Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель і методи процесу автоматизації кредитування

(тема)

Виконав:

студент II курсу, групи СПм-22-3

Балінський Д. І.
(прізвище, ініціали)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: проф. Горбачов В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Балінському Дмитру Ігоровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Модель системи керування розподілом електричної енергії з використанням машинного навчання

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи

3.1 Методи кластеризації

3.2 Теорія штучного інтелекту,

3.3 Процес кредитування,

4. Перелік питань, що потрібно опрацювати у роботі

Аналіз мети і задач проекту

Аналіз банківського процесу надання кредиту

Аналіз моделей нейронних мереж

Оцінки платоспроможності клієнта на базі нейронної моделі,

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація – 16 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Цілі проекту	26.05.24-27.05.24	
2	Аналіз проблеми та огляд літератури	28.05.24-29.05.24	
3	Визначення проблеми	30.05.24-01.06.24	
4	Аналіз методів	31.05.24-01.06.24	
5	Реалізація завдання	02.06.24-07.06.24	
6	Програмна реалізація	07.06.24-9.06.24	
7	Підготовка кваліфікаційної роботи	10.06.24-11.06.24	
8	Підготовка презентації	12.06.24	

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Горбачов В.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 82 с., 19 рис., 13 табл., 2 дод., 33 джерел.

НЕЙРОННОЇ МЕРЕЖІ, КЛАСТЕРИЗАЦІЯ, КЛАСИФІКАЦІЯ, СЕГМЕНТАЦІЯ КЛІЄНТІВ.

Метою кваліфікаційної роботи є розробка системи визначення платоспроможності клієнтів банку на основі методу штучних нейронних мереж. Вхідні дані для цієї системи нормалізуються для роботи нейронних мереж.

У ході виконання кваліфікаційної роботи була розглянуто основні методи кластеризації та встановлено, що найбільш зручним є метод кластеризації на основі карти самоорганізації Кохонена.

ABSTRACT

Master’s thesis: 82 pages, 19 figures, 13 tables, 2 appendices, 33 sources.

NEURON NETWORK, CLUSTERING, CLASSIFICATION, SEGMENTATION.

The purpose of the qualification work is to develop a system for determining the solvency of bank clients based on the method of artificial neural networks. The input data for this system is normalized for neural networks.

In the course of the qualification work, the main methods of clustering were considered and it was established that the most convenient is the method of clustering based on the Kohonen self-organization map.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DM - Розробка даних (англ., Data Mining)

ABS -Автоматизована система банку (англ., Automated Bank Systems)

AID - Автоматичний детектор взаємодії (англ., Automatic detector of interaction)

AHP – Процес аналітичної ієрархії (англ. Analytic hierarchy process)

DBMS – Система управління базами даних (англ., Database management system)

ANN – Штучна нейронна мережа (англ. Artificial neural network)

CLARA – Clustering LARge Applications

ВСТУП

Платоспроможність відображає здатність підприємства (або фізичної особи) погасити свої фінансові зобов'язання. З цієї причини найшвидшою оцінкою платоспроможності клієнта (компанії) є її активи мінус зобов'язання, які дорівнюють акціонерному капіталу. Існують також коефіцієнти платоспроможності, які можуть виділити певні сфери платоспроможності для глибшого аналізу.

Певні події можуть створити підвищений ризик для платоспроможності навіть для добре відомих компаній. У випадку підприємницької діяльності очікуване закінчення терміну дії патенту може становити ризики для платоспроможності, оскільки це дозволить конкурентам виробляти відповідний продукт, і це призводить до втрати відповідних виплат роялті. Крім того, зміни в певних нормативних актах, які безпосередньо впливають на здатність компанії продовжувати бізнес, можуть становити додатковий ризик. Як підприємства, так і приватні особи також можуть зіткнутися з проблемами платоспроможності, якщо проти них буде винесено велике судове рішення після судового позову.

Проблема визначення платоспроможності клієнта розбивається на дві задачі. Перша задача – кластеризація клієнтів на основі інформації про наявних клієнтів за кількістю класів та визначення їх характеристик. Друга задача – це класифікація нового клієнта для виявлення класу приналежності клієнта та прийняття рішення про можливість кредитування.

1 ОГЛЯД ПРОБЛЕМИ ТА МЕТА РОБОТИ

1.1 Огляд штучної нейронної мережі

Штучна нейронна мережа – це парадигма обробки інформації, яка працює подібно до біологічної нервової системи. Він складається з великої кількості взаємопов’язаних елементів обробки, які працюють паралельно для вирішення конкретних завдань.

Нейронні мережі обробляють інформацію подібно до людського мозку. Вони формуються спеціальним шляхом через навчання з вхідних даних. Навчання передбачає регуляцію зв’язків між штучними нейронами.

Нейронні мережі використовуються для виявлення тенденцій і вилучення шаблонів, які майже неможливо завершити ні людиною, ні іншими комп’ютерними техніками, але легко зробити ШНМ через їх чудову здатність отримувати значення зі складних або неточних даних. Навчену нейронну мережу можна розглядати як «експерта» в галузі, яку їй було надано для аналізу.

Основні переваги штучних нейронних мереж.

Самоорганізація. ШНМ може створювати власне представлення інформації, отриманої під час навчання.

Адаптивне навчання. Це означає, що нейронні мережі можуть навчитися виконувати завдання на основі вхідних даних, наданих для навчання або початкового досвіду.

Робота в реальному часі. Усі елементи обробки можуть працювати, і розробляються та виготовляються спеціальні апаратні пристрої, які використовують цю можливість.

У наш час комп’ютери дотримуються набору інструкцій під час вирішення існуючої проблеми, тобто використовується алгоритмічний підхід. Це означає, що здатність комп’ютерів знаходити рішення обмежена

проблемою, яку людина вже знає та розуміє, як вирішити. Але якби комп’ютери могли працювати з проблемами, з якими люди точно не знають, як справлятися, вони були б набагато кориснішими.

З іншого боку, комп’ютери базуються на когнітивному підході до розв’язання проблем: формулювання проблеми та її рішення мають бути відомі та виражені чітко й недвозначно, а потім усі кроки перетворюються на мовну програму високого рівня та машинний код, який комп’ютер може зрозуміти. Отже, якщо щось піде не так, це пов’язано з програмним чи апаратним збоєм. Ці машини цілком передбачувані.

Нейронні мережі використовують інший підхід до вирішення проблем; їх обчислення можуть проводитися паралельно. Нейронні мережі можуть працювати з даними, що зазвичай вимагає людського мислення та здатності робити зауваження. Мережу неможливо запрограмувати на виконання конкретного завдання. Він вчиться на прикладі, який потрібно ретельно вибирати, інакше буде втрачено корисний час або, що ще гірше, мережа може працювати некоректно. Зворотним боком унікальності штучних нейронних мереж є такий недолік, що їх робота може бути непередбачуваною, оскільки мережа сама знаходить, як вирішити проблему.

Нейронні мережі та алгоритмічні комп'ютери доповнюють один одного. Вони не змагаються, тому що є завдання, які більше підходять для алгоритмічного підходу, наприклад математичні операції, і завдання, які більше підходять для нейронних мереж. І в той час для виконання багатьох завдань з максимальною ефективністю потрібні системи, які використовують комбінацію двох підходів (зазвичай комп’ютер використовується для контролю нейронної мережі).

У нейронних мереж є багато хороших моментів, і прогрес у цій галузі підвищить їх популярність. Вони чудово підходять для розпізнавання візерунків і можуть бути використані там, де традиційні методи не працюють. Нейронні мережі також широко використовуються для прогнозування та апроксимації, класифікації та кластеризації, ідентифікації

та навіть контролю даних.

У цій кваліфікаційній роботі буде описано застосування ШНМ у прийнятті рішень.

Центральне місце в інтелектуальній діяльності займає проблема формалізації процесів прийняття рішень, оскільки це обов'язковий і часто повторюваний етап будь-якої діяльності. За статистичними даними людина протягом дня приймає від двох до трьох тисяч рішень різної складності, починаючи від простих побутових рішень і закінчуючи складними професійними рішеннями. Тому синтез універсальної, інваріантної до предметної області, формальної математичної моделі прийняття рішень представляє великий теоретичний і практичний інтерес.

Структуризація процедури прийняття рішень дозволяє виділити чотири загальні етапи:

- визначення мети;
- формування множини можливих шляхів її досягнення (множини можливих рішень);
- формування оцінки, що дозволяє оцінити рішення на множині (оціночні завдання);
- визначення оптимального рішення з безлічі можливих рішень (задача оптимізації).

З перерахованих вище етапів завдання оцінки є базовим. Його реалізація вимагає формалізації лінгвістичних змінних, таких як «найкращий», «переважний», «ефективний» варіант рішення.

Це пов'язано з виділенням деяких часткових якостей системи, які адекватно описують цю систему. Потім необхідно визначити певну метрику на множині лінгвістичних змінних. За цією метрикою реалізовано порівняння якості рішень. Ця задача відома як задача багатофакторного оцінювання.

Складність вирішення даної задачі зумовлена багатовимірністю наступних фактів: факторного простору та неоднорідності розмірності, інтервалу можливих значень, шкал вимірювання та неузгодженості

локальних характеристик, що описують якість рішення. Додаткова складність пов'язана з принциповою суб'єктивністю поняття «найкраще рішення». Таким чином, виникає проблема синтезу формальної багатофакторної моделі оцінювання, найбільш адекватної кожній конкретній ситуації прийняття рішення.

Новим ефективним напрямком є застосування штучної нейронної мережі в процесі прийняття рішень. Використання нейронних мереж дозволить зберігати експериментальні знання, узагальнювати їх і робити доступними для користувача в зручній для інтерпретації та прийняття рішень формі.

1.2 Огляд підтримки прийняття рішень

Підтримка прийняття рішень виникла на початку ери розподілених обчислень. Історія таких систем починається приблизно в 1965 році, і важливо почати формалізувати записи ідей, людей, систем і технологій, задіяних у цій важливій сфері прикладних інформаційних технологій. Сьогодні все ще можливо відновити історію систем підтримки прийняття рішень за спогадами з перших рук і неопублікованими матеріалами, а також за опублікованими статтями.

До 1965 року створення великомасштабних інформаційних систем було дуже дорогим. Приблизно в цей час розробка IBM System 360 та інших потужніших систем мейнфреймів зробила більш практичним і економічно ефективним розробку інформаційних систем управління у великих компаніях. MIS зосереджена на забезпеченні менеджерів структурованими періодичними звітами. Значна частина інформації надходила з систем бухгалтерського обліку та операцій.

Наприкінці 1960-х років на практиці став новий тип інформаційних систем – модельно-орієнтовані СППР або системи управлінських рішень. Двоє піонерів DSS, Пітер Кін і Чарльз Стабелл, стверджують, що концепція

підтримки прийняття рішень виникла в результаті «теоретичних досліджень організаційного прийняття рішень, проведених в Технологічному інституті Карнегі в кінці 1950-х і на початку 60-х років, і технічної роботи над інтерактивними комп'ютерними системами. , в основному проведений в Массачусетському технологічному інституті в 1960-х роках».

Згідно зі Спрагом і Уотсоном, приблизно в 1970 році ділові журнали почали публікувати статті про системи прийняття управлінських рішень, системи стратегічного планування та системи підтримки прийняття рішень. Наприклад, Скотт Мортон і його колеги опублікували ряд статей про підтримку прийняття рішень у 1968 році. У 1969 році Фергюсон і Джонс обговорювали автоматизовану систему прийняття рішень у журналі Management Science. У 1971 році була опублікована новаторська книга Майкла С. Скотта Мортона «Системи управлінських рішень: комп'ютерна підтримка прийняття рішень». У 1966-67 роках Скотт Мортон досліджував, як комп'ютери та аналітичні моделі можуть допомогти менеджерам прийняти ключове рішення. Він провів експеримент, у якому менеджери фактично використовували систему управлінських рішень. Менеджери з маркетингу та виробництва використовували MDS для координації планування виробництва обладнання для пралень. MDS працював на 21-дюймовому ЕПТ IDI зі світловим пером, підключеним за допомогою модему 2400 біт/с до пари систем Univac 494. Дисертаційне дослідження Скотта Мортона було новаторським впровадженням, визначенням і дослідницьким тестуванням керованої моделлю системи підтримки прийняття рішень.

Т.П. Герріті-молодший зосередився на питаннях проектування систем підтримки прийняття рішень у своїй статті Sloan Management Review 1971 року під назвою «Розробка систем прийняття рішень «Людина-машина: застосування до управління портфелем». Його система була розроблена для підтримки інвестиційних менеджерів у щоденному управлінні портфелем акцій клієнтів. СППР для управління портфелем стали дуже складними з тих пір, як Герріті почав свої дослідження.

У 1974 році Гордон Девіс, професор Університету Міннесоти, опублікував свій впливовий текст про інформаційні системи управління. Він визначив інформаційну систему управління як «інтегровану систему людина/машина для надання інформації для підтримки операцій, управління та функцій прийняття рішень в організації».

До 1975 року Дж. Д. К. Літтл розширював межі комп'ютерного моделювання. DSS Little під назвою Brandaid був розроблений для підтримки продуктів, просування, цін і рекламних рішень. Крім того, Літтл у попередній статті визначив критерії для розробки моделей і систем для підтримки прийняття управлінських рішень. Його чотири критерії включали: надійність, легкість керування, простоту та повноту відповідних деталей. Усі чотири критерії залишаються актуальними для оцінки сучасних систем підтримки прийняття рішень.

Кляйн і Метлі відзначають: «Дослідження походження СППР ще належить написати. Здається, перші статті СППР були опубліковані аспірантами або професорами бізнес-шкіл, які мали доступ до першої комп'ютерної системи розподілу часу: Project MAC. у школі Слоуна, системи розподілу часу в Дартмуті в школі Так. У Франції HEC була першою французькою бізнес-школою, яка мала систему розподілу часу (встановлена в 1967 році), і перші статті DSS були опубліковані професорами школи. у 1970 році. Термін SIAD і концепція DSS були розроблені незалежно у Франції, у кількох статтях професорів HEC, які працюють над проектом SCARABEE, який розпочався у 1969 році та завершився у 1974 році».

