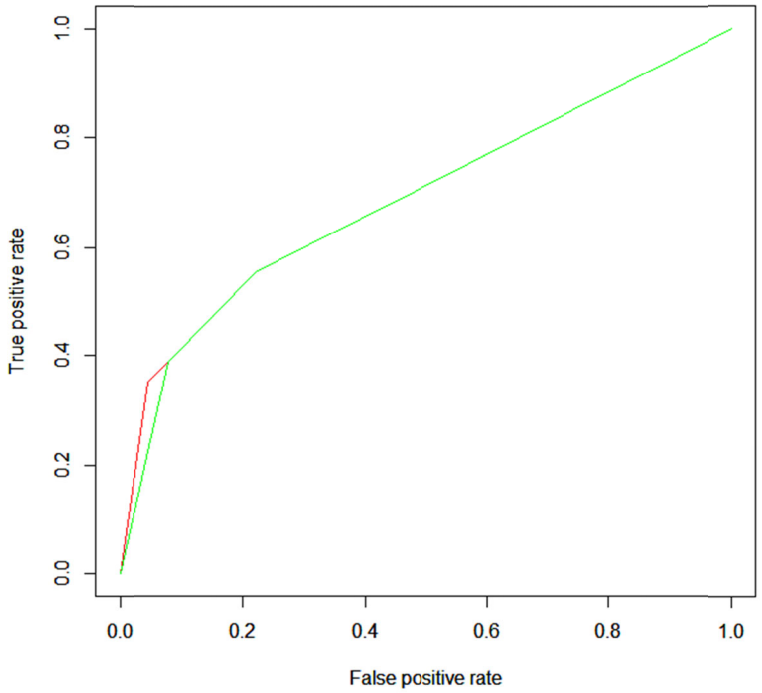

Рисунок 7.7 – Порівняння похибок

3. Алгоритм Випадковий ліс (Random forest)

Підключіть бібліотеку *randomForest*. Увага: залежна змінна повинна мати тип factor. Побудуйте випадковий ліс:

trainforest<-randomForest(as.factor(TARGET_Adjusted) ~ Age+Occupation+Income, data=train, nodesize=25, ntree=200, na.action=na.exclude)

Отримайте передбачення та точність моделі:

```
predictforest<-predict(trainforest, newdata=test)
conf.matr<-table(test$TARGET_Adjusted, predictforest)
conf.matr
predictforest
     0   1
  0 405  29
  1  75  60
accuracy<-(conf.matr['0','0']+conf.matr['1','1'])/sum(conf.matr[])
paste("Точність:", accuracy)
[1] "Точність: 0.817223198594025"
```

Отриманий ліс має точність на тестовому наборі даних приблизно 81,7% та є кращим за дерево прийняття рішень. Крос-валідацію для випадкового лісу можна не застосовувати, оскільки ліс стійкий до незначних атрибутів та менше піддається перенавчанню.

ЗМІСТОВИЙ МОДУЛЬ 5. МЕТОДИ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ТА ДИСКРИМІНАНТНИЙ АНАЛІЗ

Тема 8. Метод штучних нейронних мереж

Мета: ознайомитися із завданнями Data mining, що вирішуються з допомогою штучних нейронних мереж.

План

8.1 Класифікація нейронних мереж
8.2 Вибір структури нейронної мережі
8.3 Карти Кохонена
8.4 Карта входів та виходів нейронів

Основні поняття

Елементи нейронних мереж. Архітектура нейронних мереж. Навчання нейронних мереж. Моделі нейронних мереж.

8.1 Класифікація нейронних мереж

Одна з можливих класифікацій нейронних мереж – за спрямованістю зв'язків. Нейронні мережі бувають зі зворотними зв'язками й без зворотних зв'язків.

Мережі без зворотних зв'язків:

– мережі зі зворотним поширенням помилки. Мережі цієї групи характеризуються фіксованою структурою, ітераційним навчанням, коректуванням ваг за помилками.

– інші мережі (когнитрон, неокогнитрон, інші складні моделі).

Перевагами мереж без зворотних зв'язків є простота їх реалізації й гарантоване одержання відповіді після проходження даних по шарах.

Недоліком цього виду мереж вважається мінімізація розмірів мережі – нейрони багаторазово беруть участь в обробці даних.

Менший обсяг мережі полегшує процес навчання.

Мережі зі зворотними зв'язками:

– мережі Хопфілда (задачі асоціативної пам'яті);

– мережі Кохонена (задачі кластерного аналізу).

Перевагами мереж зі зворотними зв'язками є складність навчання, викликана більшим числом нейронів для алгоритмів того самого рівня складності.

Недоліки цього виду мереж – необхідність спеціальних умов, що гарантують збіжність обчислень.

Інша класифікація нейронних мереж: мережі прямого поширення й рекуррентні мережі.

Мережі прямого поширення:

– персептрони;

– мережа Back Propagation;

– мережа зустрічного поширення;

– карта Кохонена.

Рекуррентні мережі. Характерна риса таких мереж – наявність блоків динамічної затримки й зворотних зв'язків, що дозволяє їм обробляти динамічні моделі:

– мережа Хопфілда.

– мережа Елмана – мережа, що складається із двох шарів, у якій схований шар охоплений динамічним зворотним зв'язком, що дозволяє врахувати передісторію спостережуваних процесів і нагромадити інформацію для вироблення правильної стратегії управління. Ці мережі застосовуються в системах управління об'єктами, що рухаються.

Нейронні мережі можуть навчатися з учителем або без нього.

При навчанні з учителем для кожного навчального вхідного прикладу потрібне знання правильної відповіді або функції оцінки якості відповіді. Таке навчання називають керованим. Нейронній мережі пред'являються значення вхідних і вихідних сигналів, а вона за певним алгоритмом підбудовує ваги синоптичних зв'язків. У процесі навчання проводиться корегування ваг мережі за результатами порівняння фактичних вихідних значень із вхідними, відомими заздалегідь.

При навчанні без учителя розкривається внутрішня структура даних або кореляції між зразками в наборі даних. Виходи нейронної мережі формуються самостійно, а ваги змінюються за алгоритмом, що враховує тільки вхідні й похідні від них сигнали. Це навчання називають також некерованим. У результаті такого навчання об'єкти або приклади розподіляються по категоріях, самі категорії та їх кількість можуть бути заздалегідь не відомі.

При підготовці даних для навчання нейронної мережі необхідно звертати увагу на такі істотні моменти: кількість спостережень у наборі даних, наявність викидів, репрезентативність навчальної вибірки.

Слід враховувати той фактор, що чим більше розмірність даних, тим більше часу потрібно для навчання мережі. Слід визначити наявність викидів і оцінити необхідність їх присутності у вибірці. Навчальна вибірка не повинна містити протиріч, тому що нейронна мережа однозначно зіставляє вихідні значення із вхідними.

Нейронна мережа працює тільки із числовими вхідними даними, тому важливим етапом при підготовці даних є перетворення й кодування даних.

При використанні на вхід нейронної мережі слід подавати значення з того діапазону, на якому вона навчалася. Наприклад, якщо при навчанні нейронної мережі на один з її входів подавалися значення від 0 до 10, то при її застосуванні на вхід слід подавати значення із цього ж діапазону або прилеглих.

Існує поняття нормалізації даних. Метою нормалізації значень є перетворення даних до вигляду, який найбільше підходить для обробки, тобто дані, що надходять на вхід, повинні мати числовий тип, а їх значення повинні бути розподілені в певному діапазоні. Нормалізатор може приводити дискретні

дані до набору унікальних індексів або перетворювати значення, що лежать в довільному діапазоні, у конкретний діапазон. Нормалізація виконується шляхом розподілу кожного компонента вхідного вектора на довжину вектора, що перетворює вхідний вектор в одиничний.

8.2 Вибір структури нейронної мережі

Вибір структури нейронної мережі зумовлюється специфікою й складністю розв'язуваної задачі. Для розв'язання деяких типів задач розроблені оптимальні конфігурації.

У більшості випадків вибір структури нейронної мережі визначається на основі об'єднання досвіду й інтуїції розробника.

Однак існують основні принципи, якими слід керуватися при розробці нової конфігурації:

1) можливості мережі зростають зі збільшенням кількості гнізд мережі, щільності зв'язків між ними й кількості виділених шарів;

2) введення зворотних зв'язків поряд зі збільшенням можливостей мережі піднімає питання про динамічну стабільність мережі;

3) складність алгоритмів функціонування мережі (у тому числі, наприклад, введення декількох типів синапсів – збуджуючих, гальмуючих та ін.) також сприяє посиленню потужності нейронної мережі.

Питання про необхідні й достатні властивості мережі для розв'язання того або іншого типу задач являє собою цілий напрямок нейронної комп'ютерної науки. Оскільки проблема синтезу нейронної мережі суттєво залежить від розв'язуваної задачі, дати загальні докладні рекомендації важко. Очевидно, що процес функціонування нейронної мережі, тобто сутність дій, які вона здатна виконувати, залежить від величин синаптичних зв'язків. Розроблювач мережі повине задати певну структуру нейронної мережі, що відповідає якому-небудь завданню, та знайти оптимальні значення всіх змінних вагових коефіцієнтів (деякі синаптичні зв'язки можуть бути постійними).

8.3 Карти Кохонена

***Карти Кохонена*,** карти, що самоорганізуються (Self-Organizing Maps). Мережі, що називаються картами Кохонена, – це один із різновидів нейронних мереж, однак вони принципово відрізняються від розглянутих вище, оскільки використовують неконтрольоване навчання. Нагадаємо, що при такому навчанні навчальна множина складається лише зі значень вхідних змінних, у процесі навчання немає порівняння виходів нейронів з еталонними значеннями. Можна сказати, що така мережа вчиться розуміти структуру даних.

Ідея мережі Кохонена належить фінському вченому Тойво Кохонену (1982 рік). Основний принцип роботи мереж – введення в правило навчання нейрона інформації щодо його розташування.

В основі ідеї мережі Кохонена лежить аналогія із властивостями людського мозку. Кора головного мозку людини являє собою плаский аркуш зі згорнутими складками. Отже, можна сказати, що вона має певні топологічні

властивості (ділянки, відповідальні за близькі частини тіла, примикають одна до одної й усе зображення людського тіла відображається на цю двовимірну поверхню).

Карти, що самоорганізуються, можуть використовуватися для розв'язання таких завдань, як моделювання, прогнозування, пошук закономірностей у великих масивах даних, виявлення наборів незалежних ознак і стискання інформації.

Найпоширеніше застосування мереж Кохонена – розв'язання завдання класифікації без вчителя, тобто кластеризації.

Нагадаємо, що при такій постановці задачі задано набір об'єктів, кожному з яких зіставлений рядок таблиці (вектор значень ознак). Потрібно розбити вихідну множину на класи, тобто для кожного об'єкта знайти клас, до якого він належить.

У результаті одержання нової інформації про класи можлива корекція існуючих правил класифікації об'єктів.

Два з розповсюджених застосувань карт Кохонена: розвідницький аналіз даних і виявлення нових явищ.

Розвідницький аналіз даних. Мережа Кохонена здатна розпізнавати кластери в даних, а також установлювати близькість класів. Отже, користувач може поліпшити своє розуміння структури даних, щоб потім уточнити нейромережеву модель. Якщо в даних розпізнані класи, то їх можна позначити, після чого мережа зможе вирішувати задачу класифікації. Мережі Кохонена можна використовувати й у тих задачах класифікації, де класи вже задані, – тоді перевага буде в тому, що мережа зможе виявити подібність між різними класами.

Виявлення нових явищ. Мережа Кохонена розпізнає кластери в навчальних даних і відносить усі дані до тих або інших кластерів. Якщо після цього мережа зустрінеться з набором даних, несхожим ні на один із відомих зразків, то вона не зможе класифікувати такий набір і тим самим виявить його новизну.

Мережа Кохонена, на відміну від багатошарової нейронної мережі, дуже проста; вона являє собою два шари: вхідний і вихідний. Її також називають самоорганізованою картою. Елементи карти розташовуються в деякому просторі, як правило, двовимірному. Мережа Кохонена зображена на рис. 8.1.

Мережа Кохонена навчається методом послідовних наближень. У процесі навчання таких мереж на входи подаються дані, але мережа при цьому підбудовується не під еталонне значення виходу, а під закономірності у вхідних даних. Починається навчання з обраного випадковим чином вихідного розташування центрів.

У процесі послідовної подачі на вхід мережі навчальних прикладів визначається найбільш схожий нейрон (той, у якого скалярний добуток ваг і поданого на вхід вектора мінімальні). Цей нейрон оголошується переможцем і є центром при підстроюванні ваг у сусідніх нейронів. Таке правило навчання припускає «змагальне» навчання з урахуванням відстані нейронів від «нейрона-переможця».