Наприкінці 1970-х років як практичні, так і теоретичні питання, пов'язані з DSS, обговорювалися на наукових конференціях, включаючи зустрічі Американського інституту наук про прийняття рішень та конференцію ACM SIGBDP із систем підтримки прийняття рішень у Сан-Хосе, штат Каліфорнія, у січні 1977 року. Перша міжнародна конференція з питань прийняття рішень Support Systems відбувся в Атланті, штат Джорджія, у 1981 році. Наукові конференції забезпечили форуми для обміну

ідеями, теоретичних дискусій та обміну інформацією. Дослідники MIT, зокрема Пітер Кін і Майкл Скотт Мортон, були особливо впливовими.

У 1980 році Стівен Алтер опублікував результати своєї докторської дисертації Массачусетського технологічного інституту у впливовій книзі під назвою «Системи підтримки прийняття рішень: поточна практика та триваючі виклики». Дослідження та статті Альтера розширили рамки наших думок про управління СППР. Крім того, його тематичні дослідження забезпечили міцну описову основу прикладів системи підтримки прийняття рішень. Ряд інших дисертацій Массачусетського технологічного інституту, завершених у середині та наприкінці 1970-х років, також стосувалися питань, пов’язаних із використанням моделей для підтримки прийняття рішень.

У 1979 році Джон Рокарт з Гарвардської бізнес-школи опублікував новаторську статтю в Harvard Business Review, яка призвела до розробки інформаційних систем керівників або систем підтримки керівників.

Бончек, Холсаппл і Вінстон створили теоретичну основу для розуміння проблем, пов’язаних із проектуванням систем підтримки прийняття рішень, орієнтованих на знання. Їхня книга показала, наскільки технології штучного інтелекту та експертних систем мають відношення до розробки СППР.

Книга Ральфа Спраг та Еріка Карлсона «Створення ефективних систем підтримки прийняття рішень» стала важливою віхою. Далі пояснюється структура Sprague DSS бази даних, бази моделей і програмного забезпечення для створення та керування діалогами. Крім того, він надав практичний, зрозумілий огляд того, як організації можуть і повинні створювати СППР. Хоча їхня книга, ймовірно, створила деякі нереалістичні очікування, проблеми випливали більше з обмежень існуючих технологій для побудови СППР, а не з обмежень концепцій, які обговорювали Спраг і Карлсон.

До кінця 1970-х років ряд дослідників і компаній розробили інтерактивні інформаційні системи, які використовували дані та моделі, щоб допомогти менеджерам аналізувати напівструктуровані проблеми. Усі ці різноманітні системи називалися системами підтримки прийняття рішень. З

тих перших днів було визнано, що DSS може бути розроблена для підтримки осіб, які приймають рішення на будь-якому рівні організації. DSS може підтримувати операції, фінансовий менеджмент і прийняття стратегічних рішень. DSS може використовувати просторові дані в таких системах, як система аналізу геоданих і відображення, структуровані багатовимірні дані та неструктуровані документи. У DSS використовувалися різні моделі, включаючи оптимізацію та моделювання. Також статистичні пакети були визнані інструментами для побудови СППР. Дослідники штучного інтелекту почали працювати над експертними системами управління та бізнесу на початку 1980-х років.

Системи фінансового планування стали популярними інструментами підтримки прийняття рішень. Ідея полягала в тому, щоб створити «мову», яка «дозволить керівникам будувати моделі без посередників». Популярна система фінансового планування під назвою IFPS, абревіатура від інтерактивної системи фінансового планування, була спочатку розроблена наприкінці 1970-х років Джеральдом Р. Вагнером та його студентами з Техаського університету. Компанія Вагнера, EXECUCOM Systems, продавала IFPS до середини 1990-х років. Однією з головних переваг мови планування над електронною таблицею є те, що модель написана з використанням природної мови, і модель може бути відокремлена від даних. На початку 80-х років електронні таблиці також використовувалися для побудови DSS на основі моделі. У статті 1988 року Шарда, Барр і МакДоннелл зробили огляд перших 15 років досліджень DSS. Дослідження, пов’язані з використанням моделей і систем фінансового планування для підтримки прийняття рішень, були обнадійливими, але, звичайно, не були однозначно позитивними.

На початку 1980-х років академічні дослідники розробили нову категорію програмного забезпечення для підтримки групового прийняття рішень. Mindsight від Execucom Systems, GroupSystems, розроблені в Університеті Арізони, і система SAMM, розроблена дослідниками Університету Міннесоти, були ранніми групами DSS. Діксон, Пул і

ДеСенктіс повідомляють, що Брент Геллап, доктор філософії. студент Міннесоти, у 1984 році вирішив «запрограмувати свою власну невелику систему GDSS на BASIC і запустити її на комп'ютері VAX свого університету». Ця система стала початком досліджень GDSS Міннесоти.

Джей Нунамейкер-молодший та його колеги в 1992 році писали, що «основна концепція GroupSystems почалася в 1965 році з розробки мови постановки проблем/аналізатора постановки проблем як частини проекту ISDOS (система проектування та оптимізації інформаційних систем) у компанії Case. технологічний інститут». У 1984 році було завершено створення системи під назвою PLEXSYS і в Університеті Арізони було побудовано комп'ютерне приміщення для групових зустрічей. Перше приміщення під назвою PlexCenter містило великий U-подібний конференц-стіл із 16 комп'ютерними робочими станціями. Компанія PLEXSYS стала основою для розробки програмного забезпечення GroupSystems університету Арізони. З середини 80-х років багато досліджень вивчали вплив і наслідки Group DSS. Крім того, низка компаній комерціалізували Group DSS і групове програмне забезпечення. Клацніть тут, щоб побачити кімнату підтримки групового прийняття рішень.

Керівницькі інформаційні системи розвинулися з однокористувацьких систем підтримки прийняття рішень і вдосконалених продуктів реляційної бази даних. Перший EIS використовував попередньо визначені інформаційні екрани та підтримувався аналітиками для вищого керівництва. Наприклад, восени 1978 року в компанії Lockheed-Georgia почалася розробка EIS під назвою «Система підтримки управління інформацією та прийняття рішень». Починаючи приблизно з 1990 року, сховища даних і он-лайн аналітична обробка почали розширювати сферу EIS і визначили більш широку категорію DSS, керованих даними. Найджел Пендсе стверджує, що першим продуктом Executive Information System був Command Center від Pilot Software. Він зазначає, що і багатовимірний аналіз, і OLAP походять від мови програмування APL і таких систем, як Express і Comshare's System W.

Найджел Пендсе з OLAPReport.com написав і оновлює набагато детальнішу історію походження продуктів OLAP.

Найлунд простежує події, пов'язані з бізнес-аналітикою, до зусиль Procter & Gamble у 1985 році створити DSS, який пов'язує інформацію про продажі та дані роздрібного сканера. Metaphor Computer Systems, допоміжний продукт дослідників з Дослідницького центру Xerox у Пало-Альто, розробив першу P&G DSS. Останні випускники Metaphor заснували багато постачальників BI: Річард Танлер заснував Information Advantage, а Кетрін Глассі стала співзасновником Brio Technologies. Термін BI – це популяризований загальний термін, який нібито ввів Говард Дреснер з Gartner Group у 1989 році. BI описує набір концепцій і методів для покращення прийняття бізнес-рішень за допомогою систем підтримки, заснованих на фактах. BI іноді використовується як взаємозамінний з довідками, інструментами для звітів і запитів та інформаційними системами для керівників. Системи бізнес-аналітики є DSS, керованими даними.

Починаючи приблизно з 1990 року, Білл Інмон і Ральф Кімбол активно пропагували DSS, створені з використанням технологій реляційних баз даних. Для багатьох практиків MIS DSS, створена за допомогою Oracle або DB2, була єдиною системою підтримки прийняття рішень, з якою вони познайомилися в популярній комп'ютерній літературі. СППР, керовані моделлю, входили до сфери дослідження операцій і не були частиною інформаційних систем. Ральф Кімбол був «доктором DSS», а Білл Інмон – «батьком сховища даних». Інмон визначив систему підтримки прийняття рішень як «систему, яка використовується для підтримки управлінських рішень. Зазвичай DSS передбачає аналіз багатьох одиниць даних евристичним способом. Як правило, обробка DSS не передбачає оновлення даних». Інмон і Кімбол зосередилися на створенні DSS на основі даних.

На початку 1990-х років відбувся значний технологічний зсув від DSS на основі мейнфреймів до DSS на основі клієнт/сервер. Деякі настільні інструменти OLAP були представлені в цей період часу. У 1992-93 роках

деякі постачальники почали рекомендувати об'єктно-орієнтовану технологію для побудови «повторно використовуваних» можливостей підтримки прийняття рішень. У 1994 році багато компаній почали модернізувати свою мережеву інфраструктуру. Постачальники СУБД визнали, що підтримка прийняття рішень відрізняється від OLTP, і почали впроваджувати реальні можливості OLAP у свої бази даних. Пол Грей стверджує, що приблизно в 1993 році сховище даних і співробітники EIS знайшли одне одного, і ці дві ніші технології зближуються. У 1995 році сховища даних і Всесвітня павутина почали впливати на практиків і науковців, які цікавляться технологіями підтримки прийняття рішень. Веб-інтерфейс DSS став можливим приблизно в 1995 році.

Історія систем підтримки прийняття рішень охоплює відносно короткий проміжок років, а концепції та технології все ще розвиваються. Сьогодні все ще можливо реконструювати історію систем підтримки прийняття рішень за ретроспективними звітами ключових учасників, а також за опублікованими та неопублікованими матеріалами. Багато перших інноваторів і перших розробників виходять на пенсію, але їхні ідеї та дії можна використати, щоб керувати майбутніми інноваціями в цій галузі. Є надія, що ця стаття допоможе нам зрозуміти «справжню» історію DSS. Інтернет і Інтернет прискорили розвиток підтримки прийняття рішень і забезпечили нові засоби фіксації та документування розвитку знань у цій галузі досліджень. Піонери підтримки прийняття рішень включають багатьох академічних дослідників з програм Массачусетського технологічного інституту, Університету Арізони, Гавайського університету, Університету Міннесоти та Університету Пердью. Піонери DSS створили особливі та чіткі потоки розвитку технологій і досліджень, які служать основою для більшої частини сьогоднішньої роботи в DSS.

1.3 Цілі роботи

Основною метою кваліфікаційної роботи є дослідження методів ідентифікації для багатофакторної оцінки з використанням штучних нейронних мереж. Робота має послідовні цілі, які описані нижче.

Перш за все, необхідно зробити огляд сучасних багатокритеріальних методів прийняття рішень та порівняти їх. Деякі методи можуть бути основою запропонованого методу.

По-друге, для кращих результатів прийняття рішень доцільно вибрати ефективні методи очищення даних. Зокрема, як для прийняття рішень, так і для нормалізації даних про продуктивність нейронної мережі. Необхідно вивчити та порівняти різні підходи до очищення даних.

Далі в якості однієї з головних ідей роботи вводиться попередня кластеризація. Необхідний загальний огляд основних підходів кластеризації. У даній кваліфікаційній роботі особлива увага зосереджена на методах кластеризації на основі моделі штучної нейронної мережі. Новинка полягає в підтримці методів прийняття рішень за допомогою нейронних мереж, які допомагають обробляти та аналізувати вихідні дані, зберігати отриману інформацію. Самоорганізована карта, яка лежить в основі методів кластеризації нейронної мережі, має складене правило навчання, алгоритм навчання. Тому ще одним важливим кроком є вивчення самоорганізованої організації карти та параметрів правила навчання.

Нарешті, пропонується реалізувати в демонстраційному програмному забезпеченні обґрунтовані підходи очищення даних, кластеризації даних та прийняття багатокритеріальних рішень. Впроваджуване програмне забезпечення повинно мати зручний і зрозумілий інтерфейс, підтримувати типи файлів, сумісні з більшістю систем обробки даних, наочно демонструвати результати процесу прийняття рішень і дозволяти зберігати інформацію для подальшого аналітичного дослідження.

2 АВТОМАТИЧНА ОБРОБКА ДАНИХ ПРО ПЛАТОСПРОМОЖНІСТЬ

Сучасні реальні бази даних дуже сприйнятливі до шумних відсутніх і неузгоджених даних через їх зазвичай величезний розмір, часто кілька гігабайт або більше. Існує кілька методів попередньої обробки даних. Очищення даних можна застосувати для усунення шуму та виправлення невідповідностей у даних. Інтеграція даних об’єднує дані з кількох джерел у єдине сховище даних, наприклад сховище даних або куб даних. Можна застосувати такі перетворення даних, як нормалізація. Наприклад, нормалізація може підвищити точність і ефективність алгоритмів видобутку, що включають вимірювання відстані. Зменшення даних може зменшити розмір даних шляхом агрегування, усунення зайвих функцій або, наприклад, кластеризації. Ці методи обробки даних, якщо їх застосувати до майнінгу, можуть значно покращити загальну якість видобутих шаблонів і/або час, необхідний для фактичного майнінгу.

2.1 Очищення та попередня обробка даних про платоспроможність

Очищення даних – це початковий етап уточнення набору даних, що робить його читабельним і придатним для використання за допомогою таких методів, як видалення дублікатів, обробка відсутніх значень і перетворення типів даних, тоді як попередня обробка даних подібна до отримання цих уточнених даних і масштабування за допомогою більш просунутих методів, таких як розробка функцій, .Дані реального світу, як правило, неповні, шумні та суперечливі. Процедури очищення даних намагаються заповнити пропущені значення, згладити шум під час виявлення викидів і виправити невідповідності в даних.

Відсутні значення. Існує кілька методів заповнення пропущених значень для атрибутів, які не мають записаних значень [14].

1 Ігноруйте кортеж. Зазвичай це робиться, коли відсутня мітка класу (припускаючи, що завдання видобутку включає класифікацію або опис). Цей метод не дуже ефективний, якщо тільки кортеж не містить кількох атрибутів із відсутніми значеннями. Це особливо погано, коли відсоток відсутніх значень на атрибут значно змінюється;

2 Вручну введіть пропущене значення. Загалом цей підхід займає багато часу та може бути неможливим, оскільки великий набір даних із багатьма відсутніми значеннями;

3 Використовуйте глобальну константу, щоб заповнити пропущене значення. Замініть усі відсутні значення атрибутів тією самою константою, наприклад міткою на зразок "Невідомо" або – . Але якщо пропущені значення замінено на «Невідомо», то програма видобутку може помилково подумати, що вони утворюють цікаву концепцію, оскільки всі вони мають спільне значення «Невідомо». Тому, незважаючи на те, що цей спосіб простий, його не рекомендується використовувати;

4 Використовуйте атрибут mean, щоб заповнити пропущене значення. Використовуйте це значення $50 000, наприклад, щоб замінити відсутнє значення доходу;

5 Використовуйте атрибут mean для всіх зразків, що належать до того ж класу, що й даний кортеж;

6 Використовуйте найбільш ймовірне значення, щоб заповнити пропущене значення. Це можна визначити за допомогою регресії, інструментів на основі висновків із використанням байєсівського формалізму або індукції дерева рішень.

Методи 3-6 зміщують дані. Заповнене значення може бути неправильним. Однак метод 6 є популярною стратегією. У порівнянні з іншими методами, він використовує найбільше інформації з поточних даних для прогнозування відсутніх значень.

Зашумлені дані. Шум – це випадкова помилка або дисперсія вимірюваної змінної. Існують наступні методи згладжування даних [16]:

Групування: методи групування згладжують відсортоване значення даних, перевіряючи його «околиці», тобто значення навколо нього. Відсортовані значення розподіляються в кілька «відер» або бункерів. Оскільки методи групування консультуються з околицями значень, вони виконують локальне згладжування. Під час згладжування за допомогою біну кожне значення в діапазоні замінюється середнім значенням діапазону: середнє значення 4, 8 і 15 у діапазоні 1 дорівнює 9. Тому кожне вихідне значення в цьому діапазоні замінюється на значення 9. Подібним чином можна використовувати згладжування медіанами бункеру, у якому кожне значення бункеру замінюється медіаною бункеру. Під час згладжування за межами бункеру мінімальне та максимальне значення в даному бункер визначаються як межі бункеру. Потім кожне значення бункеру замінюється найближчим граничним значенням. Загалом, чим більша ширина, тим більший ефект згладжування. Альтернативно, біни можуть бути рівноширинні, де інтервал значень у кожному бункері є постійним. Групування також використовується як метод дискретизації.

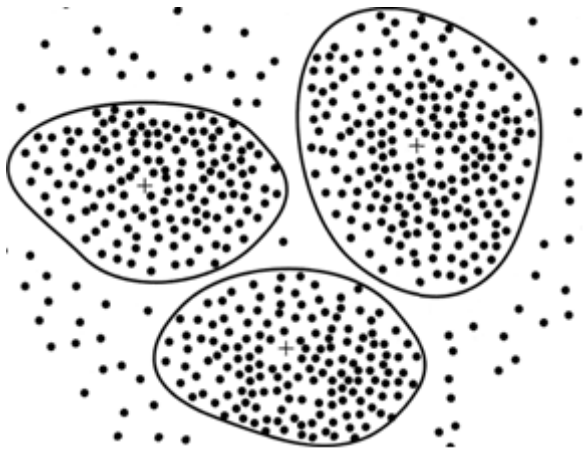

Рисунок 2.1 – Викиди, виявлені за допомогою аналізу кластеризації

Кластеризація: Викиди можуть бути виявлені шляхом кластеризації, коли подібні значення організовуються в групи або «кластери». Інтуїтивно зрозуміло, що значення, які виходять за межі набору кластерів, можуть

вважатися викидами (рисунок 2.1).

Комбінована перевірка комп’ютером і людиною: викиди можуть бути ідентифіковані за допомогою комбінації перевірки комп’ютером і людиною. В одній програмі, наприклад, інформаційно-теоретичний захід використовувався, щоб допомогти ідентифікувати викидні шаблони в базі даних рукописних символів для класифікації. Значення показника відображало «несподіваний» вміст передбачуваної мітки символу по відношенню до відомої мітки. Шаблони викидів можуть бути інформативними або "сміттєвими". Патерни, несподіваний вміст яких перевищує порогове значення, виводяться до списку. Потім людина може сортувати шаблони в списку, щоб визначити справжні сміттєві. Це набагато швидше, ніж шукати вручну всю базу даних. Потім шаблони сміття можна виключити з використання.

Регресія: дані можна згладити, підігнавши дані до функції, наприклад, за допомогою регресії. Лінійна регресія передбачає пошук «найкращої» лінії, яка відповідає двом змінним, так що одна змінна може бути використана для прогнозування іншої. Множинна лінійна регресія є розширенням лінійної регресії, де бере участь більше двох змінних, а дані підлаштовуються під багатовимірну поверхню. Використання регресії для пошуку математичного рівняння, яке відповідає даним, допомагає згладити шум. Багато методів згладжування даних також є методами скорочення даних із застосуванням дискретизації [15].

Неузгоджені дані. У записаних даних для деяких транзакцій можуть бути невідповідності. Деякі невідповідності даних можна виправити вручну за допомогою зовнішніх посилань. Це може поєднуватися з процедурами, призначеними для виправлення непослідовного використання кодів. Інструменти розробки знань також можуть використовуватися для виявлення порушення відомих обмежень даних. Наприклад, відомі функціональні залежності між атрибутами можна використовувати для пошуку значень, що суперечать функціональним обмеженням.

Також можуть бути неузгодженості через інтеграцію даних, коли даний атрибут може мати різні назви в різних базах даних. Також можуть існувати надлишки.

Інтеграція даних про платоспроможність. Інтеграція даних об’єднує дані з кількох джерел у єдине сховище даних, як у сховищі даних. Ці джерела можуть включати кілька баз даних, кубів даних або плоских файлів.

Під час інтеграції даних необхідно враховувати ряд питань. Інтеграція схеми може бути винахідливою. Проблема «як можна зіставити еквівалентні сутності реального світу з кількох джерел даних» відноситься до проблеми ідентифікації сутності. Бази даних і сховища даних зазвичай мають метадані, тобто дані про дані. Такі метадані можна використовувати, щоб уникнути помилок під час інтеграції схеми.

Надмірність – ще одне важливе питання. Атрибут може бути надлишковим, якщо його можна "вивести" з іншої таблиці. Невідповідності в іменуванні атрибутів або розмірів також можуть спричинити надмірність у результуючому наборі даних.

Деякі надмірності можна виявити за допомогою кореляційного аналізу. Кореляцію між атрибутами A і B можна виміряти за допомогою

$$r_{A,B} = \frac{\sum\left(A-\overline{A}\right)\left(B-\overline{B}\right)}{(n-1)\sigma_A\sigma_B}, \tag{2.1}$$

де n – кількість кортежів,

і є відповідними середніми значеннями A і B,

і є відповідними стандартними відхиленнями A і B.2

Якщо результуюче значення рівняння (3.1) більше 0, то A і B позитивно корелюють, тобто значення A зростають зі збільшенням значень B. Середнє значення A дорівнює

$$\overline{A} = \frac{\sum A}{n}$$

Стандартне відхилення A дорівнює

$$\sigma_A = \sqrt{\frac{\left(A - \overline{A}\right)^2}{n-1}}$$

Чим вище значення, тим більше кожен атрибут передбачає інший [18]. Отже, високе значення може означати, що A (або B) може бути видалено як надлишковість. Якщо отримане значення дорівнює 0, то A і B незалежні і між ними немає кореляції. Якщо результуюче значення менше 0, то A і B негативно корелюють, де значення одного атрибута зростають із зменшенням значення іншого атрибута. Це означає, що кожен атрибут перешкоджає іншому.

На додаток до виявлення надмірності між атрибутами, дублювання також має бути виявлено на рівні кортежу.

Третім важливим питанням інтеграції даних є виявлення та вирішення конфліктів значень даних. Ретельна інтеграція даних з багатьох джерел може допомогти зменшити та уникнути надмірностей і невідповідностей у результуючому наборі даних. Це може допомогти підвищити точність і швидкість подальшого процесу видобутку.

2.2 Перетворення даних про платоспроможність

Під час перетворення даних дані перетворюються або консолідуються у форми, придатні для видобутку. Перетворення даних може включати наступне:

Згладжування, це працює для видалення шуму з даних: такі методи включають групування, кластеризацію та регресію.

Агрегація, де до даних застосовуються операції зведення або агрегації. Цей крок зазвичай використовується під час побудови куба даних для аналізу даних із різними рівнями деталізації.

Узагальнення даних, де низькорівневі або «примітивні» (необроблені) дані замінюються поняттями вищого рівня за допомогою використання ієрархій понять. Наприклад, значення числових атрибутів, як-от вік, можуть бути зіставлені з поняттями вищого рівня, як-от молодий, середнього віку та старший.

2.3 Нормалізація даних про платоспроможність

Нормалізація, коли дані атрибутів масштабуються таким чином, щоб потрапити в невеликий заданий діапазон.

Конструкція атрибутів або конструкція функцій, коли нові атрибути створюються та додаються з заданого набору атрибутів, щоб допомогти процесу видобутку.

Нормалізація даних. Атрибут нормалізується шляхом масштабування його значень, щоб вони потрапляли в невеликий заданий діапазон, наприклад від 0,0 до 1,0. Нормалізація особливо корисна для алгоритмів класифікації, що включають нейронні мережі, або вимірювання відстані, наприклад класифікація найближчих сусідів і кластеризація. Якщо для визначення класифікації використовується алгоритм зворотного поширення нейронної мережі, нормалізація вхідних значень для кожного атрибута, виміряного в навчальних зразках, допоможе прискорити етап навчання. Існує багато способів нормалізації даних. Ми розглядаємо три: мінімально-максимальна нормалізація, нормалізація z-оцінки та нормалізація десятковим масштабуванням.

Мінімально-максимальна нормалізація. Мінімально-максимальна нормалізація виконує лінійне перетворення вихідних даних. Припустимо, що minA і maxA є мінімальним і максимальним значеннями атрибута A. Min-max нормалізація відображає значення v з A на v' в діапазоні [new_ minA , new_ maxA] шляхом обчислення

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A . \qquad (2.2)$$

Мінімально-максимальна нормалізація зберігає зв'язки між вихідними значеннями даних. Він зіткнеться з помилкою «поза межами», якщо майбутній випадок введення для нормалізації виходить за межі вихідного діапазону даних для A.

2.4 Підсумок

Попередня обробка даних є важливою проблемою як для сховищ даних, так і для інтелектуального аналізу даних, оскільки дані реального світу, як правило, неповні, шумні та суперечливі. Попередня обробка даних включає очищення даних, інтеграцію даних, перетворення даних і скорочення даних.

Процедури очищення даних можна використовувати для заповнення пропущених значень, згладжування шумових даних, виявлення викидів і виправлення невідповідностей даних.

Інтеграція даних об'єднує дані з багатьох джерел, щоб сформувати узгоджене сховище даних. Метадані, кореляційний аналіз, виявлення конфліктів даних і вирішення семантичної неоднорідності сприяють плавній інтеграції даних.

Процедури перетворення даних перетворюють дані у відповідні форми для видобутку.

Методи зменшення даних, такі як агрегація кубів даних, зменшення розмірності, стиснення даних, зменшення чисельності та дискретизація, можуть бути використані для отримання зменшеного представлення даних, мінімізуючи втрату інформаційного вмісту.

Автоматичне створення ієрархій понять для числових даних може включати такі методи, як групування, аналіз гістограми, кластерний аналіз,

дискретизація на основі ентропії та сегментація шляхом природного поділу. Для категоріальних даних ієрархії понять можуть бути створені на основі кількості різних значень атрибутів, що визначають ієрархію.

Хоча було розроблено кілька методів попередньої обробки даних, підготовка даних залишається активною областю досліджень.

3 КЛАСТЕРНИЙ АНАЛІЗ І ПРОЦЕС АВТОМАТИЗАЦІЇ КРЕДИТУВАННЯ

Кластеризація – це процес групування даних у класи таким чином, щоб об’єкти в кластері мали високу схожість один з одним, але дуже відрізнялися від об’єктів в інших кластерах. Відмінності оцінюються на основі значень атрибутів, що описують об'єкти. Часто використовуються міри відстані. Кластеризація сягає корінням у багатьох сферах, включаючи аналіз даних, статистику, біологію та машинне навчання.

3.1 Загальний огляд основних методів кластеризації

Процес групування набору фізичних або абстрактних об’єктів у класи подібних об’єктів називається кластеризацією. Кластер – це набір об’єктів даних, схожих один на одного в одному кластері та несхожих на об’єкти в інших кластерах. Кластер об’єктів даних можна розглядати разом як одну групу в багатьох програмах.

Кластерний аналіз є важливою діяльністю людини. Він широко використовується в багатьох програмах, включаючи розпізнавання образів, аналіз даних, обробку зображень і дослідження ринку. За допомогою кластеризації можна ідентифікувати щільні та розріджені регіони і, отже, виявити загальні моделі розподілу та цікаві кореляції між атрибутами даних.

У бізнесі кластеризація може допомогти маркетологам виявити окремі групи в їхніх клієнтських базах і охарактеризувати групи клієнтів на основі моделей купівлі. Його також можна використовувати для класифікації документів в Інтернеті для пошуку інформації. Як функцію інтелектуального аналізу даних кластерний аналіз можна використовувати як окремий інструмент, щоб отримати уявлення про розподіл даних, спостерігати за характеристиками кожного кластера та зосередитися на певному наборі кластерів для подальшого аналізу. Крім того, це може служити етапом

попередньої обробки для інших алгоритмів, таких як характеристика та класифікація, які потім працюватимуть на виявлених кластерах.