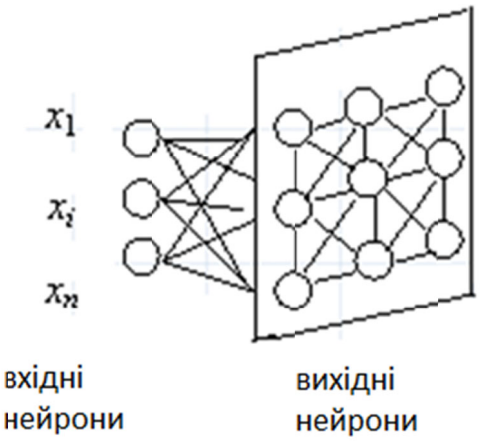

Рисунок 8.1 – Мережа Кохонена

Навчання при цьому полягає не в мінімізації помилки, а в підстроюванні ваг (внутрішніх параметрів нейронної мережі) для найбільшого збігу із вхідними даними.

Основний ітераційний алгоритм Кохонена послідовно проходить ряд епох, на кожній з яких обробляється один приклад із навчальної вибірки. Вхідні сигнали послідовно пред'являються мережі, при цьому бажані вихідні сигнали не визначаються. Після пред'явлення достатнього числа вхідних векторів синаптичні ваги мережі стають здатні визначити кластери. Ваги організують так, що топологічно близькі вузли чутливі до схожих вхідних сигналів.

У результаті роботи алгоритму центр кластера встановлюється в певній позиції, задовільним чином кластеризують приклади, для яких даний нейрон є «переможцем». У результаті навчання мережі необхідно визначити міру сусідства нейронів, тобто околицю нейрона-переможця.

Околиця являє собою кілька нейронів, які оточують нейрона-переможця.

Спочатку до околиці належить велика кількість нейронів, далі її розмір поступово зменшується. Мережа формує топологічну структуру, у якій схожі приклади утворюють групи прикладів, що близько перебувають на топологічній карті.

Отриману карту можна використовувати як засіб візуалізації при аналізі даних. У результаті навчання карта Кохонена класифікує вхідні приклади на кластери (групи схожих прикладів) і візуально відображає багатомірні вхідні дані на площині нейронів.

Унікальність методу карт, що самоорганізуються, полягає в перетворенні n-вимірного простору в двовимірний. Застосування двовимірних сіток пов'язане з тим, що існує проблема відображення просторових структур більшої розмірності.

Маючи таке представлення даних, можна візуально визначити наявність або відсутність взаємозв'язку у вхідних даних.

Нейрони карти Кохонена розташовують у вигляді двомірної матриці, розфарбовують цю матрицю залежно від аналізованих параметрів нейронів. На рис. 8.2 наведений приклад карти Кохонена.

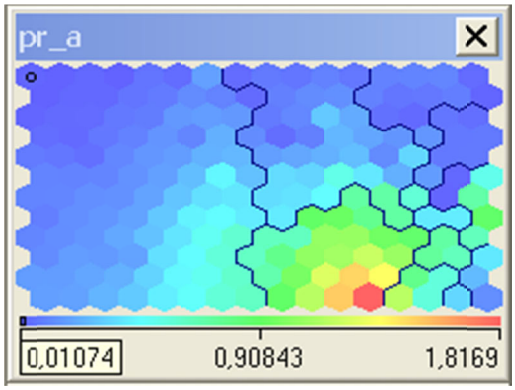

Рисунок 8.2 – Приклад карти Кохонена

На рис. 8.3 наведене розфарбування карти, а точніше, її *i*-ої ознаки, у тривимірному представленні. Як бачимо, темно-сині ділянки на карті відповідають найменшим значенням показника, червоні – найвищим.

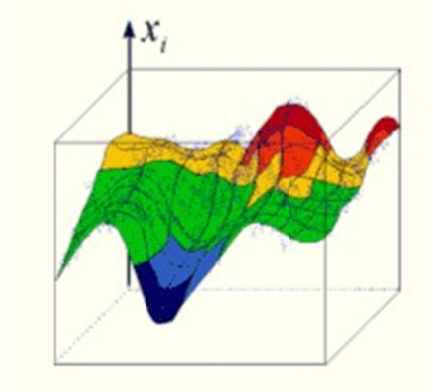

Рисунок 8.3 – Розфарбування *i*-ої ознаки в тривимірному просторі

Тепер, повертаючись до рисунку рис. 8.2, можна сказати, які об’єкти мають найбільші значення розглянутого показника (група об’єктів, позначена червоним кольором), а які – найменші значення (група об’єктів, позначена синім кольором).

Отже, карти Кохонена (як і географічні карти) можна відображати:

– у двомірному вигляді, тоді карта розфарбовується відповідно до рівня виходу нейрона;

– у тривимірному вигляді.

У результаті роботи алгоритму одержуємо такі карти:

– карта входів нейронів;

– карта виходів нейронів;

– спеціальні карти.

Координати кожної карти визначають положення одного нейрона.

Карта входів нейронів. Ваги нейронів підбудовуються під значення вхідних змінних і відображають їхню внутрішню структуру. Для кожного входу малюється своя карта, розфарбована у відповідності зі значенням конкретної ваги нейрона.

При аналізі даних використовують кілька карт входів. На одній із карт виділяють область певного кольору – це означає, що відповідні вхідні приклади мають приблизно однакове значення відповідного входу. Колірний розподіл нейронів із цієї області аналізується на інших картах для визначення схожих або відмінних характеристик.

Карта виходів нейронів. На карту виходів нейронів проєктується взаємне розташування досліджуваних вхідних даних. Нейрони з однаковими значеннями виходів утворюють кластери – замкнені області на карті, які включають нейрони з однаковими значеннями виходів.

Спеціальні карти. Це карта кластерів, матриця відстаней, матриця щільності потрапляння та інші карти, які характеризують кластери, отримані в результаті навчання мережі Кохонена.

Важливо розуміти, що між усіма розглянутими картами існує взаємозв'язок – усі вони є різними розфарбуваннями тих самих нейронів. Кожний приклад із навчальної вибірки має те саме розташування на всіх картах.

Питання для самоконтролю

1. Поясніть як відбувається процес навчання з вчителем нейронної мережі?
2. Як відбувається підготовка даних для навчання?
3. Дайте визначення поняттю карти Кохонена?
4. Для розв'язання яких задач можна застосовувати карти Кохонена?

Лабораторне заняття №6

Тема: Штучні нейронні мережі на мові R

Мета роботи: здобути навичкі роботи із нейронними мережами.

Завдання. Побудувати нейронну мережу на базі набіру даних audit.

Хід роботи.

Набір даних audit входить до бібліотеки *rattle.data*. набір даних аудиту – це штучно створений набір даних, що володіє певними характеристиками справжнього набору даних фінансового аудиту для моделювання продуктивних та непродуктивних аудитів фінансового звіту людини. Продуктивний аудит – це такий, що виявляє помилки та неточності у інформації, наданої клієнтом. Непродуктивний аудит – це зазвичай аудит, який виявив, що вся надана інформація у порядку.

Набір даних аудиту використовується для ілюстрації бінарної класифікації. Цільова змінна ідентифікується як TARGET_Adjusted. Набір даних не великий й складається лише з 2000 об'єктів. Його основна мета – проілюструвати моделювання у Rattle, тому підходить набір даних мінімального розміру. Сам набір отримано з загальнодоступних даних (які не мають нічого спільного з аудитами).

Структура набору даних:

ID – унікальний індикатор кожної людини;

Age – вік;

Employment – тип зайнятості;

Education – найвищий рівень освіти людини;

Marital – поточний родинний стан;

Occupation – тип заняття;

Income – заявлена сума доходу;

Gender – стать;

Deductions – загальна сума витрат, які людина відображає у своїй фінансовій звітності;

Hours – середній час роботи на тиждень, години;

IGNORE_Accounts – основна країна, у якій людина зберігає більшу частину своїх грошей (зверніть увагу, що перед ім'ям змінної стоїть IGNORE. Це визнається Rattle як роль за замовченням для цієї змінної);

RISK_Adjustment – ця змінна записує грошову суму будь-якого корегування фінансових претензій особи у результаті продуктивного аудиту, тобто, ця змінна розглядається не як вхідна змінна, а як міра ризику, пов'язаного з людиною;

TARGET_Adjusted –цільова змінна для моделювання (зазвичай для класифікаційного моделювання), це числове поле класу integer, але обмежене 0 та 1, що вказує на непродуктивний та продуктивний аудити відповідно. Продуктивні аудити – це ті, що призводять до корегування вінансової звітності клієнта.

1. Побудова моделі дерева.

Підготовка даних.

Підключіть бібліотеку *rattle.data* та подивіться структуру набору даних audit:

```
library(rattle.data)
str(audit)
```

Видаліть факторні значення у наборі даних та поле ID.

Наприклад, *audit$Employment<-NULL*

Нормалізуйте дані, обов'язкова умова нейронної мережі – дані мають бути від 0 до 1:

```
samplesize=0.70*nrow(audit)
set.seed(80)
index=base::sample(seq_len(nrow(audit)), size=samplesize)
audittrain=audit[index,]
audittest=audit[-index,]
max=apply(audit, 2, max)
min=apply(audit,2, min)
scaled=as.data.frame(scale(audit, center=min, scale=max-min))
```

Створіть два датасети для нейронної мережі:

trainNN=scaled[index,]
testNN=scaled[-index,]
colnames(trainNN)

Побудуйте нейронну мережу, використовуючи бібліотеку *neuralnet*:

library(neuralnet)
NN=neuralnet(TARGET_Adjusted~Age+Income+Deductions+Hours+RISK_ Adjustment, trainNN, ***hidden=1****, linear.output=F)*
plot(NN)

Кількість прихованих шарів **hidden=** встановіть самостійно за критерієм найменшої кількості кроків навчання.

Подивіться ваги у нейронній мережі:

NN$weights

Зробіть передбачення:

predict_testNN=compute(NN, testNN)
predict_testNN = (predict_testNN$net.result * (max(audit$TARGET_Adjusted) - min(audit$TARGET_Adjusted))) + min(audit$TARGET_Adjusted)

Побудуйте графік передбаченими значеннями та фактичними (рис. 8.4):

plot(audittest$TARGET_Adjusted, predict_testNN, col='blue', pch=16, ylab ="predicted NN", xlab = "real")
abline(0,1)

Розрахуйте root mean squard error (RMSE):

RMSE.NN = (sum((audittest$TARGET_Adjusted - predict_testNN)^2) / nrow(audittest))^0.5
RMSE.NN
[1] 0.1968703

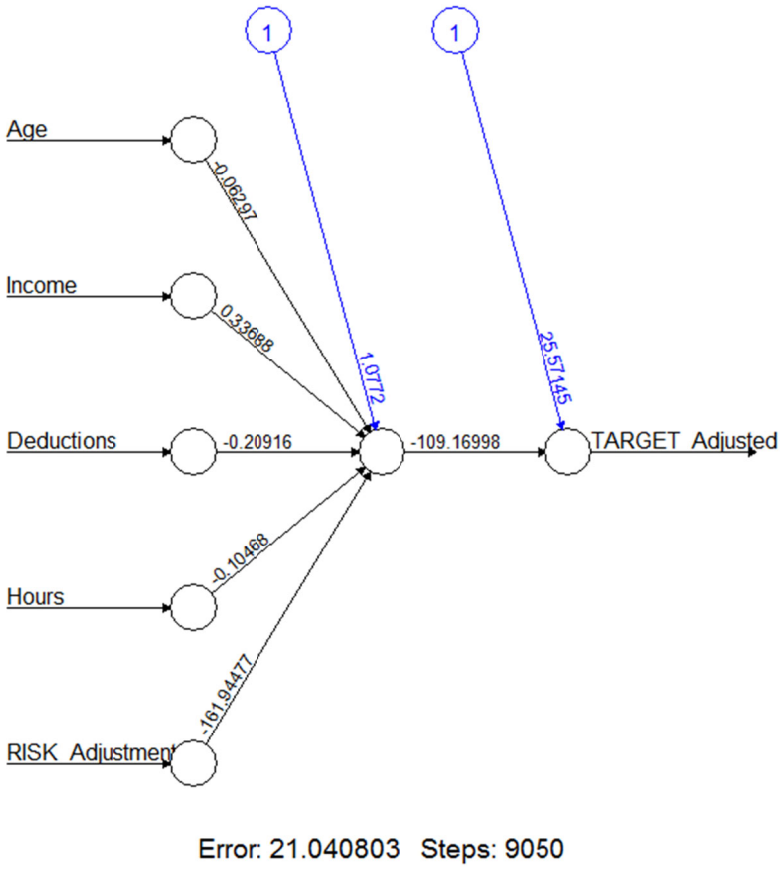

Рисунок 8.4 – Побудована нейронна мережа

Параметри функції neuralnet:

neuralnet(formula, data, hidden = 1, threshold = 0.01, stepmax = 1e+05, rep = 1, startweights = NULL, learningrate.limit = NULL, learningrate.factor = list(minus = 0.5, plus = 1.2), learningrate=NULL, lifesign = "none", lifesign.step = 1000, algorithm = "rprop+", err.fct = "sse", act.fct = "logistic", linear.output = TRUE, exclude = NULL, constant.weights = NULL, likelihood = FALSE)

formula ~ символічний опис моделі, що підлягає встановленню

data ~ дані, що містять змінні, які вказані у формулі

hidden ~ вектор цілих чисел, який визначає кількість прихованих нейронів (вершин) у кожному шарі

threshold ~ числове значення, що визначає поріг для часних похідних функцій похибки як критерій зупинки.