Кластеризація даних активно розвивається. Сфери досліджень включають інтелектуальний аналіз даних, статистику, машинне навчання, технології просторових баз даних, біологію та маркетинг. Завдяки величезній кількості даних, зібраних у базах даних, кластерний аналіз останнім часом став дуже активною темою в дослідженнях інтелектуального аналізу даних.

Як галузь статистики кластерний аналіз широко вивчався протягом багатьох років, зосереджуючись головним чином на кластерному аналізі на основі відстані. Інструменти кластерного аналізу на основі k-середніх, k-medoids та кількох інших методів також були вбудовані в багато програмних пакетів або систем статистичного аналізу, таких як S-Plus, SPSS і SAS [21]. У машинному навчанні кластеризація є прикладом неконтрольованого навчання. На відміну від класифікації, кластеризація та неконтрольоване навчання не покладаються на попередньо визначені класи та навчальні приклади з мітками класів. З цієї причини кластеризація є формою навчання шляхом спостереження, а не навчання на прикладах. У концептуальній кластеризації група об’єктів утворює клас, лише якщо її можна описати концептом. Це відрізняється від звичайної кластеризації, яка вимірює подібність на основі геометричної відстані. Концептуальна кластеризація складається з двох компонентів:

1) він виявляє відповідні класи;

2) формує описи для кожного класу, як у класифікації.

Принцип прагнення до високої внутрішньокласової подібності та низької міжкласової подібності все ще діє.

У інтелектуальному аналізі даних зусилля були зосереджені на пошуку методів ефективного та ефективного кластерного аналізу у великих базах даних. Активні теми досліджень зосереджені на масштабованості методів кластеризації, ефективності методів кластеризації складних форм і типів даних, методах кластеризації великої розмірності та методах кластеризації

змішаних числових і категоріальних даних у великих базах даних.

Кластеризація є складною сферою досліджень, де її потенційні застосування висувають свої особливі вимоги.

3.2 Класифікація основних методів кластеризації

У літературі існує велика кількість алгоритмів кластеризації. Вибір алгоритму кластеризації залежить як від типу доступних даних, так і від конкретної мети та застосування. Якщо кластерний аналіз використовується як описовий або дослідницький інструмент, можна спробувати кілька алгоритмів на тих самих даних, щоб побачити, що дані можуть розкрити.

Загалом основні методи кластеризації можна класифікувати за такими категоріями [25].

1 Методи поділу: для бази даних з n об’єктів або кортежів даних метод поділу створює k поділів даних, де кожен поділ представляє кластер і $k \leq n$. Тобто він класифікує дані на k груп, які разом задовольняють наступним вимогам:

1) кожна група повинна містити хоча б один об'єкт;

2) кожен об'єкт повинен належати рівно до однієї групи. Зверніть увагу, що друга вимога може бути послаблена в деяких методах нечіткого розділення. Посилання на такі прийоми наведено в бібліографічних примітках.

Загальний критерій хорошого поділу полягає в тому, що об’єкти в одному кластері «близько» або пов’язані один з одним, тоді як об’єкти різних кластерів «далеко» або дуже різні. Існують різні види інших критеріїв для оцінки якості перегородок.

Більшість програм використовують один із двох популярних евристичних методів:

1) алгоритм k-*means*, де кожен кластер представлений середнім значенням об’єктів у кластері;

2) алгоритм k-medoids, де кожен кластер представлений одним із об’єктів, розташованих поблизу центру кластера.

Ці евристичні методи кластеризації добре працюють для пошуку кластерів сферичної форми в базах даних малого та середнього розміру. Щоб знайти кластери складної форми та кластеризувати дуже великі набори даних, необхідно розширити методи, засновані на розділенні.

Ієрархічні методи: ієрархічний метод створює ієрархічну декомпозицію заданого набору об’єктів даних. Ієрархічний метод можна класифікувати як агломеративний або розділовий, залежно від того, як формується ієрархічна декомпозиція. Агломеративний підхід, також званий підходом знизу вгору, починається з того, що кожен об’єкт утворює окрему групу. Він послідовно об’єднує об’єкти або групи поруч один з одним, поки всі групи не будуть об’єднані в одну (найвищий рівень ієрархії) або поки не виконується умова завершення. Роздільний підхід, також званий підходом зверху вниз, починається з усіх об’єктів в одному кластері. У кожній наступній ітерації кластер розбивається на менші кластери, доки врешті-решт кожен об’єкт не опиниться в одному кластері або поки не виконується умова завершення.

Ієрархічні методи страждають від того факту, що після виконання кроку (злиття або розділення) його неможливо скасувати. Ця жорсткість корисна тим, що вона призводить до менших витрат на обчислення, не турбуючись про комбінаторну кількість різних варіантів. Однак основна проблема таких методик полягає в тому, що вони не можуть виправити помилкові рішення. Існує два підходи до підвищення якості ієрархічної кластеризації:

1) виконувати ретельний аналіз «зв’язків» об’єктів у кожному ієрархічному розділенні, наприклад у CURE та Chameleon;

2) інтегруйте ієрархічну агломерацію та ітераційне переміщення, спочатку використовуючи ієрархічний агломеративний алгоритм, а потім уточнюючи результат за допомогою ітеративного переміщення, як у BIRCH.

Методи, засновані на щільності: більшість методів поділу

кластеризують об’єкти на основі відстані між об’єктами. Такі методи можуть знаходити лише сферичні кластери і стикаються з труднощами при виявленні кластерів довільної форми. Інші методи кластеризації були розроблені на основі поняття щільності. Їхня загальна ідея полягає в тому, щоб продовжувати розвивати даний кластер до тих пір, поки щільність (кількість об’єктів або точок даних) у «сусідстві» перевищує певний поріг; тобто для кожної точки даних у даному кластері околиці заданого радіуса мають містити принаймні мінімальну кількість точок. Такий метод можна використовувати для фільтрації шумів (викидів) і виявлення кластерів довільної форми.

DBSCAN – типовий метод на основі щільності, який вирощує кластери відповідно до порогу щільності. OPTICS – це метод на основі щільності, який обчислює розширене впорядкування кластеризації для автоматичного та інтерактивного аналізу кластерів.

Методи на основі моделі. Методи на основі моделі висувають гіпотезу про модель для кожного з кластерів і знаходять найкращу відповідність даних даній моделі. Алгоритм на основі моделі може знаходити кластери шляхом побудови функції щільності, яка відображає просторовий розподіл точок даних. Це також веде до способу автоматичного визначення кількості кластерів на основі стандартних статистичних даних, враховуючи «шум» або викиди, що забезпечує надійні методи кластеризації.

Деякі алгоритми кластеризації об’єднують ідеї кількох методів кластеризації, тому іноді важко класифікувати даний алгоритм як однозначно належний лише до однієї категорії методів кластеризації. Крім того, деякі програми можуть мати критерії кластеризації, які потребують інтеграції кількох методів кластеризації.

Нижче ми детально розглядаємо кожен із наведених вище п’яти методів кластеризації.

Метод k-середніх. Алгоритм k-середніх приймає вхідний параметр k і розбиває набір з n об’єктів на k кластерів так, щоб результуюча

внутрішньокластерна подібність була високою, але міжкластерна подібність була низькою. Подібність кластера вимірюється відносно середнього значення об’єктів у кластері, яке можна розглядати як центр тяжіння кластера.

Алгоритм k-середніх виконується наступним чином. По-перше, він випадковим чином вибирає k об'єктів, кожен з яких спочатку представляє середнє або центр кластера. Для кожного з об’єктів, що залишилися, об’єкт призначається кластеру, до якого він найбільш схожий, на основі відстані між об’єктом і середнім значенням кластера. Потім він обчислює нове середнє для кожного кластера. Цей процес повторюється, доки критеріальна функція не збіжиться. Як правило, використовується критерій квадратичної помилки, визначений як

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 , \qquad (3.1)$$

де E – сума квадратичних помилок для всіх об’єктів у базі даних, p – точка в просторі, що представляє даний об’єкт, а mi – середнє значення кластера Ci (p і mi є багатовимірними). Цей критерій намагається зробити отримані k кластерів максимально компактними та розділеними. Алгоритм намагається визначити k розділів, які мінімізують функцію квадратної помилки. Це добре працює, коли скупчення є компактними хмарами, які досить добре відокремлені одне від одного. Метод є відносно масштабованим і ефективним при обробці великих наборів даних, оскільки обчислювальна складність алгоритму становить O(nkt), де n – загальна кількість об’єктів, k – кількість кластерів, а t – кількість ітерацій. У нормі k і t n. Метод часто завершується на локальному оптимумі.

Однак метод k-середніх можна застосовувати лише тоді, коли визначено середнє для кластера. Це може бути не так у деяких програмах, наприклад, коли використовуються дані з категоріальними атрибутами. Необхідність для користувачів заздалегідь вказувати k, кількість кластерів,

можна розглядати як недолік. Метод k-середніх не підходить для виявлення кластерів з неопуклою формою або кластерів дуже різного розміру. Крім того, він чутливий до шуму та викидів даних, оскільки невелика кількість таких даних може суттєво вплинути на середнє значення.

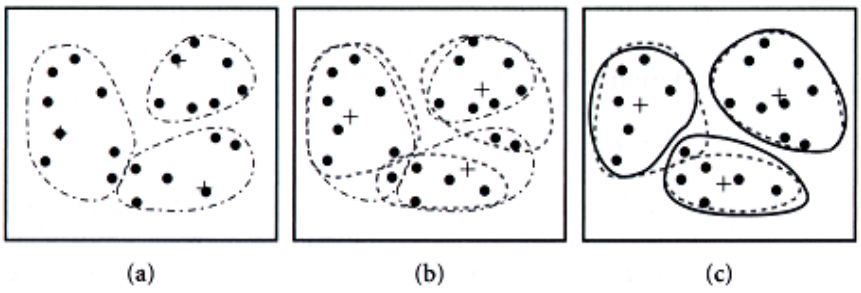

Рисунок 3.1 – Кластеризація набору об’єктів на основі методу k-середніх (Середнє значення кожного кластера позначено знаком "+")

Ієрархічні методи. Метод ієрархічної кластеризації працює шляхом групування об’єктів даних у дерево кластерів. Загалом існує два типи методів ієрархічної кластеризації:

Агломеративна ієрархічна кластеризація: ця стратегія «знизу вгору» починається з розміщення кожного об’єкта у власному кластері, а потім об’єднує ці атомарні кластери.

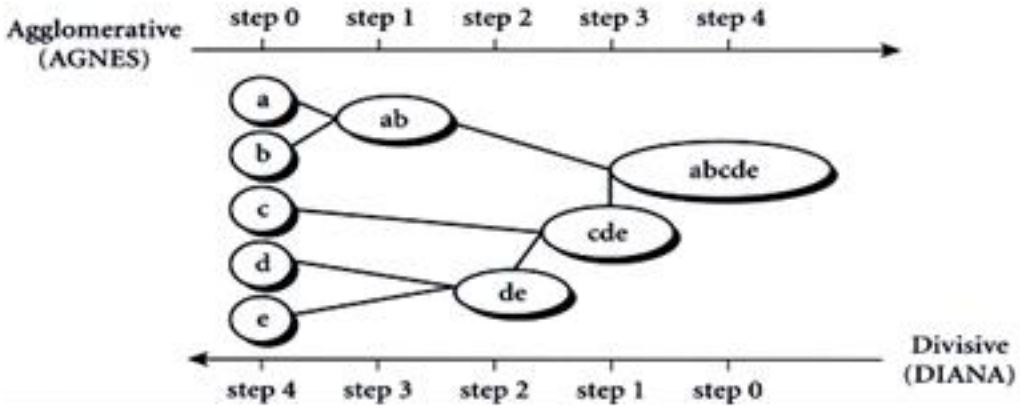

Рисунок 3.2 – Агломеративна та роздільна ієрархічна кластеризація

Агломеративна та роздільна ієрархічна кластеризація на об’єктах даних {a, b, c, d, e} у все більші й більші кластери, поки всі об’єкти не будуть в одному кластері або поки не будуть задоволені певні умови завершення. Більшість методів ієрархічної кластеризації належать до цієї категорії. Вони відрізняються лише визначенням міжкластерної подібності.

Роздільна ієрархічна кластеризація: ця низхідна стратегія виконує зворотну дію агломеративної ієрархічної кластеризації, починаючи з усіх об’єктів в одному кластері. Він ділить кластер на дедалі менші частини, доки кожен об’єкт не утворить кластер самостійно або поки він не задовольнить певні умови завершення, наприклад, отримано бажану кількість кластерів або відстань між двома найближчими кластерами перевищує певну порогову відстань. .

У ієрархічній кластеризації з агломерацією або розділенням користувач може вказати бажану кількість кластерів як умову завершення.

Чотири широко використовувані міри відстані між кластерами є такими, де

|p – p'| це відстань між двома об'єктами або точками p і p', mi є середнім для кластера Ci і ni є кількістю об'єктів у C,:

Мінімальна відстань:

$$d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'| \; ; \tag{3.2}$$

Максимальна відстань:

$$d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|; \tag{3.3}$$

Середня відстань:

$$d_{mean}(C_i, C_j) = |m_i - m_j| \; ; \tag{3.4}$$

Середня відстань:

$$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|. \quad (3.5)$$

Метод ієрархічної кластеризації, хоч і простий, часто стикається з труднощами щодо вибору точок злиття або розбиття. Таке рішення є критичним, тому що після об'єднання або розділення групи об'єктів процес на наступному кроці працюватиме на щойно згенерованих кластерах. Таким чином, рішення про злиття або розділення, якщо вони неправильно вибрані на певному етапі, можуть призвести до кластерів низької якості. Крім того, метод погано масштабується, оскільки рішення про злиття або розділення потребує перевірки та оцінки значної кількості об'єктів або кластерів.