stepmax ~ максимальні кроки для навчання нейронної мережі. Досягнення цього максимуму призводить до зупинки процесу навчання нейронної мережі.

rep ~ кількість повторень для навчання нейронної мережі

startweights ~ вектор, що містить початкові значення для ваг. Ваги не будуть ініціалізовані випадковим чином.

learningrate.limit ~ вектор або список, що містить найбільш низьку та найбільш високу межу для швидкості навчання. Використовується тільки для RPROP та GRPROP.

learningrate.factor ~ або список, що містить коефіцієнти множення для верхньої та нижньої швидкості навчання. Використовується тільки для RPROP та GRPROP.

learningrate ~ числове значення, що визначає швидкість навчання, яка використовується традиційним оберненим розповсюдженням. Використовується тільки для традиційного backpropagation.

lifesign ~ рядок, який визначає, наскільки функція буде друкуватися під час обчислення нейронної мережі. ‘none’, ‘minimum’ або ‘full’.

lifesign.stepr ~ ціле число, яке визначає крок, щоби надрукувати мінімальну межу у режимі повного життя

algorithm ~ рядок, який містить тип алгоритму для обчислення нейронної мережі. Можливі такі типи: backprop, rprop +, rprop-, sag або slar. «backprop» відноситься до backpropagation, «rprop +» та «rprop-» відносяться до відмовостійкової backpropagation з поверненням backtracking та без нього, тоді як «sag» та «slr» викликають використання модифікованого глобально-конвергентного алгоритму (grprop).

err.fct ~ диференційована функція, що використовується для обчислення похибки. Як альтернативу можна використовувати рядки sse ‘та’ ce ‘, які означають суму квадратів похибок та крос-ентропії.

act.fct ~ диференційована функція, яка використовується для згладжування результату поперечного добутку коваріату або нейронів та ваг. Крім того, для логістичної функції та дотичной гіперболічності можливі рядки, «логістика» та «тань»

linear.output ~ логічний. Якщо act.fct не слід застосовувати до вихідних нейронів, встановіть лінійний вихід як TRUE, інакше FALSE

exclude ~ вектор або матриця, що визначають вагові коефіцієнти, які виключаються з розрахунку. Якщо задано як вектор, то точне положення ваг має бути відомо. Матриця з n-рядками та трьома стовпцями виключає n ваг, де перший стовпець означає шар, другий стовпець для вхідного нейрону та третій стовпець для вихідного нейрону ваги.

constant.weights ~ вектор, що визначає значення ваг, які виключаються з навчального процесу та розглядаються як виправленіня.

likelihood ~ логічний. Якщо функція похибки дорівнює від’ємній функції логарифмічної правдоподібності, будуть обчислені інформаційні критерії AIC та BIC. Крім того, використання trust.interval має сенс.

Тема 9. Метод дискримінантного аналізу

Мета: ознайомитися із методом дискримінантного аналізу, його проблемами та місцем застосування.

План

9.1 Дискримінантний аналіз

9.2 Відстань Махаланобіса

9.3 Проблема оцінки дискримінантних функцій та їх послідовного відбору

Основні поняття

Дискримінантний факторний аналіз. Геометричний прогнозний дискримінантний аналіз. Ймовірнісний дискримінантний аналіз. Лямбда Уїлкса. Коефіцієнт детермінації. Скорегований коефіцієнт детермінації.

9.1 Дискримінантний аналіз

Дискримінантний аналіз призначений для з'ясування того, до якого з відомих класів потрібно віднести новий об'єкт, що характеризується певною множиною значень кількісних ознак. Передбачається, що для кожного класу вже складені характерні вибірки об'єктів (навчальні підсукупності). Ці вибірки зазвичай складаються кваліфікованими спеціалістами-експертами, які для діагностики (віднесення об'єкта до того чи іншого класу) можуть використовувати всі доступні їм знання, в тому числі й неформалізовані.

Проблема процедури класифікації. Класичний дискримінантний аналіз складається з формулювання деяких правил, критеріїв класифікації, «інформантів», «дискримінантних функцій», які дозволяють оцінити ймовірність належності нового об'єкта до кожного класу. Зазвичай у якості таких критеріїв класифікації вибирають відстань Махаланобіса від об'єкта до центрів кожного класу. За цим критерієм об'єкт відноситься до найближчого класу. Цю стандартну методику (ядро, основу дискримінантного аналізу) можна доповнити. По-перше, деякі класи (групи) об'єктів більш поширені, розповсюджені, ніж інші, й за однакових умов мають деякі переваги. Якщо ми володіємо апріорною інформацією про кількісне співвідношення (про обсяг) класів, тобто про ймовірності q_i належності об'єктів до кожного класу ($q_1 + q_2 + \cdots + q_m = 1$), то цю додаткову інформацію слід врахувати у правилах класифікації (інформантах). По-друге, можна оцінити наслідки від можливих помилок класифікації. Якщо експерти складуть матрицю збитків c_{ij} (від неправильного віднесення об'єкта з класу i до класу j), то і цю додаткову інформацію можна врахувати в класифікаційних правилах, які тепер будуть оцінювати величину можливих збитків від віднесення об'єкта до того чи іншого класу. Об'єкт потрібно відносити до того класу, для якого ризик (величина можливих збитків) буде найменшим.

Теоретично можливо було б урахувати особливості багатовимірного розподілу об'єктів в кожному класі та прийняти замість відстані Махаланобіса обґрунтованішу міру. Проте ця можливість залишається

практично не реалізованою, оскільки з багатовимірних розподілів достатньо добре вивчений лише багатовимірний нормальний закон, що і визначає вибір в якості міри саме відстань Махаланобіса.

9.2 Відстань Махаланобіса

Відстань Махаланобіса є квадратичним інформантом – квадратичною функцією виміряних ознак об'єктів. Проте на практиці вводять додаткове спростовувальне припущення про рівність всіх коваріаційних матриць для кожного класу (класи тепер відрізняються тільки центрами групувань). Зазвичай навчальні підсукупності (характерні вибірки) надто малі, щоб для кожної вибірки достатньо надійно можна було визначити дисперсії і коваріації ознак – саме тому і доводиться використовувати загальну (об'єднану) коваріаційну матрицю. Вимушене прийняття припущення про рівність коваріаційних матриць для всіх класів істотно спростовує (та полегшує) класифікаційні правила – тепер замість квадратичних інформантів одержали зручніші лінійні інформанти. Викладене вище в загальних рисах описує методи класичного дискримінантного аналізу.

Останнім часом класичний дискримінантний аналіз доповнений теорією «канонічних дискримінантних функцій» або просто «дискримінантних функцій». У зв'язку з цим пропонується за класифікаційними правилами залишити назву «інформанти» та не використовувати для них назву «дискримінантні функції», що тепер будуть позначати дещо інше.

Дискримінантними функціями за новим змістом є деякі узагальнюючі (агреговані) змінні, в просторі яких найчіткіше видно відмінності між класами (відомими підсукупностями). Ці узагальнюючі змінні складаються у вигляді лінійних комбінацій початкових ознак, а коефіцієнти цих лінійних комбінацій визначаються з умови максимуму певного функціонала, що характеризує відмінності між класами. Декілька перших (головних) дискримінантних функцій вичерпно описують практично всі відмінності між класами.

Попередній перехід до скороченого простору головних дискримінантних функцій усуває багато проблем класичного дискримінантного аналізу – результати стають стійкішими й надійнішими. Справа в тому, що відстань Махаланобіса враховує кореляції між ознаками, що вважається позитивною властивістю даної міри. Але, з іншого боку, за високих кореляцій ця міра стає нестійкою, а за точної мультиколінеарності ознак – виродженою. Слід зазначити, що спеціалісти не зобов'язані перевіряти (та й не перевіряють) незалежність ознак у системі, що визначає об'єкт. У скороченому просторі (головних) дискримінантних функцій виродженість виключається, оскільки дискримінантні функції ортогональні. Крім того, часто для опису відмінностей між класами достатньо всього двох перших дискримінантних функцій і тоді з'являється можливість візуального відображення об'єктів кожного класу (і нового спірного об'єкта) на площині, що є надзвичайно цінним надбанням.

Теорія канонічних дискримінантних функцій дуже схожа за методами і цілями на теорію головних компонент факторного аналізу.

Проблема наочності (візуалізації) полягає у наступному. Впровадження канонічних дискримінантних функцій здійснюється через створення упорядкованих за значущістю функцій, аналогічних до головних компонентів факторного аналізу. Перша дискримінантна функція обирається так, щоб її середні значення для різних класів відрізнялись найістотніше. Друга дискримінантна функція добирається за ідентичних умов, але вона не повинна корелювати з першою. Третя функція не повинна корелювати з упровадженими раніше функціями. Тоді кілька перших дискримінантних функцій (головних) несуть усю інформацію про відмінність між класами.

Після завершення аналізу можна виконувати зворотний перехід у початковий простір реальних змінних.

Позначимо k – кількість класів; m – кількість дискримінантних ознак; h_i – кількість об'єктів (спостережень) класу p ; n – загальна кількість об'єктів усіх класів. У дискримінантному аналізі приймають наступні математичні допущення:

1) існує два чи більше класів $k >= 2$;

2) у кожному класі маємо принаймні два об'єкти $h_p >= 2$;

3) кількість дискримінантних ознак необмежена, але не перевищує загальну кількість об'єктів мінус два, $0 < m < (n - 2)$;

4) вимірювання дискримінантних ознак повинне бути здійснене за інтервальною шкалою;

5) дискримінантні ознаки лінійно незалежні;

6) передбачається приблизна рівність коваріаційних матриць для кожного класу(якщо не використовуються спеціальні прийоми);

7) приймається гіпотеза про багатовимірну нормальність закону розподілу об'єктів для кожного класу.

9.3 Проблема оцінки дискримінантних функцій та їх послідовного відбору

Дискримінантним методом передбачається обчислення таких числових характеристик:

1. Процентний вміст $\frac{\lambda_j}{\sum_{j=1}^{m}\lambda_j} * 100\%$ показує, наскільки важливіша та чи інша дискримінантна функція.

2. Квадрат коефіцієнта канонічної кореляції (кореляційного відношення) $\eta_i^2 = \frac{\lambda_i}{1+\lambda_i}$ показує, яка частка повної мінливості дискримінантної функції пояснюється відмінністю груп.

3. Критерій Фішера $F = \lambda\frac{n-k}{k-1}$, який можна порівняти з табличними значеннями $\mathrm{F}_{0.05}(k-1, n-k)$ і $F_{0.01}(k-1, n-k)$ та визначити, чи є значущою одержана дискримінантна функція.

4. Статистика Уїлкса. Нехай $l = \min\{k-1, m\}$ – загальна кількість дискримінантних функцій з ненульовими λ_i. Тоді $\Delta_0 = \frac{1}{1+\lambda_1} * \frac{1}{1+\lambda_2} * \frac{1}{1+\lambda_3} \ldots \frac{1}{1+\lambda_l}$ є

мірою остаточної мінливості, якщо врахувати всі дискримінантні функції, тобто λ_0 оцінює дискримінантну здатність усієї системи функцій. Тепер потрібно оцінити дискримінантну здатність системи без першої, найважливішої функції: $\Delta_1 = \frac{1}{1+\lambda_2} * \frac{1}{1+\lambda_3} \dots \frac{1}{1+\lambda_l}$.

Ця величина вже перевищує λ_0. Що ближче λ_1 до одиниці, тим слабшою є дискримінантна здатність інших функцій системи, які можна враховувати.

Потім слід обчислити λ_2, λ_z, … до λ_{l-1} Значущість послідовних значень Δ_j оцінюється за допомогою критерію Пірсона: $\chi_j^2 = -\left[n - \frac{m+k}{2} - 1\right] ln\Delta_j$, який слід порівнювати з табличними значеннями $\chi_{0,05}^2(v)$ і $\chi_{0,01}^2(v)$ де кількість ступенів свободи визначається як $v_j = (m-j)(k-j-1)$.

На певному етапі система функцій, що залишилась, відкидається, і можна отримати систему інформативних показників – систему головних дискримінантних функцій.

Дискримінантні функції корисні для візуальної наочності розподілу об’єктів за класами, а також для одержання стабільних інформантів, які слід складати з дискримінантних функцій, а не вихідних ознак. Отже, переходимо до створення класифікаційних функцій (інформантів), за чисельними значеннями яких можна визначати, до якого класу найімовірніше віднести той чи інший об’єкт.

Питання для самоконтролю

1. Назовіть числові характеристики, які обчислюються при дискримінантному методі.
2. Перелічте математичні допущення, які приймають у дискримінантному аналізі.
3. Опишіть проблему наочності (візуалізації).
4. Дайте визначення відстані Махаланобіса.
5. Опишіть проблему процедури класифікації.

Лабораторне заняття №7

Тема: Дискримінантний аналіз на мові R

Мета роботи: здобути практичні навички застосування дискримінантного аналізу.