Одним з перспективних напрямків для покращення якості кластеризації ієрархічних методів є інтеграція ієрархічної кластеризації з іншими методами кластеризації для багатофазної кластеризації. Кілька таких методів представлено нижче. Перший, який називається BIRCH, починається з ієрархічного поділу об'єктів за допомогою деревоподібних структур, а потім застосовує інші алгоритми кластеризації для уточнення кластерів. Другий, який називається CURE, представляє кожен кластер певною фіксованою кількістю репрезентативних об'єктів, а потім зменшує їх у напрямку до центру кластера на певну частку. Третій, який називається ROCK, об'єднує кластери на основі їх взаємозв'язку.

Методи на основі щільності. Для виявлення кластерів довільної форми були розроблені методи кластеризації на основі щільності. Вони зазвичай розглядають кластери як щільні області об'єктів у просторі даних, які розділені областями низької щільності (що представляють шум). Метод кластеризації на основі щільності, заснований на з'єднаних областях із достатньо високою щільністю, є алгоритмом кластеризації на основі щільності. Алгоритм вирощує області з достатньо високою щільністю в кластери і виявляє кластери довільної форми в просторових базах даних з шумом. Він визначає кластер як максимальний набір точок, пов'язаних за

щільністю. Основні ідеї кластеризації на основі щільності містять ряд нових визначень: DBSCAN шукає кластери, перевіряючи ∈-околиці кожної точки в базі даних. Якщо ∈ -околиця точки p містить більше ніж MinPts, створюється новий wimp кластера як основний об’єкт. Потім DBSCAN ітеративно збирає об’єкти, досяжні безпосередньо за щільністю, з цих основних об’єктів, що може включати злиття кількох кластерів, досяжних за щільністю. Процес завершується, коли до жодного кластера не можна додати нову точку.

Околиця в радіусі ∈ даного об’єкта називається □-околицею об’єкта.

Якщо ∈ – околиці об'єкта містять принаймні мінімальну кількість, MinPts, об'єктів, тоді об'єкт називається основним об'єктом.

Маючи набір об’єктів, D, ми говоримо, що об’єкт p є безпосередньо досяжним за щільністю від об’єкта q, якщо p знаходиться в межах ∈ околиці q, а q є основним об’єктом.

Об’єкт p досяжний за щільністю від об’єкта q відносно ∈ і MinPts у наборі об’єктів, D, якщо існує ланцюжок об’єктів $p_1,\ldots,p_n$, p1 = q і pn = p такий, що pi+i є досяжним безпосередньо за щільністю від pi відносно ∈ і MinPts, для 1 ≤ i ≤ n, pi ∈ D.

Об’єкт p пов’язаний за щільністю з об’єктом q відносно □ і MinPts у наборі об’єктів, D, якщо існує об’єкт o e D такий, що обидва p і q досяжні за щільністю з o відносно ∈ та MinPts.

Досяжність щільності є транзитивним замиканням досяжності прямої щільності, і цей зв’язок є асиметричним. Тільки основні об'єкти є взаємно доступними. Щільність зв'язності, однак, є симетричним відношенням.

Кластер на основі щільності – це набір пов’язаних за щільністю об’єктів, який є максимальним щодо досяжності щільності. Кожен об'єкт, що не міститься в жодному кластері, вважається шумом.

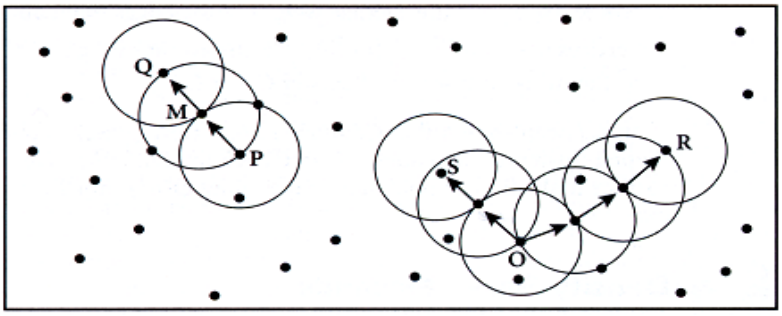

Рисунок 3.3 – Досяжність щільності та підключення щільності в кластеризації на основі щільності

DBSCAN шукає кластери, перевіряючи ∈-околиці кожної точки в базі даних. Якщо ∈ -околиця точки p містить більше ніж MinPts, створюється новий wimp кластера як основний об’єкт. Потім DBSCAN ітеративно збирає об’єкти, досяжні безпосередньо за щільністю, з цих основних об’єктів, що може включати злиття кількох кластерів, досяжних за щільністю. Процес завершується, коли до жодного кластера не можна додати нову точку.

Якщо використовується просторовий індекс, обчислювальна складність DBSCAN становить O(nlogn), де n – кількість об’єктів бази даних. В іншому випадку це O(n2). Алгоритм чутливий до визначених користувачем параметрів.

Методи кластеризації на основі моделі нейронної мережі. Методи кластеризації на основі моделі намагаються оптимізувати відповідність між заданими даними та деякою математичною моделлю. Такі методи часто базуються на припущенні, що дані генеруються сумішшю базових розподілів ймовірностей. Методи кластеризації на основі моделі використовують два основні підходи: статистичний підхід або підхід нейронної мережі.

Підхід нейронної мережі до кластеризації має тенденцію представляти кожен кластер як приклад. Приклад діє як «прототип» кластера і не обов’язково повинен відповідати певному прикладу даних або об’єкту. Нові об’єкти можуть бути розподілені в кластер, екземпляр якого є найбільш схожим, на основі деякої міри відстані. Атрибути об'єкта, призначеного

кластеру, можна передбачити за атрибутами зразка кластера.

Буде згадано два методи нейромережевого підходу до кластеризації. Перший – це змагальне навчання, а другий – самоорганізовані карти функцій, обидві з яких включають конкуруючі нейронні одиниці.

Конкурентне навчання передбачає ієрархічну архітектуру кількох одиниць (або штучних «нейронів»), які конкурують у формі «переможець отримує все» за об’єкт, який зараз представлений системі. На малюнку 4 показано приклад конкурентної системи навчання. Кожне коло представляє одиницю. Одиниця-переможець у кластері стає активною (позначена зафарбованим кружечком), тоді як інші неактивні (позначені порожніми кружечками). Зв’язки між рівнями є збудливими – одиниця на даному рівні може отримувати дані від усіх одиниць на наступному нижчому рівні. Конфігурація активних одиниць у шарі представляє шаблон введення для наступного вищого рівня. Блоки в кластері на даному рівні конкурують між собою, щоб реагувати на шаблон, який виводиться з нижнього рівня. З’єднання всередині шарів є гальмівними, тому лише один блок у будь-якому даному кластері може бути активним. Блок-переможець регулює ваги своїх зв’язків між іншими блоками в кластері, щоб він ще сильніше реагував на майбутні об’єкти, які є такими ж або схожими на поточний. Якщо ми розглядаємо ваги як визначення зразка, то нові об’єкти призначаються кластеру з найближчим зразком. Кількість кластерів і кількість одиниць на кластер є вхідними параметрами.

Наприкінці кластеризації (або будь-якої кластеризації взагалі) кожен кластер можна розглядати як нову «особливість», яка виявляє певну закономірність в об’єктах. Таким чином, отримані кластери можна розглядати як відображення функцій низького рівня на функції вищого рівня.

Кількість шарів може бути довільним. За допомогою самоорганізуючих карт властивостей (SOM) кластеризація також виконується шляхом конкуренції кількох одиниць за поточний об’єкт. Одиниця, вектор ваги якої найближче до поточного об’єкта, стає виграшною або активною одиницею.

Щоб наблизитися до вхідного об’єкта, коригуються ваги виграшної одиниці, а також її найближчих сусідів. SOM припускає, що серед вхідних об’єктів існує певна топологія або порядок, і що одиниці з часом візьмуть цю структуру в просторі. Кажуть, що організація одиниць формує карту ознак. Вважається, що SOM нагадує обробку, яка може відбуватися в мозку, і корисна для візуалізації високовимірних даних у 2- або 3-D просторі.

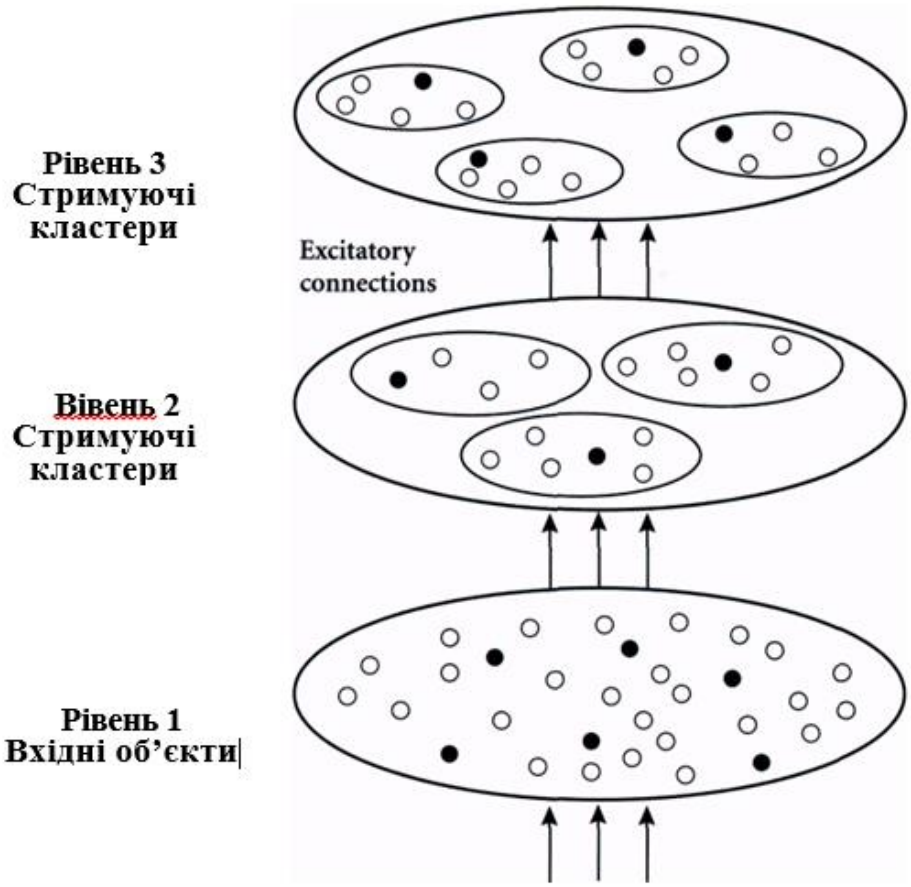

Рисунок 3.4 – Архітектура для змагального навчання

Кількість шарів може бути довільним. За допомогою самоорганізуючих карт властивостей (SOM) кластеризація також виконується шляхом конкуренції кількох одиниць за поточний об’єкт. Одиниця, вектор ваги якої найближче до поточного об’єкта, стає виграшною або активною одиницею. Щоб наблизитися до вхідного об’єкта, коригуються ваги виграшної одиниці, а також її найближчих сусідів. SOM припускає, що серед вхідних об’єктів

існує певна топологія або порядок, і що одиниці з часом візьмуть цю структуру в просторі. Кажуть, що організація одиниць формує карту ознак. Вважається, що SOM нагадує обробку, яка може відбуватися в мозку, і корисна для візуалізації високовимірних даних у 2- або 3-D просторі.

Підхід нейронної мережі до кластеризації має міцні теоретичні зв'язки з реальною обробкою даних мозку. Потрібні подальші дослідження, щоб зробити його легко застосовним до великих баз даних через тривалий час обробки та складні дані.

Аналіз викидів. Дуже часто існують об'єкти даних, які не відповідають загальній поведінці або моделі даних. Такі об'єкти даних, які суттєво відрізняються від решти набору даних або несумісні з ним, називаються викидами.

Викиди можуть бути спричинені помилкою вимірювання або виконання. Наприклад, відображення віку людини як -999 може бути спричинено налаштуванням програми за замовчуванням про незаписаний вік.

Багато алгоритмів інтелектуального аналізу даних намагаються мінімізувати вплив викидів або усунути їх усі разом. Однак самі викиди можуть становити особливий інтерес, наприклад, у випадку виявлення шахрайства, де викиди можуть вказувати на шахрайську діяльність. Таким чином, виявлення та аналіз викидів є цікавим завданням інтелектуального аналізу даних, яке називається аналізом викидів.

Інтелектуальний аналіз викидів можна описати так: задано набір з n точок даних або об'єктів і k, очікувану кількість викидів, знайдіть k перших об'єктів, які значно відрізняються, є винятковими або несумісними щодо решти даних. Проблему аналізу викидів можна розглядати як дві підпроблеми:

1) визначити, які дані можна вважати суперечливими в даному наборі даних;

2) знайти ефективний метод визначення викидів, визначених таким чином.

Проблема визначення викидів є нетривіальною. Якщо для моделювання даних використовується регресійна модель, аналіз залишків може дати хорошу оцінку «екстремальності» даних.

Методи візуалізації даних слабкі у виявленні викидів у даних з багатьма категоріальними атрибутами або в даних високої розмірності. Існує три підходи для виявлення викидів: статистичний підхід, підхід на основі відстані та підхід на основі відхилень. Ми розглядаємо лише два останні з них.

3.3 Характеристика автоматизації процесу кредитування

Частиною управління маркетингом, яку часто забувають, є планування ринку , який поєднує маркетингові очікування виробника з фінансовими вимогами бізнесу. Цей план слід створювати разом із прогнозованим звітом про рух грошових коштів. Однак, щоб бути корисним, прогнозований звіт про рух грошових коштів клієнта повинен передбачати майбутні ціни та заплановані продажі. Кредитор проаналізує історичні доходи та витрати клієнта та прогнозовані потреби в грошових потоках. Здатність клієнта виконувати прогнози часто пов’язана з надійним маркетинговим планом.