Завдання. на основі набору даних, включеного до базового дистрибутиву R, побудувати моделі багатокласової класифікації, що оцінюють кожен з трьох видів рослини за даними проведених вимірювань, застосовуючи метод дискримінантного аналізу.

Хід роботи.

Набір даних iris входить до бібліотеки *datasets*. Іриси Фішера – найбільш популярний у статистичній літературі набір даних, який часто

використовується для ілюстрації роботи різних алгоритмів класифікації. У сучасних реальних додатках такі компактні набори даних, які дозволяють побудувати хороший класифікатор при мінімумі вхідних ознак, зустрічаються дуже рідко.

Вибірка складається з 150 екземплярів ірисів трьох видів, для яких вимірювалися чотири характеристики: довжина та ширина чашелистика (Sepal.Length та Sepal.Width), довжина та ширина пелюстки (Petal.Length та Petal.Width).

1. Дослідження характеру статистичної варіації ознак з використанням категорізованих діаграм, де дані згруповані за окремими видами:

```
library(datasets)
library(ggplot2)
library(gridExtra)
data(iris)
a <-qplot(Sepal.Length, Sepal.Width, data = iris) +
   facet_grid(facets = ~ Species) +
   geom_smooth(color = "red", se = FALSE)
b <- qplot(Petal.Length, Petal.Width,data = iris) +
   facet_grid(facets = ~ Species) +
   geom_smooth(color = "red", se = FALSE)
grid.arrange(a, b, nrow = 2)
```

2. Класифікація у лінійному дискримінаційному просторі

Підключіть бібліотеку *MASS*:

```
library(MASS)
```

Для стандартизованих даних розрахуйте коефіцієнти лінійних дискримінантів та параметр λ – відношення міжгрупової дисперсії до внутрішньогрупової дисперсії, обчисліть частку міжгрупової дисперсії, пояснену кожним лінійним дискримінантом:

```
LDA_iris<-lda(scale(iris[, 1:4]), gr = iris$Species)
LDA_iris
```

параметр λ:

```
LDA_iris$svd
```

Частка міжгрупової дисперсії, пояснену кожним лінійним дискримінантом:

```
(prop = LDA_iris$svd^2/sum(LDA_iris$svd^2))
```

Зробіть висновки за результатами аналізу.
Побудуйте ординаційну діаграму спостережень (рис.9.1):

```
prop <- scales::percent(prop)
pred <- predict(LDA_iris, newdata = iris)
scores <- data.frame(Species = iris$Species, pred$x)
ggplot() + geom_point(data = scores,
                      aes(x = LD1, y = LD2, shape = Species,
                          colour = Species), size = 3) +
   scale_colour_manual(values = c('purple', 'green', 'blue')) +
   labs(x = paste("LD1 (", prop[1], ")", sep = ""),
        y = paste("LD2 (", prop[2], ")", sep = "")) + theme_bw()
```

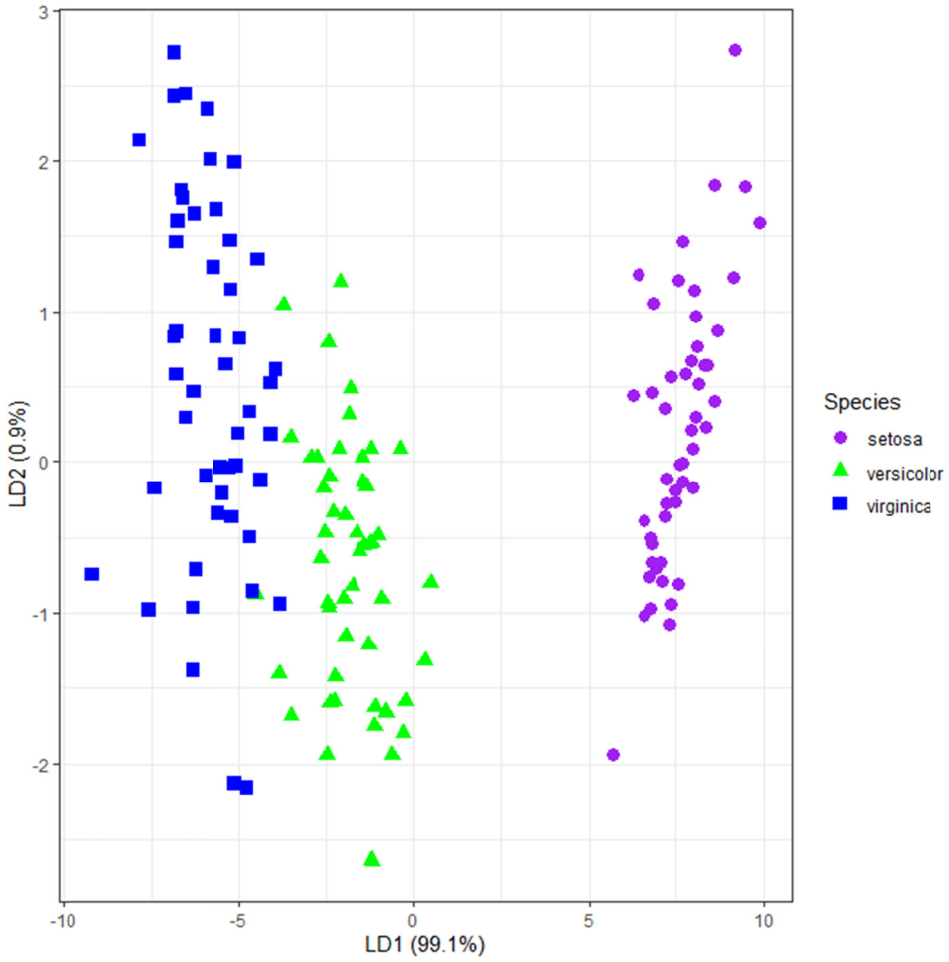

Рисунок 9.1 - Ординаційна діаграма спостережень

Виделіть з навчальної вибірки 10 об'єктів для перевірки результатів:

```
train <- sample(1:150, 140)
LDA_iris1 <- lda(Species ~ ., iris, subset = train)
plda = predict(LDA_iris1, newdata = iris[-train, ])
data.frame(Species = iris[-train, 5], plda$class, plda$posterior)
    Species plda.class      setosa   versicolor    virginica
```

```
7      setosa     setosa 1.000000e+00 9.755826e-20 3.324083e-39
28     setosa     setosa 1.000000e+00 8.037718e-23 1.345452e-43
34     setosa     setosa 1.000000e+00 5.978542e-30 1.474824e-52
44     setosa     setosa 1.000000e+00 1.417993e-16 4.246343e-34
65  versicolor versicolor 1.237444e-14 9.999991e-01 8.648938e-07
94  versicolor versicolor 8.137735e-15 9.999999e-01 5.039540e-08
101  virginica  virginica 8.380479e-55 4.291300e-09 1.000000e+00
107  virginica  virginica 9.621370e-35 5.911060e-02 9.408894e-01
133  virginica  virginica 5.525958e-48 2.143151e-06 9.999979e-01
148  virginica  virginica 7.527234e-37 2.793664e-03 9.972063e-01
```

Розрахуйте точність класифікації:

```
Acc <- mean(pred$class == iris$Species)
paste("Точність=", round(100*Acc, 2), "%", sep = "")
[1] "Точність=98%"
```

Проведіть ковзний контроль та розрахуйте для нього точність класифікації:

```
LDA_irisCV <- lda(Species ~ ., data = iris,  CV = TRUE)
(table(Факт = iris$Species, Прогноз = LDA_irisCV$class))
           Прогноз
Факт        setosa versicolor virginica
  setosa        50          0         0
  versicolor     0         48         2
  virginica      0          1        49
Acc <- mean(LDA_irisCV$class == iris$Species)
paste("Точність=", round(100*Acc, 2), "%", sep = "")
[1] "Точність=98%"
```

3. Проведіть квадратичний дискримінантний аналіз (QDA), використовуючи всі ознаки. Для цього підключіть бібліотеку *caret*.

```
library(caret)
set.seed(123)
 train(Species ~ ., data = iris, method = "qda",
+       trControl = trainControl(method = "cv"))
Quadratic Discriminant Analysis
150 samples
  4 predictor
  3 classes: ‘setosa’, ‘versicolor’, ‘virginica’
No pre-processing
Resampling: Cross-Validated (10 fold)
```

Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
Resampling results:
Accuracy Kappa
0.98 0.97

Повторіть цей аналіз лише з обраними ознаками. Зробіть висновки.

4. Проведіть регулятизований дискримінантний аналіз (RDA). Для цього підключіть бібліотеку *klaR*.

```
library(klaR)
set.seed(123)
```

Етап грубої оптимізації:

```
train(Species ~ ., data = iris, method = "rda",
+       trControl = trainControl(method = "cv"))
Regularized Discriminant Analysis
150 samples
  4 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
Resampling results across tuning parameters:
  gamma  lambda  Accuracy   Kappa
  0.0    0.0     0.9800000  0.97
  0.0    0.5     0.9800000  0.97
  0.0    1.0     0.9800000  0.97
  0.5    0.0     0.9533333  0.93
  0.5    0.5     0.9533333  0.93
  0.5    1.0     0.9533333  0.93
  1.0    0.0     0.9200000  0.88
  1.0    0.5     0.9200000  0.88
  1.0    1.0     0.9200000  0.88
Accuracy was used to select the optimal model using the
 largest value.
The final values used for the model were gamma = 0 and lambda = 1.
```

Етап з заданим діапазоном λ та γ (наприклад, λ від 0,1 до 0,5, γ – від 0,02 до 0,1):

```
RDAGrid <- expand.grid(.lambda = (1:5)/10, .gamma = (1:5)/50)
set.seed(123)
RDA.iris <- train(Species ~ ., data = iris, method = "rda",
```

```
    tuneGrid = RDAGrid, trControl = trainControl(method = "cv"))
RDA.iris
Regularized Discriminant Analysis
150 samples
  4 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
Resampling results across tuning parameters:
  lambda  gamma  Accuracy  Kappa
  0.1     0.02   0.98      0.97
  0.1     0.04   0.98      0.97
  0.1     0.06   0.98      0.97
  0.1     0.08   0.98      0.97
  0.1     0.10   0.98      0.97
  0.2     0.02   0.98      0.97
  0.2     0.04   0.98      0.97
  0.2     0.06   0.98      0.97
  0.2     0.08   0.98      0.97
  0.2     0.10   0.98      0.97
  0.3     0.02   0.98      0.97
  0.3     0.04   0.98      0.97
  0.3     0.06   0.98      0.97
  0.3     0.08   0.98      0.97
  0.3     0.10   0.98      0.97
  0.4     0.02   0.98      0.97
  0.4     0.04   0.98      0.97
  0.4     0.06   0.98      0.97
  0.4     0.08   0.98      0.97
  0.4     0.10   0.98      0.97
  0.5     0.02   0.98      0.97
  0.5     0.04   0.98      0.97
  0.5     0.06   0.98      0.97
  0.5     0.08   0.98      0.97
  0.5     0.10   0.98      0.97
Accuracy was used to select the optimal model using the
 largest value.
The final values used for the model were gamma = 0.02 and lambda
 = 0.5.
```

Оцінка точності ітогової моделі:

```
pred <- predict(RDA.iris)
```

```
Acc <- mean(pred == iris$Species)
paste("Точність=", round(100*Acc, 2), "%", sep = "")
[1] "Точність=98%"
```

Зробіть висновки.

ЗМІСТОВИЙ МОДУЛЬ 6. МЕТОД АНАЛІЗУ ЧАСОВИХ РЯДІВ

Тема 10. Метод аналізу часових рядів

Мета: ознайомитися із аналізом часових рядів із використанням середовища R.

План

10.1 Задачі прогнозування часових рядів
10.2 Прогнозування і часові ряди
10.3 Тренд, сезонність і цикл
10.4 Види помилок та прогнозів

Основні поняття

Послідовність етапів регресійного аналізу. Завдання регресійного аналізу. Тренд. Сезонність. Цикл.

10.1 Задачі прогнозування часових рядів

Задачі прогнозування розв'язуються в найрізноманітніших сферах людської діяльності, таких як наука, економіка, виробництво й безліч інших сфер. Прогнозування є важливим елементом організації управління як окремими господарюючими суб'єктами, так і економікою в цілому.

Розвиток методів прогнозування безпосередньо пов'язаний із розвитком інформаційних технологій, зокрема, із зростанням обсягів збережених даних і ускладненням методів і алгоритмів прогнозування, реалізованих в інструментах Data Mining.

Задача прогнозування, мабуть, може вважатися однією з найбільш складних задач Data Mining, вона вимагає ретельного дослідження вихідного набору даних і методів, що задовольняють аналізу.

Прогнозування (від грецького Prognosis), у широкому розумінні цього слова, визначається як випереджаюче відображення майбутнього**.**

Метою прогнозування є передбачення майбутніх подій.

Прогнозування (forecasting) є однією з задач Data Mining і одночасно одним із ключових моментів при прийнятті рішень.

Прогностика (prognostics) – теорія й практика прогнозування.

Прогнозування спрямоване на визначення тенденцій динаміки конкретного об'єкта або події на основі ретроспективних даних, тобто аналізу його стану колись і тепер. Отже, розв'язок задачі прогнозування вимагає деякої навчальної вибірки даних.

Прогнозування – установлення функціональної залежності між залежними й незалежними змінними.

Прогнозування є розповсюдженим і затребуваним завданням у багатьох сферах людської діяльності. У результаті прогнозування зменшується ризик прийняття невірних, необґрунтованих або суб'єктивних рішень.

Приклади задач прогнозування: прогноз руху грошових коштів, прогнозування врожайності агрокультури, прогнозування фінансової стабільності підприємства.

Крім економічної й фінансової сфери, задачі прогнозування постають в медицині, фармакології; популярним зараз стає політичне прогнозування.

Загалом розв'язок задачі прогнозування зводиться до розв'язку таких підзадач:

– вибір моделі прогнозування;

– аналіз адекватності й точності побудованого прогнозу.

Прогнозування подібне із задачею класифікації. Багато методів Data Mining використовуються для розв'язку задач класифікації і прогнозування. Це, наприклад, лінійна регресія, нейронні мережі, дерева рішень (які, іноді, так і називають – дерева прогнозування й класифікації).

Задачі класифікації й прогнозування мають подібності й відмінності. При розв'язку обох задач використовується двоетапний процес побудови моделі на основі навчального набору та її використання для прогнозування невідомих значень залежної змінної. Відмінність задач класифікації й прогнозування полягає в тому, що в першій задачі передбачається клас залежної змінної, а в другій – числові значення залежної змінної, пропущені або невідомі (які відносяться до майбутнього).

10.2 Прогнозування і часові ряди

Основою для прогнозування служить історична інформація, що зберігається в базі даних у вигляді ***часових рядів***.

Існує поняття ***Data Mining часових рядів*** (Time-Series Data Mining). На основі ретроспективної інформації у вигляді часових рядів можливий розв'язок різних задач Data Mining.

Приведемо дві принципові відмінності часового ряду від простої послідовності спостережень:

– члени часового ряду, на відміну від елементів випадкової вибірки, не є статистично незалежними.

– члени часового ряду не є однаково розподіленими.

Часовий ряд – послідовність спостережуваних значень будь-якої ознаки, упорядкованих у невипадкові моменти часу.

Відмінністю аналізу часових рядів від аналізу випадкових вибірок є припущення про рівні проміжки часу між спостереженнями та їх хронологічний порядок. Прив'язка спостережень до часу відіграє тут ключову роль, тоді як при аналізі випадкової вибірки вона не має ніякого значення. Типовий приклад часового ряду – дані біржових торгів.

Інформація, накопичена в різноманітних базах даних підприємства, є часовими рядами, якщо вона розташована в хронологічному порядку і зроблена в послідовні моменти часу.

Аналіз часового ряду здійснюється з метою:

– визначення природи ряду;

– прогнозування майбутніх значень ряду.

У процесі визначення структури й закономірностей часового ряду передбачається виявлення: шумів і викидів, тренду, сезонного компонента,

циклічного компонента. Визначення природи часового ряду може бути використане як своєрідна «розвідка» даних. Знання аналітика про наявність сезонного компонента необхідне, наприклад, для визначення кількості записів вибірки, яка повинна брати участь у побудові прогнозу.

Аналіз часового ряду ускладнюють ***шуми й викиди***. Існують різні методи визначення й фільтрації викидів, що дають можливість виключити їх з метою більш якісного Data Mining.

10.3 Тренд, сезонність і цикл

Основними складовими часового ряду є тренд і сезонний компонент.

Тренд є систематичним компонентом часового ряду, який може змінюватися в часі.

Трендом називають невипадкову функцію, яка формується під дією загальних або довгочасних тенденцій, що впливають на часовий ряд.

Прикладом тенденції може виступати, наприклад, фактор зростання досліджуваного ринку.

Автоматичного способу виявлення трендів у часових рядах не існує. Але якщо часовий ряд включає монотонний тренд (тобто відзначене його стійке зростання або стійке спадання), аналізувати часовий ряд у більшості випадків неважко.

Існує велика різноманітність постановок задач прогнозування, які можна поділити на дві групи: прогнозування односерійних рядів і прогнозування мультисерійних, або взаємовпливаючих, рядів.

Група прогнозування односерійних рядів включає задачу побудови прогнозу однієї змінної за ретроспективними даними тільки цієї змінної, без врахування впливу інших змінних і факторів.

Група прогнозування мультисерійних, або взаємовпливаючих, рядів включає задачу аналізу, де необхідно враховувати взаємовпливаючі фактори на одну або декілька змінних.

Крім розподілу на класи по односерійності й багатосерійності, ряди також бувають сезонними й несезонними.

Останній розподіл має на увазі наявність або відсутність у часового ряду такої складової як сезонність, тобто включення ***сезонного компонента***. Сезонна складова часового ряду є періодично повторюваним компонентом часового ряду.

Властивість сезонності означає, що через приблизно рівні проміжки часу форма кривої, яка описує поведінку залежної змінної, повторює свої характерні обриси. Властивість сезонності важлива при визначенні кількості ретроспективних даних, які будуть використовуватися для прогнозування.

Наприклад, на рис. 10.1 наведено фрагмент ряду, який ілюструє поведінку змінної «обсяги продажу товару X» за період, що становить один місяць. При вивченні кривої, наведеної на рисунку, аналітик не може зробити припущень щодо повторюваності форми кривої через рівні проміжки часу.

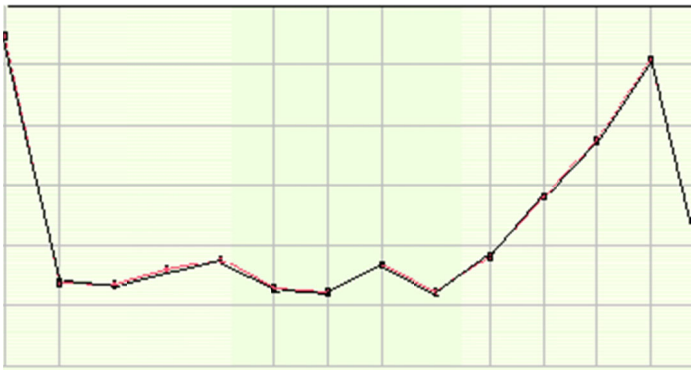

Рисунок 10.1 – Фрагмент часового ряду за сезонний період

Однак при розгляді більш тривалого ряду (за 12 місяців), зображеного на рис. 10.2, можна побачити явну наявність сезонного компонента. Отже, про сезонність продажів можна говорити тільки, коли розглядаються дані за кілька місяців.

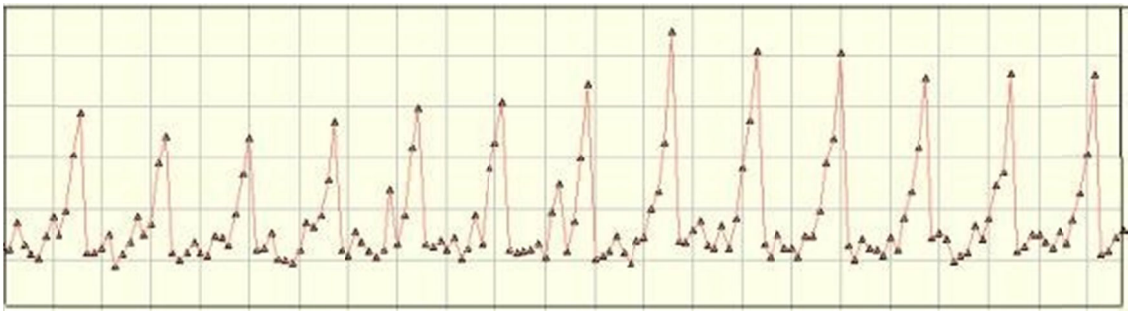

Рисунок 10.2 – Фрагмент часового ряду з 12-ти сезонних періодів

Отже, у процесі підготовки даних для прогнозування аналітикові слід визначити, чи має ряд, який він аналізує, властивість сезонності.

Визначення наявності компоненти сезонності необхідне для того, щоб вхідна інформація мала властивість репрезентативності.

Ряд можна вважати несезонним, якщо при розгляді його зовнішнього вигляду не можна зробити припущень про повторюваність форми кривої через рівні проміжки часу.

Іноді по зовнішньому вигляду кривої ряду не можна визначити, є він сезонним чи ні.

Існує поняття сезонного мультиряду. У ньому кожний ряд описує поведінку факторів, які впливають на залежну (цільову) змінну. Приклад такого ряду – ряди продажів декількох товарів, що піддаються сезонним коливанням.

При зборі даних і виборі факторів для розв’язку задачі прогнозування в таких випадках слід ураховувати, що вплив обсягів продажів товарів один на одного тут набагато менше, ніж вплив фактору сезонності.

Важливо не плутати поняття сезонного компонента ряду й сезонів природи. Незважаючи на близькість їх звучання, ці поняття відрізняються. Так, наприклад, обсяги продажів морозива влітку набагато більше, ніж в інші сезони, однак це є тенденцією попиту на даний товар.

Дуже часто тренд і сезонність присутні в часовому ряді одночасно. Наприклад, прибуток фірми зростає протягом декількох років (тобто в часовому ряді присутній тренд); ряд також містить сезонний компонент.

Відмінності циклічного компонента від сезонного:

1. Тривалість циклу, як правило, більше, ніж один сезонний період.

2. Цикли, на відміну від сезонних періодів, не мають певної тривалості.

При виконанні яких-небудь перетворень зрозуміти природу часового ряду значно простіше, такими перетвореннями можуть бути, наприклад, видалення тренда й згладжування ряду.

Перед початком прогнозування необхідно відповісти на такі питання:

1. Що потрібно прогнозувати?

2. У яких часових елементах (параметрах)?

3. З якою точністю прогнозу?

При відповіді на перше питання, ми визначаємо змінні, які будуть прогнозуватися. Це може бути, наприклад, рівень проведення конкретного виду продукції в наступному кварталі, прогноз суми продажу цієї продукції і т.д.

При виборі змінних слід ураховувати доступність ретроспективних даних, переваги осіб, що ухвалюють рішення, остаточну вартість Data Mining.

Часто при розв’язку задач прогнозування виникає необхідність прогнозування не самої змінної, а зміни її значень.

Друге питання при розв’язку задачі прогнозування – визначення таких параметрів:

– періоду прогнозування;

– горизонту прогнозування;

– інтервалу прогнозування.

Період прогнозування – основна одиниця часу, на яку робиться прогноз. Наприклад, ми прагнемо довідатися дохід компанії через місяць. Період прогнозування для цієї задачі – місяць.

Горизонт прогнозування – це число періодів у майбутньому, які покриває прогноз. Якщо ми прагнемо дізнатися прогноз на 12 місяців уперед, із даними по кожному місяцю, то період прогнозування в цьому завданні – місяць, горизонт прогнозування – 12 місяців.

Інтервал прогнозування – частота, з якою робиться новий прогноз. Інтервал прогнозування може збігатися з періодом прогнозування.

При виборі параметрів необхідно враховувати, що горизонт прогнозування повинен бути не менше, ніж час, який необхідний для реалізації розв’язку, прийнятого на основі цього прогнозу. Тільки в цьому випадку прогнозування буде мати сенс.

Зі збільшенням горизонту прогнозування точність прогнозу, як правило, знижується, а зі зменшенням горизонту – підвищується.

Ми можемо поліпшити якість прогнозування, зменшуючи час, необхідний на реалізацію розв’язку, для якого реалізується прогноз, і, отже, зменшити при цьому горизонт і помилку прогнозування.

При виборі інтервалу прогнозування слід вибирати між двома ризиками: вчасно не визначити зміни в аналізованому процесі й високою вартістю прогнозу. При тривалому інтервалі прогнозування виникає ризик не ідентифікувати зміни, які відбуваються в процесі, при короткому – зростають витрати на прогнозування.

При виборі інтервалу необхідно також ураховувати стабільність аналізованого процесу й вартість проведення прогнозу.

Точність прогнозу, що необхідна для розв'язку конкретної задачі, дуже впливає на прогнозуючу систему. Помилка прогнозу залежить від використовуваної системи прогнозу.

Чим більше ресурсів має така система, тим більше шансів одержати більш точний прогноз. Однак прогнозування не може повністю усунути ризики при прийнятті рішень. Тому завжди враховується можлива помилка прогнозування.

10.4 Види помилок та прогнозів

Точність прогнозу характеризується помилкою прогнозу.

Найпоширеніші види помилок: середня помилка, середня абсолютна помилка, сума квадратів помилок, відносна помилка.

Середня помилка (СП) обчислюється простим усередненням помилок на кожному кроці. Недолік цього виду помилки – позитивні й негативні помилки анулюють одна одну.

Середня абсолютна помилка (САП) розраховується як середнє абсолютних помилок. Якщо вона дорівнює нулю, то ми маємо досконалий прогноз. У порівнянні із середньою квадратичною помилкою, цей захід «не надає занадто великого значення» викидам.

Сума квадратів помилок (SSE), середньоквадратична помилка обчислюється як сума (або середнє) квадратів помилок. Це найбільше часто використовувана оцінка точності прогнозу.