Форвардні контракти та ф’ючерсні ринки є прикладами прийняття цінових рішень до фактичної доставки товару. Однак виробники повинні знати витрати на беззбитковість виробництва, щоб ефективно здійснювати маркетинг за допомогою цих методів. Попередня ефективність також є індикатором того, чи включають клієнти такі альтернативи в систему прийняття рішень, а також демонструє наявність чи відсутність здатності до управління ризиками.

Оцінюючи здатність виробника керувати маркетингом своєї продукції, кредитори також розглядають, наскільки добре виробник контролює свій маркетинговий ризик через пряму роздрібну торгівлю, переробку, пакування або створення кооперативів, альянсів і партнерств.

Здатність клієнта витримувати або подолати несприятливі економічні обставини оцінюється як частина загальної здатності управління. Ведення належної фінансової документації та відповідних фінансових інструментів, таких як бюджети руху грошових коштів, запаси, бухгалтерський облік підприємства тощо, забезпечує міцну основу для фінансового менеджменту. Позикодавець повинен оцінити здатність клієнта контролювати витрати та керувати ними, а також оцінювати операційну ефективність (вартість/одиниця тощо). Кредитор також оцінить минуле виконання клієнтом боргових зобов'язань і здатність структурувати борг, щоб він або вона могли інвестувати в капітальні об'єкти. Кредитор повинен оцінити, наскільки добре виробник розуміє своє фінансове становище і що операція може, а що ні.

Загалом надійність розглядається як необхідність, але не єдина умова прийнятного виконання кредиту. Важливість характеру завжди збалансована з рівнем фінансування та загальним ризиком для кредитної організації. Окрім характеру, під час аналізу позики враховуються здібності до управління, прибутковість та інші фактори. Найкращі у світі наміри не можуть повернути позику, якщо від операції не отримано прибутку.

Остаточно треба вирішити такі проблеми: кластерізацію клієнтів і їх класифікацію

3.4 Кроки, за якими працює автоматизація банківських кредитів

Аналіз елементів системи автоматизації процесу кредитування

Сьогодні складність процесу кредитування є одним із найбільш дискусійних питань у банківському секторі з високими перевагами клієнтів і зростанням стандартів комплаєнсу.

З одного боку, клієнти завжди хочуть отримати швидкий і плавний доступ до різноманітних кредитних послуг у будь-який зручний час і в будь-якому зручному місці. З іншого боку, банки конкурують за нові технології,

щоб задовольнити потреби ринку та задовольнити своїх клієнтів. Автоматизація бізнес-процесів має позитивний ефект за рахунок багаторазового прискорення процесів всередині банку, що знижує операційні витрати.

Помітно, що позитивні наслідки процесів автоматизації можна спостерігати в кредитній лінії. До автоматизації процес кредитування міг займати один-два тижні, однак при автоматизації кредитування займає близько 30 хвилин від звернення клієнта через різні сервісні канали до видачі кредиту. Подібна ситуація виникає з випискою про стан рахунку клієнта. До автоматизації дані збирали вручну протягом приблизно трьох днів, після того як система автоматизації передбачала, що процес збору даних займає кілька хвилин.

Ми знаємо, як спростити та покращити ручний процес кредитування, замінивши його автоматизованим: більше немає паперової тяганини, дзвінків, візитів, готівкових операцій.

Розглянемо основні кроки, як працює автоматизація банківських кредитів:

1 збір та аналіз інформації;

2 аналіз ринку кредитування;

3 дослідження процесу кредитування;

4 збір і аналіз даних;

5 початкова попередня обробка даних;

6 аналіз та вибір методів і способів впровадження системи автоматизації процесу кредитування;

7 сегментація клієнта банку;

8 необхідно провести класифікацію клієнта банку.

Рисунок 3.5 – Схема процесу автоматизації системи кредитування

Розробка інформаційної системи

Для демонстрації практичних результатів розроблено та впроваджено програмне забезпечення, яке дозволяє реалізувати запропонований метод автоматизованої обробки вхідних даних ризиків безпеки, що використовуються при оцінці, та отримати кінцеві результати.

Усі дані, які використовуються під час функціонування, зберігаються та обробляються вільно розповсюджуваним MS SQL Express у вигляді бази даних, який можна отримати на сайті розробника [25]. Є багато різних джерел, але поточний додаток має одну відмітну особливість, яка дозволяє обробляти найрізноманітніші набори даних. Програма використовує профілі для зберігання інформації між семінарами та цілими частинами оцінювання. Для початку роботи з програмою необхідно вибрати профіль, який ви хочете використовувати в поточній сесії.

У моделюванні даних ми проходимо три різні етапи:

Концептуальна модель даних: найбільш абстрактна модель даних, яка описує елементи даних без особливих деталей.

Логічна модель даних: концептуальна модель з більшою кількістю технічних деталей.

Фізична модель даних: додана логічна модель з усіма деталями фізичної бази даних (типи даних, обмеження, індекси, схеми тощо).

Концептуальна модель даних є першою та найбільш абстрактною моделлю даних у процесі моделювання даних. Це високорівнева діаграма, яку ми використовуємо для визначення, опису, організації та представлення елементів даних і їхніх зв'язків із відносно невеликою кількістю деталей. Концептуальні моделі даних містять лише:

Сутності реального світу, які є нашими основними елементами даних.

Ця модель не має технічних деталей, таких як атрибути, типи даних тощо. Ми використовуємо концептуальну модель даних для спілкування з різними діловими людьми під час визначення бізнес-вимог до бази даних і представлення концепцій (наприклад, для їхнього відгуку). Ми не використовуємо ці моделі для спілкування з технічними командами. Таким чином, ми використовуємо прості терміни в концептуальній моделі даних.

Сутностями бази даних є такі підсистеми: Клієнт, Код Паспорт, Договір, Платежі, Житло, Авто, Товар, Кредит, Освіта, Дохід.

Рисунок 3.6 демонструє концептуальну модель даних.

Побудова ER- діаграми. Діаграму «сутність-зв'язок» можна вважати планом бази даних. Ми використовуємо діаграми сутності-зв'язку, коли моделюємо дані, що зберігаються (або будуть зберігатися) у базі даних. Діаграми ER дозволяють обговорювати вимоги, наприклад, яку інформацію потрібно зберігати, які аспекти інформації потрібно захищати та як інформація пов'язана між собою. Під час і після процесу проектування архітектори бази даних та інші переглядають цей план, щоб переконатися, що всі важливі міркування враховано. Крім того, діаграми ER сприяють обговоренню вмісту з бізнес-стейкхолдерами та іншими, допомагаючи

розробникам переконатися, що база даних відповідає потребам користувачів. Отже, давайте поговоримо про те, що таке ERD і як їх можна створити.

Рисунок 3.6 – Концептуальна модель бази даних

Діаграма сутність-зв’язок має важливе значення для моделювання даних і проектування бази даних. Це базовий дизайн, на основі якого ми будуємо базу даних. Діаграма ER складається з:

Сутності або дані, які нам потрібно зберігати. Це може бути особа, місце, річ, процес тощо. У базі даних роздрібного продавця клієнти, продукти та замовлення є сутностями.

Атрибути, або характеристики сутностей. Це деталі, які додають інформацію про кожну сутність. Наприклад, сутність клієнта включатиме такі атрибути, як ім’я та прізвище клієнта, його адресу електронної пошти або номер телефону та номер його рахунку. Якщо ви не впевнені щодо цих понять, зверніться до цієї статті, яка описує різницю між сутностями та атрибутами. Відносини, або як пов’язані дані у двох сутностях. Наприклад, існуватиме зв’язок між сутністю замовлення та сутністю клієнта, оскільки клієнти розміщують замовлення.

Складання ER діаграми. Одне з місць, з яких можна розпочати

креслення ERD – це дошка. Але в якийсь момент вам потрібно буде створити добре задокументований і точний план моделі даних; для цього вам знадобиться інструмент моделювання бази даних. Іноді використовують звичайні програми для малювання для моделювання даних. Перевага, яку я бачу в спеціальному інструменті моделювання даних і ERD, полягає в тому, що він створений спеціально для цього завдання.

Описана вище модель даних представлена у вигляді ER-діаграми на рисунку 3.7.

Сутність	Атрибути
Договір	Номер договору (PK, FK)
	Номер клієнта (FK)
	Кредитна валюта
	Процентна ставка
	Вид кредиту
	опис
	Спосіб погашення кредиту
Погашення	Номер погашення (PK)
	Номер договору (FK)
	Місяць оплати
	Дата оплати
	Основна сплата цього погашення
	Відсотки, сплачені за це погашення
Житло	Номер договору (PK, FK)
	Загальний кредит
	Сума початкового платежу
	Термін кредитування
	Застава
	Паспортні дані продавця
	Гаранти
	Рахунок для молодіжного фонду
Авто	Номер договору (PK, FK)
	Загальний кредит
	Сума початкового платежу
	Термін кредитування
	Застава
	Гаранти

Рисунок 3.7 – ER-діаграма

4 ОЦІНКА КРЕДИТУВАННЯ НА БАЗІ ШТУЧНОГО ІНТЕЛЕКТУ

4.1 Застосування алгоритмів кластеризації

Алгоритми кластеризації розгортаються як частина широкого спектру технологій. Науковці даних покладаються на алгоритми, щоб допомогти з класифікацією та сортуванням.

Наприклад, велика кількість програм для роботи з людьми може бути більш успішною за допомогою кращих алгоритмів кластеризації. Школи, можливо, захочуть розподілити учнів у секції класу на основі їхніх талантів і здібностей. Алгоритми кластеризації об’єднають студентів зі схожими інтересами та потребами.

Деякі компанії хочуть розділити своїх потенційних клієнтів на різні категорії, щоб вони могли надавати клієнтам більш відповідні послуги. Покупцям-початківцям можна запропонувати велику допомогу, щоб вони могли зрозуміти продукти та варіанти. Досвідчених клієнтів можна відразу перевести до пропозицій і, можливо, отримати спеціальні ціни, які працюють для подібних покупців.

Є багато інших прикладів із різноманітних галузей, таких як виробництво, банківська справа та судноплавство. Усі покладаються на алгоритми, щоб розділити навантаження на менші підмножини, які можуть однаково оброблятися. Усі ці параметри значною мірою залежать від збору даних.

Якщо кластер визначається відстанями між елементами даних, вимірювання відстані є важливою частиною процесу. Багато алгоритмів покладаються на стандартні способи обчислення відстані, але деякі покладаються на різні формули з різними перевагами.

Багато хто вважає саму ідею «відстані» незрозумілою. Ми так часто використовуємо цей термін, щоб визначити, яку відстань нам потрібно

подолати в кімнаті чи навколо земної кулі, що може здатися дивним розглядати дві точки даних – наприклад, опис уподобань користувача щодо морозива чи кольору фарби – як розділені будь-якою відстанню. Але це слово є природним способом описати число, яке вимірює, наскільки елементи можуть бути близькі один до одного.

Вчені та математики зазвичай покладаються на формули, які задовольняють те, що вони називають «нерівністю трикутника». Тобто відстань між точками A і B плюс відстань між B і C більша або дорівнює відстані між A і C. Коли формула гарантує це, процес стає більш послідовним. Деякі також покладаються на більш суворі визначення, такі як «ультраметрика», які пропонують більш складні гарантії. Алгоритмам кластеризації, строго кажучи, не потрібно наполягати на цьому правилі, оскільки будь-яка формула, яка повертає число, може це зробити, але результати, як правило, кращі.

SageMaker: готове рішення Amazon для створення моделей ШІ підтримує низку підходів, як-от кластеризація K-means. Їх можна протестувати в ноутбуках і розгорнути після того, як програмне забезпечення створить модель.

Google включає різноманітні алгоритми кластеризації, які можна розгорнути, зокрема алгоритми на основі щільності, центроїда та ієрархічні алгоритми. Їхня співпраця пропонує гарну можливість вивчити потенціал перед розгортанням алгоритму.

Інструменти Azure від Microsoft, як і дизайнер машинного навчання, пропонують усі основні алгоритми кластеризації у формі, яка відкрита для експериментів. Його системи спрямовані на обробку багатьох деталей конфігурації для створення конвеєра, який перетворює дані на моделі.

IBM пропонує кластеризацію як у рамках своєї науки про дані, так і в рамках своїх інструментів ШІ. Обидва реалізують основні алгоритми та надають такі інструменти, як Cloud Pak for Data або Watson Studio.

Oracle також пропонує технологію кластеризації в усіх своїх програмах

штучного інтелекту та наукових даних. Він також вбудував алгоритми у свою провідну базу даних, щоб кластери можна було створювати всередині сховища даних

В роботі пропонується модель нейронної мережі для визначення платоспроможності клієнта.

4.2 Метод кластеризації клієнтів на базі нейронної мережі

Зі швидким поширенням Інтернету також стрімко розвиваються онлайн-системи для покупок, банківських операцій, здійснення платежів, біржової торгівлі тощо. Однак через відкритість мережі форми мережевого вторгнення стають все більш різноманітними, тому мережі та системи зазнають все більших загроз. Тому виявлення вторгнення в мережу стало критичною проблемою безпеки мережі. Останніми роками вчені всього світу приділяють все більшу увагу виявленню вторгнень. Мета полягає в тому, щоб визначити будь-яку поведінку, яка може поставити під загрозу цілісність, конфіденційність або доступність системи. Його можна визначити як ідентифікацію людей, які мають доступ до комп’ютерної системи. Поточні методи виявлення вторгнень у мережу можна розділити на дві категорії: виявлення вторгнень зловживання та виявлення аномальних вторгнень. Можливість виявлення вторгнень зловживання в основному залежить від повноти бази знань виявлення. Його недолік полягає в тому, що він не може знайти невідомі форми вторгнення. Виявлення ненормального вторгнення базується на визначенні різниці між виявленою та прийнятною поведінкою.