Відносна помилка (ВП) виражає якість припасування в термінах відносних помилок.

Прогноз може бути короткостроковим, середньостроковим і довгостроковим.

Короткостроковий прогноз являє собою прогноз на кілька кроків уперед, тобто здійснюється побудова прогнозу не більше ніж на 3% від обсягу спостережень або на 1-3 кроку вперед.

Середньостроковий прогноз **–** це прогноз на 3-5% від обсягу спостережень, але не більш 7-12 кроків уперед; також під цим типом прогнозу розуміють прогноз на один або половину сезонного циклу. Для побудови короткострокових і середньострокових прогнозів цілком підходять статистичні методи.

Довгостроковий прогноз **–** це прогноз більш ніж на 5% від обсягу спостережень.

При побудові даного типу прогнозів статистичні методи практично не використовуються, крім випадків дуже «гарних» рядів, для яких прогноз можна просто «намалювати».

Дотепер ми розглядали аспекти прогнозування, так чи інакше пов'язані із процесом ухвалення рішення. Існують і інші фактори, які необхідно враховувати при прогнозуванні.

Задача 1. Відомо, що аналізований процес відносно стабільний у часі, зміни відбуваються повільно, процес не залежить від зовнішніх факторів.

Задача 2. Аналізований процес нестабільний і дуже сильно залежить від зовнішніх факторів.

Розв'язок першої задачі повинен бути зосереджений на використанні великої кількості ретроспективних даних. При розв'язку другої задачі особливу увагу слід звернути на оцінки фахівця в предметній області, експерта, щоб мати можливість відобразити в прогнозуючій моделі всі необхідні зовнішні фактори, а також приділити час для збору даних по цих факторах (збір зовнішніх даних часто набагато складніший збору внутрішніх даних інформаційної системи). Доступність даних, на основі яких буде здійснюватися прогнозування, – важливий фактор побудови прогнозної моделі. Для можливості виконання якісного прогнозу дані повинні бути представницькими, точними й достовірними.

Серед розповсюджених методів Data Mining, використовуваних для прогнозування, відзначимо ***нейронні мережі*** й ***лінійну регресію***.

Вибір методу прогнозування залежить від багатьох факторів, у тому числі від параметрів прогнозування. Вибір методу слід провадити з обліком усіх специфічних особливостей набору ретроспективних даних і цілей, заради яких він будується.

Програмне забезпечення Data Mining, використовуване для прогнозування, повинно забезпечувати користувача точним і достовірним прогнозом. Однак одержання такого прогнозу залежить не тільки від програмного забезпечення й методів, закладених у його основу, але також і від інших факторів, серед яких повнота й вірогідність вихідних даних, своєчасність і оперативність їх поповнення, кваліфікація користувача.

Питання для самоконтролю

1. Назвіть мету та визначення поняттю прогнозування.
2. Дайте визначення поняттю часовий ряд.
3. Які існують помилки прогнозу?
4. Які існують види прогнозів? Чим вони відрізняються?

Лабораторне заняття №8

Тема: Аналіз часових рядів на мові R

Мета роботи: здобути навичкі аналізу часових радів на мові R.

Завдання. Побудувати модель множинної регресії на заданих данних та проаналізувати її.

Хід роботи.

Набір даних longley входить до бібліотеки *datasets*. Цей набір макроекономічних даних є добре відомим прикладом висококолінеарної регресії. Набір даних складається з 7 економічних показників, які спостерігалися протягом 16 років (1947-1962).

Структура набору даних:

GNP.deflator – неявний дефлятор цін (1954р. =100);

GNP – валовий національний продукт;

Unemployed – кількість безробітних;

Armed.Forces – кількість людей у збройних силах;

Population – кількість населення 914 років й старше);

Year – рік;

Employed – кількість зайнятих.

1. Ознайомтесь з даними, зробіть візуалізацію даних, побудуйте гістограми змінних(рис. 10.3):

```
library(datasets)
library(ggplot2)
library(gridExtra)
data(longley)
str(longley)
x<-longley[ ,1:7]
par(mfrow=c(1,7))
  for(i in 1:7) {
  boxplot(x[,i], main=names(longley)[i])
  }
```

Побудуємо, наприклад, гістограму дефлятора цін (рис. 10.4):

```
hist.GNP.deflator<-ggplot(longley, aes(GNP.deflator))+
+ geom_histogram(aes(y=..density..))+theme_bw()+
+ labs(x='Дефлятор цін', y='Щільність')
hist.GNP.deflator+stat_function(fun=dnorm,args=list(mean=mean(longley$GNP.deflator, na.rm=TRUE),sd=sd(longley$GNP.deflator, na.rm=TRUE)), color='red')
```

Аналогічно побудуйте гістограми інших змінних (GNP, Unemployed, Armed.Forces, Population, Employed).

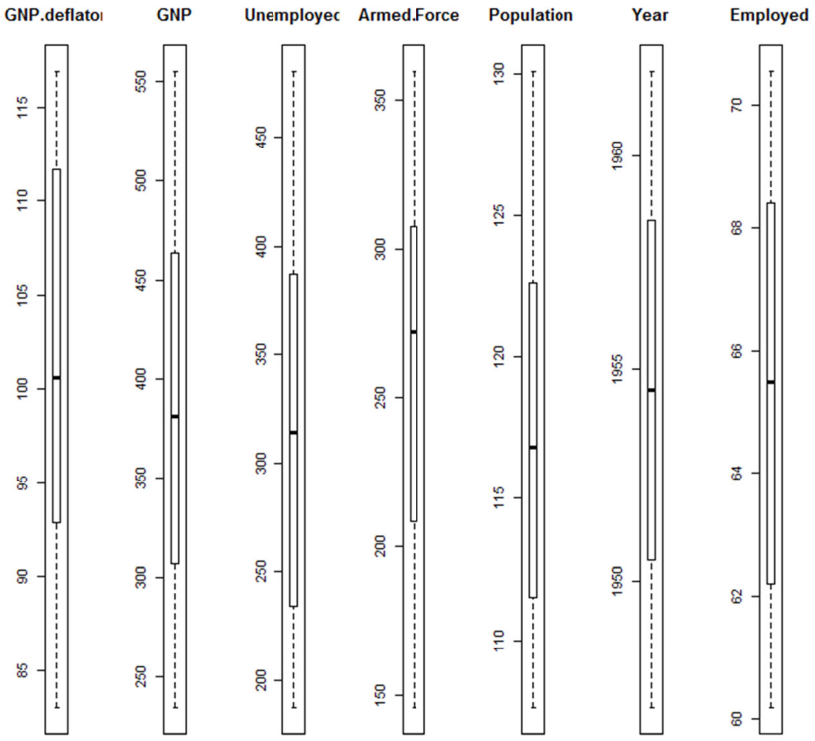

Рисунок 10.3 – Гістограма змінних

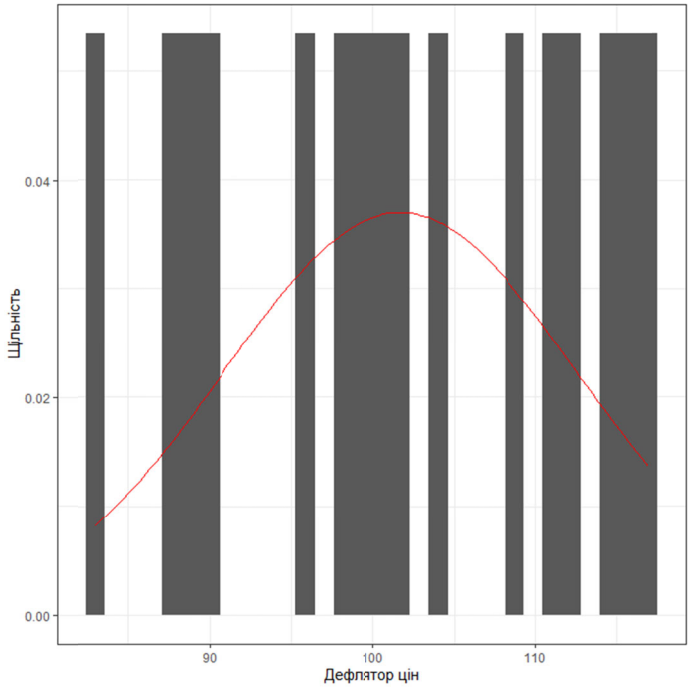

Рисунок 10.4 – Гістограма дефлятора цін

Для кількості зайнятих подивіться на дані qq-графіка (квантіль-квантіль) (рис. 10.5). Квантіль – частка випадків, менших певного значення. Якщо вимога “нормальності” виконується, то дані мають лежати на одній лінії.

qqplot.emloyed<-qplot(sample= longley$Employed, stat=‘qq’)

qqplot.emloyed +theme_bw()

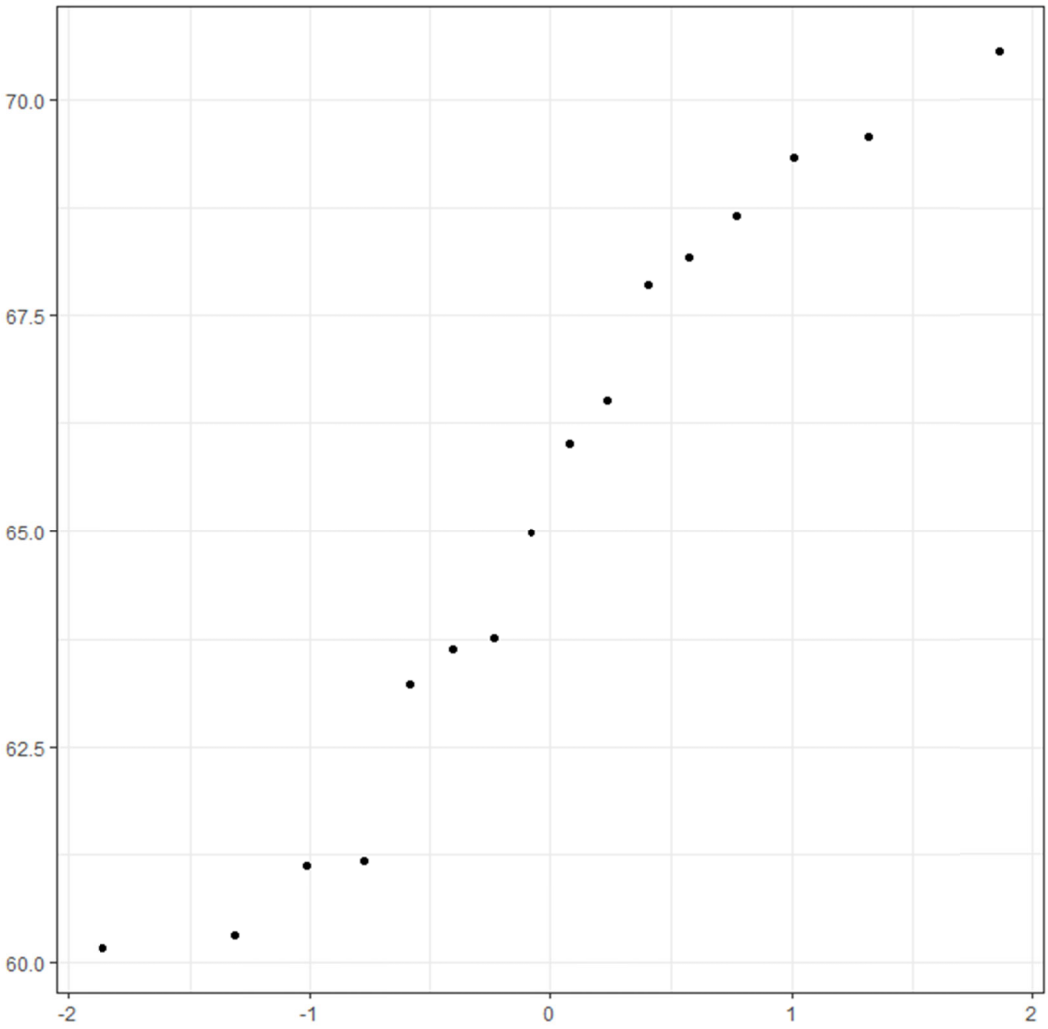

Рисунок 10.5 – qq-графік

2. Побудуйте модель множинної регресії.
Побудуємо співвідношення змінних (рис.10.6):

library(car)
scatterplotMatrix(longley, spread=FALSE, lty.smooth=2, main='Співвідношення змінних')

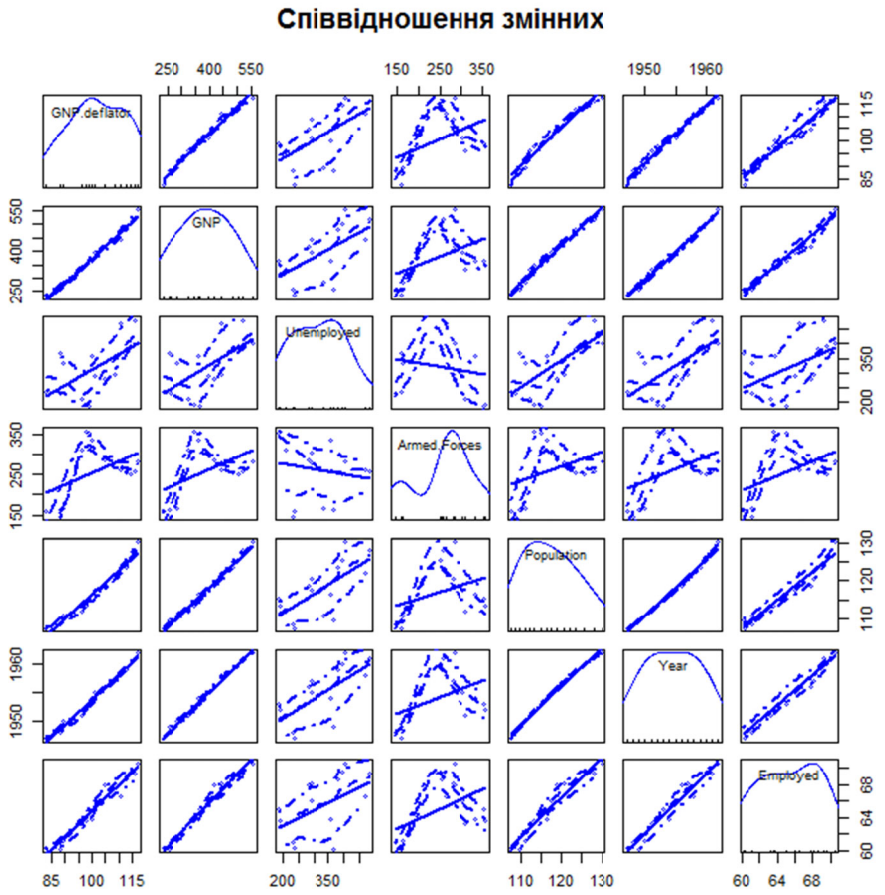

Рисунок 10.6 – Співвідношення змінних

Розрахуємо модель:

```
model<-lm(Employed~., data=longley)
summary(model)
Call:
lm(formula = Employed ~ ., data = longley)
Residuals:
    Min      1Q  Median      3Q     Max
-0.41011 -0.15767 -0.02816  0.10155  0.45539
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 **
GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
GNP          -3.582e-02  3.349e-02  -1.070 0.312681
Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
Population   -5.110e-02  2.261e-01  -0.226 0.826212
Year          1.829e+00  4.555e-01   4.016 0.003037 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error: 0.3049 on 9 degrees of freedom
Multiple R-squared: 0.9955, Adjusted R-squared: 0.9925
F-statistic: 330.3 on 6 and 9 DF, p-value: 4.984e-10
confint(model)

	2.5 %	97.5 %
(Intercept)	*-5.496529e+03*	*-1.467988e+03*
GNP.deflator	*-1.770290e-01*	*2.071528e-01*
GNP	*-1.115811e-01*	*3.994274e-02*
Unemployed	*-3.125067e-02*	*-9.153930e-03*
Armed.Forces	*-1.517949e-02*	*-5.485050e-03*
Population	*-5.625172e-01*	*4.603090e-01*
Year	*7.987875e-01*	*2.859515e+00*

Зробіть висновок про значущість параметрів моделі. Побудуйте нову модель тільки зі значущими параметрами.

3. Аналіз моделі (рис.10.7).

Нормальність – похибки мають нормальний розподіл:

qqPlot(model, labels=row.names(longley), simulate=TRUE, main='Графік Q-Q')

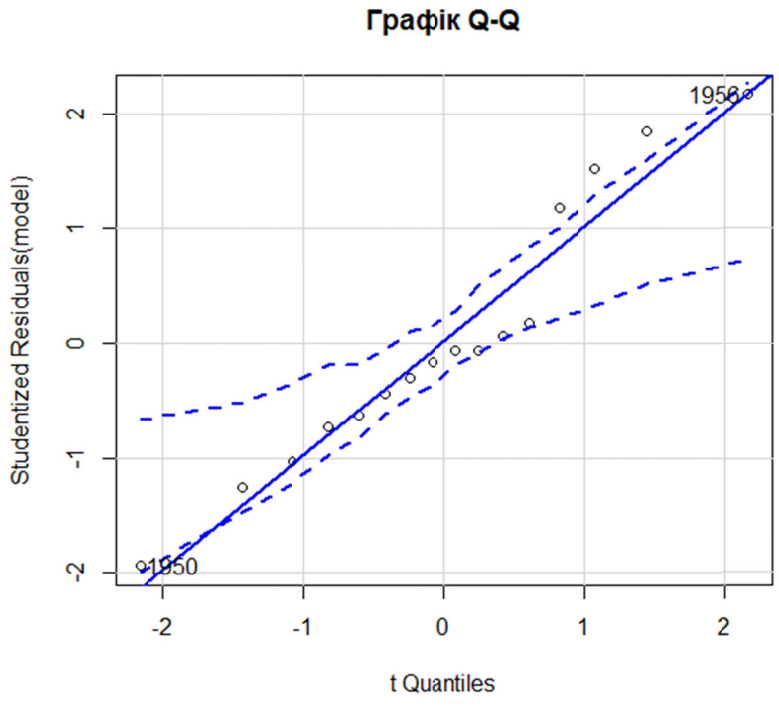

Рисунок 10.7 – Аналіз моделі

Незалежність похибок – тест Дарбіна-Уотсона:

durbinWatsonTest((model))
lag Autocorrelation D-W Statistic p-value

1 -0.3480223 2.559488 0.988
Alternative hypothesis: rho != 0

Результат – немає автокореляції.
Лінійність(рис.10.8):

crPlots((model))

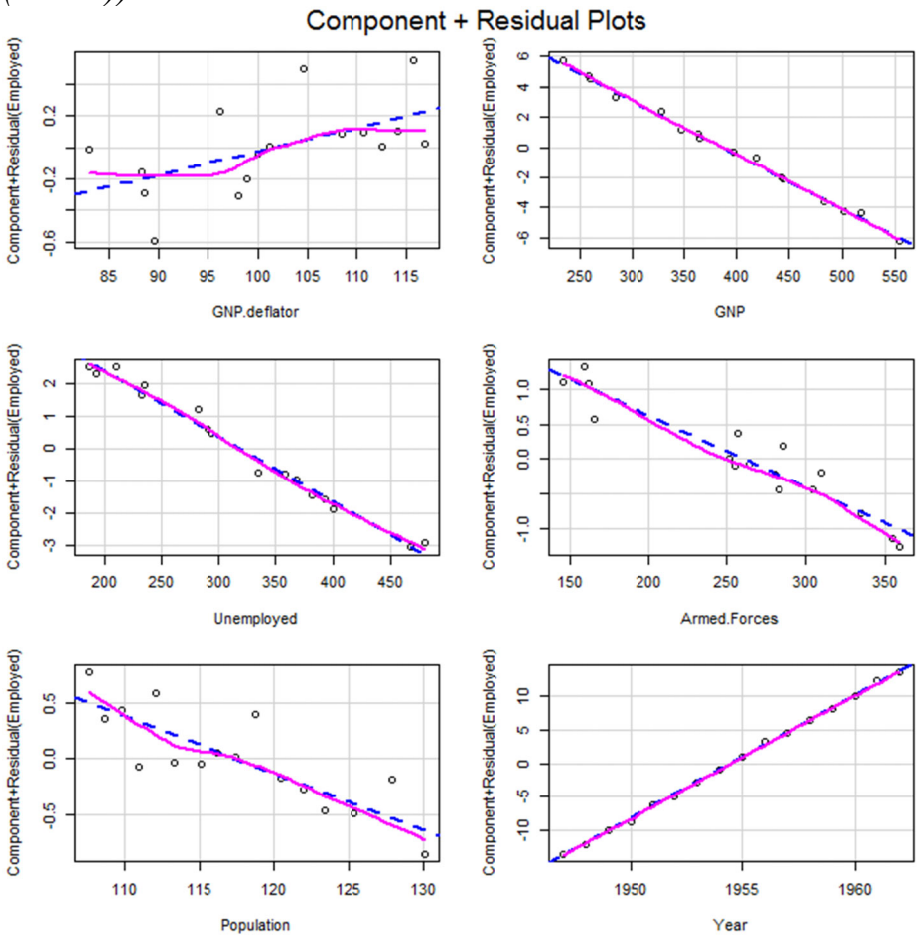

Рисунок 10.8 – Аналіз на нелінійні зв'язки

Результат – нелінійні зв'язки відсутні.
Проведемо тест на гомоскедастичність:

ncvTest((model))
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.02539171, Df = 1, p = 0.87339
spreadLevelPlot(model)
Suggested power transformation: 2.039502

Тест показує на незначущість нульової гіпотези - залишки постійні (рис. 10.9).

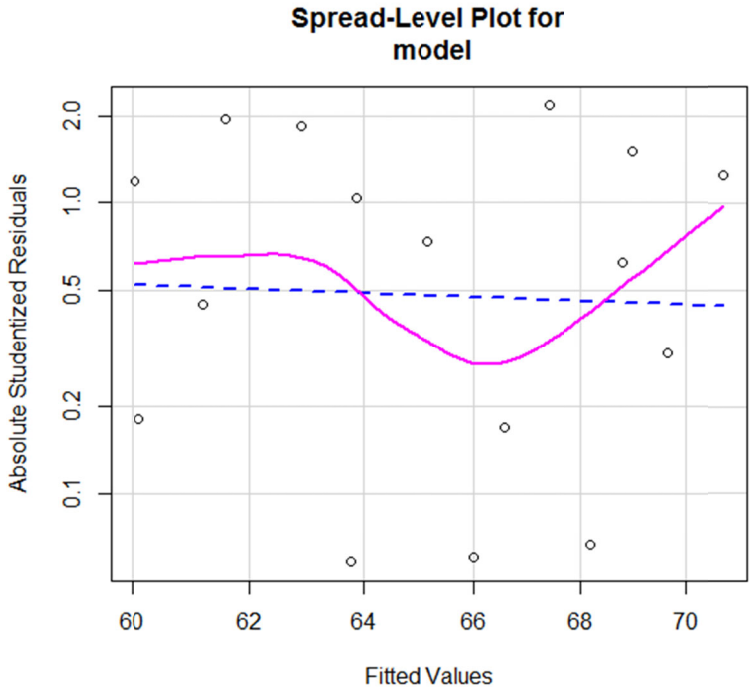

Рисунок 10.9 – Тест на гомоскедастичність

Глобальна оцінка припущень:

library(gvlma)
summary(gvlma(model))
Call:
lm(formula = Employed ~ ., data = longley)
Residuals:
Min 1Q Median 3Q Max
-0.41011 -0.15767 -0.02816 0.10155 0.45539
Coefficients:
Estimate Std. Error t value Pr(>|t|)
*(Intercept) -3.482e+03 8.904e+02 -3.911 0.003560 **
GNP.deflator 1.506e-02 8.492e-02 0.177 0.863141
GNP -3.582e-02 3.349e-02 -1.070 0.312681
Unemployed -2.020e-02 4.884e-03 -4.136 0.002535 **
Armed.Forces -1.033e-02 2.143e-03 -4.822 0.000944 ***
Population -5.110e-02 2.261e-01 -0.226 0.826212
Year 1.829e+00 4.555e-01 4.016 0.003037 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3049 on 9 degrees of freedom
Multiple R-squared: 0.9955, Adjusted R-squared: 0.9925
F-statistic: 330.3 on 6 and 9 DF, p-value: 4.984e-10
ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:*

Level of Significance = 0.05
Call:
gvlma(x = model)
Value p-value Decision
Global Stat 0.77371 0.9419 Assumptions acceptable.
Skewness 0.47036 0.4928 Assumptions acceptable.
Kurtosis 0.21377 0.6438 Assumptions acceptable.
Link Function 0.02492 0.8746 Assumptions acceptable.
Heteroscedasticity 0.06465 0.7993 Assumptions acceptable.
Мультіколінеарність:
sqrt(vif(model))>2
GNP.deflator GNP Unemployed Armed.Forces Population Year
TRUE TRUE TRUE FALSE TRUE TRUE

Тест показує наявність мультіколінеарності.
Проведіть тести для нової моделі.

САМОСТІЙНА РОБОТА

Самостійна робота студентів передбачає виконання лабораторних робіт та опрацювання ряду теоретичних питань із дисципліни «Data mining», які не увійшли до лекційного матеріалу курсу. У табл. А наведено теоретичні питання для самостійного опрацювання за кожною розгляненою темою.

Таблиця А – Питання для самостійного опрацювання

№ з/п	Назва теми	Питання
1)	Data Mining як мультидисциплінарна галузь	Етапи та методи знаходження нових знань
2)		Основні моделі інтелектуальних обчислювань
3)		Засоби програмної підтримки інтелектуального аналізу даних
4)	Набір даних та їх атрибутів	Новітні напрямки застосування Data Mining
5)		Концепція нечітких обчислень
6)		Нечітка логіка в системах Data Mining
7)	Методи та стадії Data Mining	Класичні технології класифікації в Data Mining
8)		Програмні засоби реалізації нейрокомп’ютерних технологій
9)	Завдання Data Mining. Інформація та знання	Практичний аспект застосування технології асоціативних правил
10)	Метод пошуку асоціативних правил	Програмні засоби пошуку асоціативних правил
11)		Основні поняття теорії асоціативних правил
12)	Метод кластерного аналізу	Основні положення теорії генетичних алгоритмів
13)		Програмне забезпечення та сфері застосування генетичних алгоритмів
14)		Концептуальні засади еволюційної теорії
15)	Метод дерева рішень	Дерева рішень – загальні принципи технології
16)		Моделі генетичних алгоритмів
17)		Комп’ютерні системи та напрямки застосування дерев рішень

Таблиця А (продовження)

№ з/п	Назва теми	Питання
18)	Метод штучних нейронних мереж	Мурашині алгоритми та генетичне програмування
19)		Поняття та можливості нейрокомп’ютерних технологій
20)		Сучасна практика та перспективні напрямки застосування нейротехнологій
21)		Програмне забезпечення нечітких методів
22)	Метод дискримінантного аналізу	Сучасна практика застосування нечітких методів
23)		Архітектура нейронних мереж
24)	Метод аналізу часових рядів	Програмне забезпечення задач класифікації
25)		Програмне забезпечення задач кластеризації

ПІДСУМКОВИЙ КОНТРОЛЬ

Підсумковий контроль проходить у вигляді заліку.

Залік складається з двох завдань: теоретичне завдання та практичне завдання.

Теоретичне завдання представлене у вигляді тестування за змістовими модулями у системі Moodle та має вигляд контрольного тестування за вивченим матеріалом курсу, яке складається з 10 питань. За кожну правильну відповідь студент отримує 2 бали.

Питання до заліку:

1. Зміст терміна «Data Mining».
2. Поняття статистики. Поняття машинного навчання. Поняття штучного інтелекту.
3. Розвиток технології баз даних.
4. Data Mining як частина ринку інформаційних технологій: класифікація аналітичних систем, думки експертів про Data Mining.
5. Проблеми технології Data Mining.
6. Відмінності Data Mining від інших методів аналізу даних. Існуючі підходи до аналізу.
7. Поняття про дані у широкому розумінні. Набір даних та їх атрибутів.
8. Змінна. Значення. Генеральна сукупність. Вибірка. Параметри. Статистики. Гіпотези.
9. Шкали: номінальна, порядкова, інтервальна, відносна, діхотомічна.
10. Типи наборів даних.
11. Табличні дані. Графічні дані. Формати зберігання даних.
12. Основні положення баз даних.
13. Системи управління базами даних (СУБД). Вимоги до СУБД.
14. Класифікація видів даних. Метадані.
15. Класифікація стадій Data Mining.
16. Класифікація методів Data Mining.
17. Статистичні методи Data Mining.
18. Кібернетичні методи Data Mining.
19. Властивості методів Data Mining.
20. Завдання Data Mining: класифікація, кластеризація, асоціація, послідовна асоціація, прогнозування, оцінювання, аналіз зв’язків, візуалізація.
21. Класифікація завдань Data Mining: за стратегіями: навчання з вчителем, навчання без вчителя, інші.
22. Класифікація завдань Data Mining в залежності від моделей, що використовуються: дескриптивні, прогнозуючі.
23. Розподіл завдань Data Mining: автоматичне дослідження та відкриття (вільний пошук), пояснення та опис, зв’язок понять.
24. Поток «від даних до рішень». Поток «від завдання до додатку».
25. Інформація. Властивості інформації. Знання та їх властивості.
26. Сфери застосування асоціативних правил.
27. Завдання аналізу ринкового кошика. Транзакційна база даних.

28. Визначення та характеристики асоціативних правил.
29. Межі підтримки та достовірності асоціативного правила.
30. Алгоритм Apriori та його різновиди.
31. Програмні засоби реалізації методу пошуку асоціативних правил.
32. Реалізація методу пошуку асоціативних правил у пакеті arules середовища R.
33. Поняття про кластерний аналіз.
34. Завдання кластерного аналізу.
35. Методи кластерного аналізу: ієрархічні та неієрархічні.
36. Міри подібності: квадрат евклідової відстані, Манхеттінська відстань, відстань Чебишева, відсоток незгоди.