Було запропоновано неконтрольовану самоорганізовану конкурентну нейронну мережу під назвою нейронна мережа Кохонена. Він може досягти автоматичної кластеризації за допомогою самоорганізуючого відображення функцій для налаштування ваг мережі. Нейронна мережа Кохонена [22] складається з двох рівнів прямого зв’язку, а саме вхідного рівня та вихідного рівня. Вхідний шар відображається у двовимірній сіті відповіді на вихідному

шарі на основі вагових коефіцієнтів. Топологія WTA нейронної мережі Кохонена показана на рисунку 4.1.

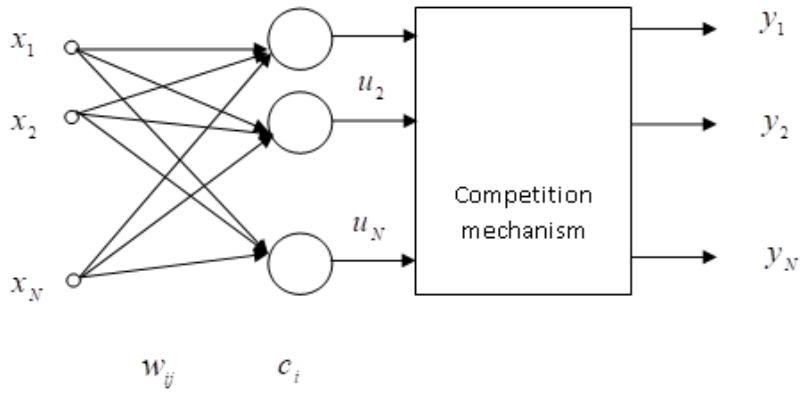

Рисунок 4.1 – Структура WTA нейронної мережі

У нейронній мережі Кохонена евклідова відстань кожного нейрона отримується шляхом обчислення вхідного власного вектора для відповідного вихідного рівня. Нейрон із найменшою евклідовою відстанню є вищим нейроном, і його вагові коефіцієнти зв'язку налаштовані, щоб зробити його ближчим до вихідного вхідного вектора. Область, прилегла до виграшного нейрона, також регулюється вагою з'єднання, щоб зробити його ближчим до вхідного вектора.

На етапі навчання кожен вхідний вектор Xs вводиться в мережу, і тільки ті виграшні нейрони, найближчі до поточного вагового вектора введення, отримують відповідний стимул.

Змагальний процес навчання

Основним завданням для WTA мереж є вивчення моделі середовища, в яке він вбудований, і підтримка моделі, достатньо узгодженої з реальним світом, щоб досягти визначених цілей відповідної програми. WTA мережа дізнається про своє оточення через інтерактивний процес коригування, застосованого до вагових коефіцієнтів з'єднання. В ідеалі мережа стає більш обізнаною про своє середовище після кожної ітерації процесу навчання.

У змагальному навчанні вихідні нейрони нейронної мережі конкурують

між собою за те, щоб стати активними (за те, щоб бути «звільненими»). У той час як у багатокористувацьких персептронах кілька вихідних нейронів можуть бути активними одночасно, у змагальному навчанні в будь-який момент часу активний лише один вихідний нейрон. Є три основні елементи, необхідні для побудови мережі з правилом конкурентного навчання, стандартною технікою для цього типу штучних нейронних мереж [30]:

- набір нейронів, які мають однакову структуру і з'єднані з початково випадково вибраними вагами. Тому нейрони по-різному реагують на заданий набір вхідних зразків;
- граничне значення, яке визначається силою кожного нейрона;
- механізм, який дозволяє нейронам конкурувати за право реагувати на певну підмножину вхідних даних, так що одночасно активним є лише один вихідний нейрон.

У найпростішій формі змагального навчання WTA мережа має один шар вихідних нейронів, кожен з яких повністю підключений до вхідних вузлів (рисунок 4.2).

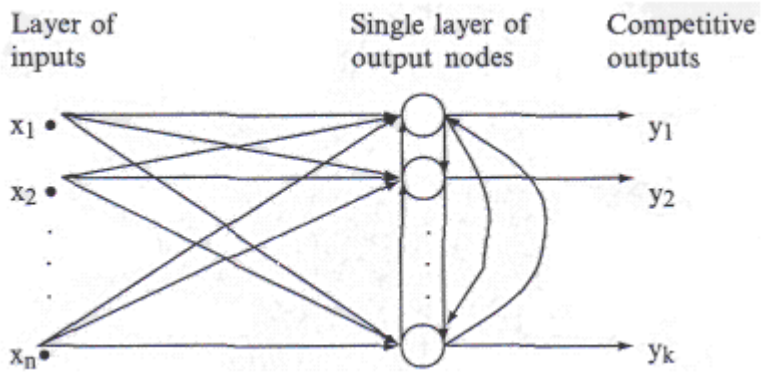

Рисунок 4.2 – Схема простої архітектури конкурентної мережі

WTA: вимірювання подібності добутку. Щоб нейрон k був нейроном-переможцем, його чисте значення netk для заданої вхідної вибірки X = {x1, x2, ..., xn} має бути найбільшим серед усіх нейронів у мережі. Вихідний сигнал yk переможного нейрона k встановлюється рівним одиниці; вихідні

дані всіх інших нейронів, які програли конкуренцію, встановлюються рівними нулю. Це можна написати

$$y_k = \begin{cases} 1 & if\ \ net_k > net_j,\ for\ all\ \ j,\ j \neq k \\ 0 & otherwise \end{cases}, \qquad (4.1)$$

де індуковане локальне значення netk представляє сукупну дію всіх прямих і зворотних вхідних даних до нейрона k.

Нехай w_{kj} відзначає синаптичні ваги, що з'єднують вхідний вузол j з нейроном k. Далі нейрон навчається, переміщуючи синаптичні ваги зі своїх неактивних вхідних вузлів на активні вхідні вузли. Якщо певний нейрон перемагає в змаганні, кожен вхідний вузол цього нейрона відмовляється від деякої частки своєї синаптичної ваги, а втрачена вага потім розподіляється між активними вхідними вузлами. Згідно зі стандартним правилом конкурентного навчання, зміна Δwjk, застосована до синаптичної ваги W_{kj}, визначається як

$$\Delta w_{kj} = \begin{cases} \eta(x_j - w_{kj}) & if\ \ neuron k\ wins\ the\ competition \\ 0 & otherwise \end{cases}, \qquad (4.2)$$

де η – параметр швидкості навчання. Правило має загальний ефект переміщення синаптичних ваг нейрона-переможця до вхідного шаблону X. Геометрична аналогія, представлена на рисунку 4.3, ілюструє суть змагального навчання. Кожен вихідний нейрон виявляє кластер вхідних зразків, переміщуючи свої синаптичні ваги до центру тяжіння виявленого кластера. Рисунок 4.3 ілюструє

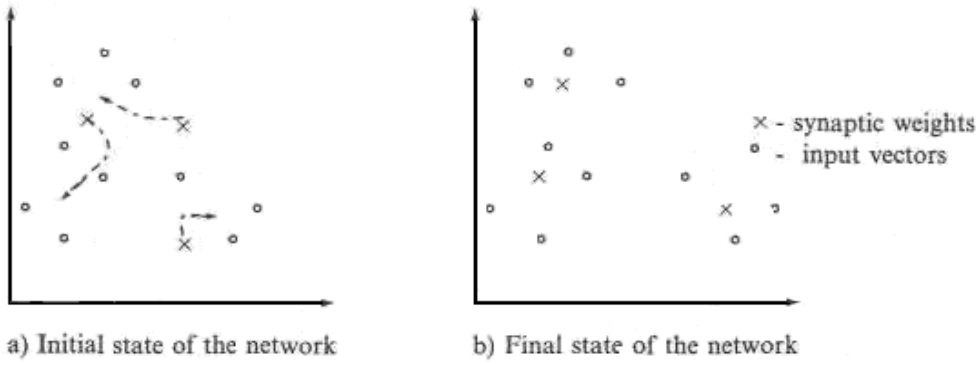

Рисунок 4.3 – Геометрична інтерпретація змагального навчання

Це групування на основі кореляції даних виконується автоматично. Однак для того, щоб ця функція виконувалася стабільно, вхідні зразки повинні розпадатися на досить різні групи. Інакше мережа може працювати нестабільно.

Змагальні (або переможець отримує все) нейронні мережі часто використовуються для кластеризації вхідних даних, де кількість вихідних кластерів задана заздалегідь. Добре відомі приклади WTA мереж, що використовуються для кластеризації на основі неконтрольованого індуктивного навчання, включають векторне квантування навчання Кохонена, самоорганізуючу карту (SOM) і мережі на основі моделей теорії адаптивного резонансу.

WTA: міра подібності евклідової відстані. Недоліком використання скалярного добутку як міри подібності є те, що (принаймні) вхідні вектори повинні бути нормалізовані. Але для деяких завдань нормалізація може бути недійсною і навіть некоректною. Однією з проблем, яка може виникнути, є перекриття кластерів (рисунок 4.4). Після нормалізації опорні вектори двох різних кластерів і розташовані в одній області одиничної гіперсфери, що призводить до неправильної кластеризації. Звичайно, можна виконати відповідну попередню обробку даних, але проблема полягає в тому, що потрібно мати апріорну інформацію про розподіл даних, яка, як правило, невідома.

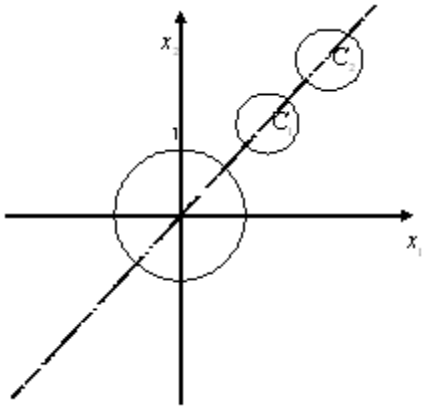

Рисунок 4.4 – Перекриття кластерів

Подібність між еталонними векторами та вхідними шаблонами можна виміряти евклідовою мірою відстані. Переможцем є нейрон, опорний вектор якого є найближчим для вхідної вибірки:

$$k: \ \left\|\mathrm{w_k} - x\right\| \ \le \left\|w_j - x\right\| \ \ \forall \mathrm{j} \tag{4,3}$$

Слід зауважити, що за умови нормалізації (4.3) дорівнює (2.1, 2.2). Тому евклідова відстань є більш загальною формою міри подібності скалярного добутку. У цьому випадку замість обертання опорних векторів вони зсуваються в бік вхідних шаблонів:

$$w_k(t+1) = w_k(t) + \eta(t)(x(t) - w_k(t)) \tag{4.4}$$

І знову оновлюються тільки ваги переможця.

Для алгоритму WTA надзвичайно важливо правильно ініціалізувати опорні вектори. Якщо опорний вектор нейрона ініціалізовано областю вхідного простору з низькою щільністю даних, ймовірність того, що нейрон стане переможцем для будь-якого вхідного шаблону, дуже низька. Тому такі

нейрони (так звані «мертві нейрони») можуть ніколи не використовуватися системою. Щоб подолати цю проблему, було запропоновано декілька підходів. Можна, напр. ініціалізувати опорні вектори випадково взятими вхідними зразками або розширити (4.4) до так званого «навчання з витоком»:

$$w_l(t+1) = w_l(t) + \gamma(t)(x(t) - w_l(t)) \quad \forall \mathrm{l} \neq \mathrm{k} \tag{4.5}$$

при чому $\gamma(t) << \eta(t)$.

Інший широко використовуваний підхід полягає в тому, щоб «запам'ятати» кількість адаптацій, виконаних для кожного нейрона, і частіше нейрон виграє, він стає менш чутливим до наступних вхідних шаблонів.

4.3 Алгоритм сегментації клієнтів за допомогою нейронної мережі

Мережа Кохонена має лише два шари: вхідний рівень і вихідний рівень радіальних одиниць (також відомий як шар топологічної карти). Одиниці на топологічній карті розташовані в просторі, як правило, у двох вимірах.

Мережі Кохонена навчаються за допомогою ітераційного алгоритму. Починаючи з початково випадкового набору радіальних центрів, алгоритм поступово налаштовує їх, щоб відобразити кластеризацію навчальних даних. На одному рівні це можна порівняти з алгоритмами підвибірки та K-Means, які використовуються для призначення центрів у мережах RBF та GRNN, і справді алгоритм Кохонена можна використовувати для призначення центрів для цих типів мереж. Однак алгоритм також діє на іншому рівні.

Процедура ітераційного навчання також організовує мережу так, що одиниці, що представляють центри, розташовані близько один до одного у вхідному просторі, також розташовані близько один до одного на топологічній карті. Ви можете думати про топологічний рівень мережі як про грубу двовимірну сітку, яку потрібно згорнути та спотворити у N-вимірному вхідному просторі, щоб максимально зберегти вихідну структуру. Очевидно,

будь-яка спроба представити N-вимірний простір у двох вимірах призведе до втрати деталей; однак ця техніка може бути корисною, оскільки дозволяє користувачеві візуалізувати дані, які інакше було б неможливо зрозуміти.

Базовий ітераційний алгоритм Кохонена просто проходить через кілька епох, у кожній епосі виконується кожен випадок навчання та застосовується наступний алгоритм:

1) ініціалізувати ваги малими випадковими значеннями;

2) представити новий вхідний вектор;

3)знайти вихідний вузол, ваги якого найближчі до вхідного вектора, використовуючи міру відстані;

4) модифікувати вагові вектори вузла та його топологічних сусідів за допомогою рівняння навчання;

5) змінити розмір околиці та значення коефіцієнта швидкості навчання. Часто обидва лінійно скорочуються повільно протягом періоду навчання;

6) переходіть до кроку 2, доки навчання не буде завершено.

Алгоритм використовує швидкість навчання із загасанням у часі, яка використовується для виконання зваженої суми та гарантує, що зміни стають дедалі помітнішими з плином епох. Це гарантує, що центри встановлюються на компромісне представлення випадків, які призводять до того, що нейрон виграє.