37. Методи об’єднання або зв’язку.
38. Алгоритм k-середніх.
39. Перевірка якості кластеризації.
40. Застосування методу дерева рішень для завдань класифікації та прогнозування.
41. Переваги дерев рішень.
42. Процес конструювання дерева рішень.
43. Критерій розщеплення. Зупинка побудови дерева.
44. Скорочення дерева або відсікання гілок.
45. Алгоритми, що реалізують дерева рішень.
46. Завдання Data Mining, що вирішуються з допомогою штучних нейронних мереж.
47. Елементи нейронних мереж. Архітектура нейронних мереж.
48. Навчання нейронних мереж.
49. Моделі нейронних мереж: одношаровий та багатошаровий персептрон.
50. Програмне забезпеення для роботи з нейронними мережами.
51. Карти Кохонена, що самоорганізуються.
52. Проблеми дискримінантного аналізу.
53. Дискримінантний факторний аналіз.
54. Геометричний прогнозний дискримінантний аналіз.
55. Ймовірнісний дискримінантний аналіз.
56. Вимірювання якості моделі: лямбда Уїлкса, коефіцієнт детермінації, скорегований коефіцієнт детермінації.
57. Дискримінантний аналіз якісних змінних (метод DISQAL).
58. Переваги та недоліки дискримінантного аналізу.
59. Послідовність етапів регресійного аналізу.
60. Завдання регресійного аналізу.
61. Тренд, сезонність та цикл.
62. Точність прогнозу часових рядів.
63. Види прогнозів часових рядів.
64. Методи прогнозування часових рядів.
65. Однофакторна лінійна регресія в середовищі R.

66. Множинна лінійна регресія у середовищі R.

Практичне завдання приймається як захист індивідуального практичного завдання залікової роботи у системі Moodle. Воно полягає в розв’язанні двох аналітичних завдань на проведення аналізу даних на мові R, за правильне виконання кожного з яких студент отримує 10 балів. Результат виконання завдань оцінюється за такою шкалою:

- 10 балів – завдання повністю виконано без помилок;
- 9 балів – студент в цілому виконав практичне завдання, але не повно та допустивши деякі неточності;
- 8 балів – студент правильно визначив сутність практичного завдання, але виконав його частково й допустив при цьому одну помилку, що не впливає на загальне розуміння практичного завдання;
- 7 балів – студент правильно визначив сутність практичного завдання, але виконав його частково й допустив при цьому дві помилки, що не впливають на загальне розуміння практичного завдання;
- 6 балів – студент правильно визначив сутність практичного завдання, але виконав його частково й допустив при цьому три помилки, що не впливають на загальне розуміння практичного завдання;
- 5 балів – студент правильно визначив сутність практичного завдання, але виконав його недостатньо або поверхово, допустивши при цьому одну помилку, що впливає на загальне розуміння практичного завдання;
- 4 бали – студент правильно визначив сутність практичного завдання, але виконав його недостатньо або поверхово, допустивши при цьому дві помилки, що впливають на загальне розуміння практичного завдання;
- 3 бали – студент правильно визначив сутність практичного завдання, але виконав його недостатньо або поверхово, допустивши при цьому три помилки, що впливають на загальне розуміння практичного завдання;
- 2 бали – студент частково або поверхово виконав практичне завдання, допустивши при цьому одну помилку, що суттєво впливає на загальне розуміння практичного завдання;
- 1 бал – студент частково або поверхово виконав практичне завдання, допустивши при цьому дві помилки, що суттєво впливають на загальне розуміння практичного завдання;
- 0 балів – студент не виконав практичне завдання.

Усього за підсумковий семестровий контроль студент може отримати 40 балів.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Черняк О. І., Захарченко П. В. Інтелектуальний аналіз даних : підручник. Київ : Знання, 2014. 599 с.

2. Ситник В. Ф., Краснюк М. Т. Інтелектуальний аналіз даних (дейтамайнінг) : навч. посіб. Київ : КНЕУ, 2007. 376 с.

3. Олійник А. О., Субботін С. О., Олійник О. О. Інтелектуальний аналіз даних : навч. посіб. Запоріжжя : ЗНТУ, 2012. 278 с.

4. Марченко О. О., Россада Т. В. Актуальні проблеми Data Mining : навч. посіб. Київ : КНУ ім. Т. Шевченка, 2017. 150 с.

5. Ланде Д. В., Субач І. Ю., Бояринова Ю. Є. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки: навч. посіб. Київ : ІСЗЗІ КПІ ім. Ігоря Сікорського, 2018. 297 с.

6. Чорноус Г., Фаренюк Я., Діденко І. Дата майнінг для економістів: навч. посіб. Київ : Видавництво Ліра-К, 2023. 290 с.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Ситник В. Ф., Краснюк М. Т. Інтелектуальний аналіз даних (дейтамайнінг) : навч. посіб. Київ : КНЕУ, 2007. 376 с.

2. Черняк О. І., Захарченко П. В. Інтелектуальний аналіз даних : підруч. Київ : Знання, 2014. 599 с.

3. Олійник А. О., Субботін С. О., Олійник О. О. Інтелектуальний аналіз даних : навч. посіб. Запоріжжя : ЗНТУ, 2012. 278 с.

4. Гладун А. Я., Рогушина Ю. В. Data Mining: пошук знань в даних. Київ : ТОВ «ВД «АДЕФ-Україна», 2016. 452 с.

5. Додонов О. Г., Кузмичов А. І. Датамайнінг в Excel. Розвідувальний аналіз даних та прогнозування з використанням надбудови Analytic Solve Data Mining. Київ : Видавництво Ліра-К, 2023. 240 с.

6. Провост Фостер, Фоусетт Том Data Science для бізнесу: Як збирати, аналізувати і використовувати дані / пер. з англ. А. Дудченко. Київ : Наш формат, 2019. 400 с.

Додаткова:

1. Ciaburro Giuseppe, Venkateswaran Balaji. Neural Networks with R. Birmingham : Packt Publishing, 2017. 314 p.

2. Berry M.J.A., Linoff G. Data mining techniques: for marketing, sales, and customer relationship management. Indianapolis : Wiley Publishing, 2004. 672 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi4/0006100.pdf.

3. Pyle Dorian. Business modeling and data mining. Burlington : Morgan Kaufmann Publishers, 2003. 650 p.

4. Yanchang Z., Yonghua C. Data Mining Applications with R. Waltham, Oxford, Amsterdam : Elsevier, 2014. 471 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Kudin/0036204.pdf.

5. Azzalini A., Bruno S. Data Analysis and Data Mining. An Introduction. New York : Oxford University Press, 2012. 289 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Kudin/0036206.pdf.

6. Gisele L. P., Alex A. F. Automating the Design of Data Mining Algorithms: an Evolutionary Computation Approach. Heidelberg : Springer-Verlag Berlin Heidelberg, 2010. 197 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Kudin/0036216.pdf.

7. Stephane T. Data Mining and Statistics for Decision Making. New York : John Wiley & Sons, 2011. 704 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Kudin/0036219.pdf.

8. Плескач В. Л., Затонацька Т. Г. Інтелектуальні технології Data Mining і Text Mining. Інформаційні системи і технології на підприємствах. Київ : Знання, 2011. С. 540–559.

9. Kandethody M. Ramachandran, Chris P. Tsokos Mathamatical Statistics With Applications in R. London, San Diego, Cambridge, Oxford : Esevier, 2021. 680 p. URL: https://doi.org/10.1016/C2018-0-02285-9.

10. Paolo Giordani, Maria Brigida Ferraro, Francesca Martella. An Introduction to Clustering with R. Singapore : Springer Singapore, 2020. 340 p. DOI: https://doi.org/10.1007/978-981-13-0553-5.

11. Laura Chihara, Tim Hesterberg. Mathematical Statistics with Resampling and R.Hoboken, New Jersey : Wiley, 2011. 434 p.

12. Peter Dalgaard. Introductory Statistics with R. Second Edition. New York : Springer, 2008. 370 p. URL: https://www.academia.dk/BiologiskAntropologi/Epidemiologi/PDF/Introductory_Statistics_with_R__2nd_ed.pdf.

13. Danielle Navarro. Learning statistics with R: A tutorial for psychology students and other beginners (Version 0.6). University of New South Wales, 613 p. URL: http://compcogscisydney.org/learning-statistics-with-r.

14. Vijay Kotu and Bala Deshpande. Data Science. Concept and Practice. Second Edition. Cambridge : Elsevier, 2019. 549 p. URL: https://asolanki.co.in/wp-content/uploads/2019/04/DataScience-Concepts-and-Practice-2nd-Edition-3.pdf.

Навчально-методичне видання
(*українською мовою*)

Іванов Сергій Миколайович
Очеретін Дмитро Валерійович

DATA MINING

Навчально-методичний посібник для здобувачів ступеня вищої освіти магістра спеціальності «Економіка» освітньо-професійної програми «Економічна кібернетика»

Рецензент *М.М .Іванов*
Відповідальний за випуск *Н.К. Максишко*
Коректор *В.В. Рянічева*