Властивість топологічного впорядкування досягається додаванням поняття околиці до алгоритму. Околиці – це набір нейронів, що оточують нейрон-переможець. Сусідство, як і швидкість навчання, слабшає з часом, так що спочатку досить велика кількість нейронів належить до околиці (можливо, майже вся топологічна карта); на останніх етапах околиці будуть нульовими (тобто складатимуться виключно з самого нейрона-переможця). В алгоритмі Кохонена коригування нейронів фактично застосовується не тільки до нейрона-переможця, але й до всіх членів поточного оточення, згідно з правилом

$$\Delta w_i = \eta \Omega_c(i)(x - w_i), i = \overline{1,n} \quad (4.6)$$

де η – коефіцієнт швидкості навчання,

$\Omega_c(i)$ – функція сусідства,

$(x - w_i)$ – відстань між сигналом і синаптичною вагою.

Часто навчання навмисно проводиться у дві окремі фази: відносно коротка фаза з високими темпами навчання та сусідством, і довга фаза з низьким рівнем навчання та нульовим або майже нульовим сусідством.

Мережі Кохонена також використовують поріг прийняття під час виконання класифікації. Оскільки рівень активації нейрона в мережі Кохонена є відстанню нейрона від вхідного регістра, поріг прийняття діє як максимальна розпізнана відстань. Якщо активація нейрона-переможця перевищує цю відстань, мережа Кохонена вважається невизначеною. Таким чином, позначаючи всі нейрони та відповідним чином встановлюючи поріг прийняття, мережа Кохонена може діяти як детектор новизни (вона повідомляє про невизначеність, лише якщо вхідний випадок досить несхожий на всі радіальні одиниці) [33].

4.4 Програмна реалізація моделі нейронної мережі для сегментації та класифікації клієнтів банку

Програма для класифікації клієнтів банку та вирішення завдань кластеризації призначена для відділів маркетингу банку. Дозволяє відобразити сегменти ринку клієнтів банку. Це основна інформація для розробки стратегії розвитку банку, побудови маркетингового плану.

Програма має зручний інтерфейс, реалізований за допомогою стандартного набору компонентів.

Для вирішення завдання сегментації клієнта банку створена можливість зміни набору вхідних даних, завдяки чому сегментація може

здійснюватися динамічно. При проведенні сегментації можна використовувати різні методи та алгоритми. Це зроблено тому, що для різних зразків використання різних методів і алгоритмів буде більш ефективним.

Зручний інтерфейс дозволяє бачити топологію мережі, значення центроїда класу, результати, вхідні та вихідні дані.

Під час класифікації розв'язування завдань нові клієнтські дані можна вводити вручну або завантажувати з файлу.

У програмі реалізована можливість роботи нейронної мережі зустрічного поширення як в режимі акредитації, так і в режимі інтерполяції. Це дозволяє аналізувати не тільки кількість класів, до яких належить клієнт, але ймовірність належності до всіх класів.

Система працює під керуванням операційної системи Windows. Програма написана мовою програмування C++ з підключенням таблиць в ExcelТому таблиці в Excel потрібно зберігати як файли Excel 2019 для Windows.

Результати роботи нейронної мережі Кохонена. Вирішення першої задачі – сегментація ринку клієнта банку – здійснювалося за допомогою нейронної мережі Кохонена. Навчання та робота нейронної мережі проводилась на основі набору, який складався із 144 додатків клієнтів банку. У результаті було виділено чотири сегменти. Інтерпретовані значення кожної характеристики кожного класу представлені в таблиці 4.1.

Проаналізувавши результати кластеризації, отримані як результати роботи нейронної мережі, центроїду кластера можна дати такі назви:

Клас 1 – «Клієнт ненадійний із середнім доходом»;

Клас 2 – «Клієнт ненадійний, забезпечений»;

Клас 3 – «Клієнт надійний із середнім доходом»;

Клас 4 – «Клієнт із середньою надійністю та низьким доходом».

Таблиця 4.1 – Характеристики центроїда класу

№	Характеристика	Середнє характерне значення			
		Клас 1	Клас 2	Клас 3	Клас 4
1	Стать	чоловічий	чоловічий	жіноча	жінка чоловік
2	Подружній статус	одружений	неодружений	одружений	неодружений
3	діти	Є діти	Дітей немає	Є діти	Дітей немає
4	Валюта	UKR	DM	США/УКР	UKR
5	сума	1454	3179	1206	1118
6	Відсоткова ставка	25,79%	28,82%	26,02%	20,96%
7	термін	13 місяців	13 місяців	12 місяців	7 місяць
8	Тип платежів	щомісяця	В кінці терміну	В кінці терміну	щомісяця
9	Тип	студент / діти	звичайний	пенсіонер / студент	пенсіонер / студент
10	Дані про відкриття вкладу	15.03.2000	15.03.2000	13.12.1999	03.11.1999
11	реклама	радіо	Радіо / метро	Радіо / метро	метро
12	Досвід роботи з банками	4,94 з 7	4,47 з 7	5,35 з 7	4,11 з 7

Таблиця 4.2 – Віднесення нового клієнту до четвертого класу у результаті класифікації

№	Свойства	Характеристики нового кліента	Средние значения свойств Сегмент 4
1	стать	Чол.	Жін./Чол. .(1.56)
2	Сімейний стан	Неодружений	Неодружений (1.6)
3	Діти	Нема дітей	Нема дітей (1.6)
4	Валюта вкладу	UKR	UKR (2.3)
5	Сума вкладу	700	1118 (1250)
6	Відсоток	22.79%	20.96%(20.9)
7	Срок	13мес.	7міс. .(7.4)
8	Характер вкладу	Кожного місяця о	Кожного місяця
9	Вид вкладу	пенсійний	пенсійний/студентський
10	Дата відкриття рахунку	Метро	Метро
11	Вид рекламы	15.08.2023	03.11.2023
12	Досвід роботи з банками	4.77 из 7	4.59 из 7

4.5 Підсумок

Конкурентне навчання забезпечує ефективну адаптивну класифікацію, але воно страждає від кількох методологічних проблем. Перша проблема

полягає в тому, що вибір швидкості навчання r) змушує компроміс між швидкістю навчання та стабільністю кінцевих вагових факторів. Швидкість навчання, близька до нуля, призводить до повільного навчання. Однак коли вектор ваги досягає центру кластера, він буде прагнути залишатися поблизу центру. Навпаки, висока швидкість навчання призводить до швидкого, але нестабільного навчання. Більш серйозна проблема стабільності виникає, коли кластери розташовані близько один до одного, що призводить до того, що вагові вектори також стають близькими, і процес навчання змінює свої значення та відповідні класи з кожним новим прикладом. Проблеми зі стабільністю конкурентного навчання можуть також виникнути, коли початковий ваговий вектор нейрона розташований настільки далеко від будь-якого вхідного вектора, що він ніколи не виграє змагання, а отже, ніколи не навчається. Нарешті, процес змагального навчання завжди має стільки кластерів, скільки вихідних нейронів. Це може бути неприйнятним для деяких програм, особливо коли кількість кластерів невідома або якщо її важко оцінити заздалегідь.

У більшості випадків алгоритм Кохонена використовує швидкість навчання із загасанням у часі. Мережа Кохонена може навчитися розпізнавати кластери даних, а також може пов’язувати подібні класи один з одним. Користувач може створити розуміння даних, які використовуються для уточнення мережі. Коли класи даних розпізнаються, їх можна позначати, щоб мережа стала здатною виконувати завдання класифікації. Мережі Кохонена також можна використовувати для класифікації, коли вихідні класи доступні негайно – перевагою в цьому випадку є їх здатність підкреслювати подібність між класами. Мережі Кохонена можуть навчитися розпізнавати кластери в навчальних даних і реагувати на них.

ВИСНОВКИ

Однією з основних цілей цієї роботи було вивчення основи багатокритеріального процесу прийняття рішень, існуючих методів кластеризації, аналізу методів кластеризації додатків у процесі MCDM та вивчення методів попередньої обробки даних. Важливо було порівняти різні моделі оцінки та знайти нову комбінацію кластеризації, попередньої обробки та підходів до прийняття рішень, яка має максимальну ефективність.

За результатами дослідження можна стверджувати, що більшість методів прийняття рішень використовують функцію оцінювання на основі адитивної або мультиплікативної форми. Функція на основі евклідової відстані збільшує обчислювальне навантаження, але також значно покращує точність результату. Евклідова відстань обчислює точну різницю між двома альтернативами в декартових координатах.

Для продуктивності кластеризації та рішень для прийняття рішень дані повинні бути очищені та нормалізовані до уніфікованих показників. Запропоновано підхід мінімаксного методу, який дозволяє масштабувати дані в інтервалі [0,1], однозначно визначаючи найгірше та найкраще значення часткових критеріїв. Модифікація мінімаксного методу масштабує дані в будь-якому інтервалі, зберігаючи точність нормалізації, коригує значення, враховуючи значущість кожного значення критерію.

Попередня кластеризація дозволяє зробити висновки про відповідність і важливість даних. За результатами кластеризації експерт може зробити висновок про відповідну структуру моделі оцінки та вагові коефіцієнти моделі. Новизна в роботі полягає в поєднанні методів прийняття рішень з кластеризацією даних.

Використання нейромережевого підходу до кластеризації даних дозволяє зберігати, регулювати, додатково аналізувати та обробляти результати кластеризації, згадувати тенденції та поведінку спостережуваних

даних у задачах прийняття рішень, створювати шаблони для подібних задач.

Розроблене демонстраційне програмне забезпечення порівнює ефективність розглянутих підходів. Доведено, що методи ідентифікації для багатофакторної оцінки з використанням нейронних мереж працюють ефективно навіть у тих сферах, де класичні методи прийняття рішень неспроможні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Хоревський Л., Старовойтова О . Споживче кредитування: чи можна досягти успіху за допомогою автоматизації? Банківські технології . М.: Профі – Прес . – 2004 рік.

2. Nijkamp, P., P. Rietveld and H. Voogd, Multicriteria Analysis for Physical Planning, Elsevier, Amsterdam, 1990.

3. Slowinski, R., Intelligent Decision Support, Kluwer, Dordrecht, 1995.

4. Зозулєв А. В. Сегментація ринку. М. Рібарі , 2003 – 178с

5. F.S. Roberts, Measurement Theory, Addison-Wesley, 1979.

6. J. Barzilai, “Notes on the Analytic Hierarchy Process,” Proceedings of the NSF Design and Manufacturing Research Conference, Tampa, Florida, pp. 1–6, 2001.

7. R.L. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley & Sons, New York, 1976

8. Raiffa, H. and Schleifer, R. Applied Statistical Decision Theory, Harvard University, Press, Boston, 1961.

9. French, S. Decision theory: an introduction to the mathematics of rationality, Ellis, Horwood, Chichester, 1988.

10. DeGroot, M. H. Changes in utility as information, Theory and Decision 17: 287–303, 1984.

11. K. Yoon and G. Kim, “Multiple attribute decision analysis with imprecise information,” IIE Trans., vol. 21, no. 1, pp. 21–25, 1989.

12. B. F. Hobbs and P. M. Meier, “Multicriteria method for resource planning: an experimental comparison,” IEEE Trans. on Power Systems, vol. 9, no. 4, pp. 1811–1817, Nov. 1994.

13. Agosta, L., The Essential Guide to Data Warehousing, Prentice Hall, Upper Saddle River: N.J., 2000.

14. Han, J. and M. Kamber, Data Mining: Concepts and Techniques,

Morgan Kaufmann, San Francisco, 2000.

15. Kaudel, A., M. Last, H. Bunke, eds., Data Mining and Computational Intelligence, Physica-Verlag, Heidelberg: Germany, 2001.

16. Adriaans, P. and D. Zantinge, Data Mining, Addison-Wesley, New York, 1996.

17. Kasif, S., Datascope: Mining Biological Sequences, IEEE Intelligent Systems, (Nov/Dec 1999): 38-43.

18. Westphal, C. and T. Blaxton, Data Mining Solutions: Methods Solving Real-World Problems, John Wiley, New York, 1998.

19. Berson, A., S. Smith, K. Thearling, Building Data Mining Applications for CRM, McGraw-Hill, New York, 2000.

20. Weiss, S. M. and N. Indurkhya, Predictive Data Mining: a Practical Guide, Morgan Kaufman Publishers, San Francisco, 1998.

21. Jain, A. K., M. N. Murty, P. J. Flynn. Data Clustering: A Review, ACM Computing Surveys. 31, no. 3, (September 1999): 264-323.

22. Miyamoto S., Fuzzy Sets и Information Retrieval and Cluster Analysis, Cluver Academic Publishers, Dodrecht: Germany, 1990.

23. Everitt, B.S., Cluster Analysis, 2nd Edition, London: Heineman Educational Books Ltd, 1980.

24. Hartigan, J.A., Clustering Algorithms, New York: John Wiley & Sons, 1975.

25. Spath, H., Cluster Analysis Algorithms, Chichester, UK: Ellis Horwood, 1980.

26. Anderberg, M.R., Cluster Analysis for Applications, New York: Academic Press, 1973.

27. Hagan, M. T., H. B. Demuth, M. Beale, Neural Network Design, PWS Publishing Co., Boston, 1996.

28. White, H., Artificial Neural Networks: Approximation and Learning Theory, Oxford, UK: Blackwell, 1992.

29. Haykin, S., Neural Networks: A Comprehensive Foundation, Prentice

Hall, Upper Saddle River: NJ, 1999.

30. Zurada, J. M., Introduction to Artificial Neural Systems, West Publishing Co., St. Paul: MN 1992.

31. Pao, Y, Adaptive Pattern Recognition and Neural Networks, Reading, MA: Addison-Wesley, 1989.

32. Wasserman, P.D., Advanced Methods in Neural Computing, New York: Van Nostrand Reinhold, 1993.

33. Weiss, S.M. and Kulikowski, C.A., Computer Systems That Learn, San Mateo, CA: Morgan Kaufmann, 1991.