Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних систем управління
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка і дослідження ефективних методів ідентифікації об'єктів в системах прийняття рішень»

Виконав: магістр 2 курсу, групи 2АКІТ-21м
спеціальності 151- Автоматизація та комп'ютерно- інтегровані технології"
(шифр і назва напряму підготовки, спеціальності)

Вероніка ТОКАРЕНКО
(ім'я ПРІЗВИЩЕ)

Керівник: к.т.н., професор каф. КСУ
Микола БИКОВ
(ім'я ПРІЗВИЩЕ)
«16» грудня 2022 р.

Опонент: д.т.н.,професор каф. АІІТ
Роман КВЄТНИЙ
(ім'я ПРІЗВИЩЕ)
«17» грудня 2022 р.

Допущено до захисту
Завідувач кафедри КСУ
Вячеслав КОВТУН
(прізвище та ініціали)
«14» грудня 2022 року

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти ІІ-й (магістерський)
Галузь знань – 15 Автоматика та приладобудування
Спеціальність - 151 Автоматизація та комп'ютерно-інтегровані технології
Освітньо-професійна програма - Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
_____ Вячеслав КОВТУН
«03» жовтня 2022 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Токаренко Вероніці Михайлівні
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка і дослідження ефективних методів ідентифікації об'єктів в системах прийняття рішень»
керівник роботи Биков Микола Максимович, к. т. н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ВНТУ від "14" вересня 2022 року № 203

2. Строк подання студентом роботи "12" грудня 2022 року

3. Вихідні дані до роботи: опис навчальних прецедентів, представлених векторами в ознаковому просторі; методи ідентифікації – кластеризація і класифікація; сфера застосування – підтримка прийняття рішень на управління в комп'ютерно-інтегрованих системах управління виробництвом; середовища моделювання – Python, Java; Visual Studio 2020.

4. Зміст текстової частини: Анотація; Вступ; Аналіз стану проблеми і постановка задач; Математичний опис методів ідентифікації; Розробка алгоритмічного і программного забезпечення для моделювання; Експериментальна частина. Економічний розділ

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) 1. Схема класифікаційна методів машинного навчання – Плакат; 2. Алгоритм кластеризації за методом поверхневого натягу – Схема програми; 3. Класифікація даних методом SVM – Плакат; 4 Алгоритм класифікації за модифікованим методом SVM – Схема програми; 5 UML діаграма використання програмного забезпечення. 6 Результати машинного експерименту – Плакати.

6. Консультанти розділів роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 4	Небава М.І., професор кафедри ЕПВМ	Небава	Небава

7. Дата видачі завдання "_03_" __жовтня___2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Багатоваріантний аналіз методів машинного навчання, обґрунтування доцільності розробки	29.09.2022 р.	
2	Розробка математичного обґрунтування досліджуваного методу ідентифікації	15. 10.2022 р.	
3	Розробка алгоритмів функціонування системи	25. 10.2022 р.	
4	Розробка програмного забезпечення системи	10. 11.2022 р.	
5	Тестування програмного забезпечення	20.11.2022 р.	
6	Оформлення пояснювальної записки і графічного матеріалу	30.11.2022 р.	
7	Апробація результатів дослідження	07.12.2022 р.	
8	Публікації	10.12.2022 р.	
9	Графічні матеріали: Розробка схем алгоритмів Розробка плакатів демонстраційних Розробка UML-діаграм	05.12.2022 р. 10.12.2022 р. 12.12.2022 р.	
10	Остаточний захист	22.12.2022 р.	

Студентка _____ Вероніка ТОКАРЕНКО

Керівник роботи _____ Микола БИКОВ

АНОТАЦІЯ

УДК 681.142.2 (075.5)

Токаренко В. М. Розробка і дослідження ефективних методів ідентифікації об’єктів в системах прийняття рішень. Магістерська кваліфікаційна робота зі спеціальності 151 – автоматизація та комп’ютерно-інтегровані технології, освітня програма – інтелектуальні комп’ютерні системи. Вінниця: ВНТУ, 2022. 112 с.

На укр. мові. Бібліогр.: 44 назв; рис.: 15; табл. 5.

В магістерській кваліфікаційній роботі розроблено покращені методи кластерного аналізу і класифікації даних, що використовуються під час ідентифікації об’єктів для підтримки прийняття рішень в комп’ютерно-інтегрованих системах управління. Запропоновані метод кластеризації даних, який використовує фізичний принцип поверхневого натягу, і метод машинного навчання класифікатора типу SVM, модифікований на його основі, дозволили підвищити ефективність ідентифікації об’єктів і станів систем, описаних в ознаковому просторі. Надано математичне обґрунтування методу кластерного аналізу на основі методу градієнтного спуску.

У розділі програмного забезпечення розроблено алгоритми і програми ідентифікації даних запропонованими методами. Проведений машинний експеримент підтвердив адекватність розроблених методів і програмного забезпечення.

Розроблені методи і програмне забезпечення для ідентифікації об’єктів в системах управління дозволили підвищити такі показники ефективності прийняття рішень, як швидкість і якість.

Графічна частина містить 12 плакатів і креслень з ілюстрацією результатів роботи.

Ключові слова: системи управління, методи ідентифікації, кластерний аналіз, класифікація даних, прийняття рішень.

ANNOTATION

UDC 681.142.2 (075.5)

Tokarenko V. M. Development and research of effective methods of object identification in decision-making systems. Master's thesis on specialty 151 - automation and computer-integrated technologies, educational program - intelligent computer systems. Vinnytsia: VNTU, 2022. 112 p.

In Ukrainian. Bibliography: 44 titles; fig.: 15; table 5.

The master's thesis developed improved methods of cluster analysis and classification of data, which are used during the identification of objects to support decision-making in computer integrated control systems. The proposed method of data clustering, which uses the physical principle of surface tension, and the machine learning method of the SVM-type classifier, modified on its basis, made it possible to increase the efficiency of identification of objects and states of systems described in the feature space. A mathematical justification of the cluster analysis method based on the gradient descent method is given.

Algorithms and data identification programs using the proposed methods have been developed in the software section. The conducted machine experiment confirmed the adequacy of the developed methods and software.

The developed methods and software for data identification in control systems made it possible to increase such indicators of decision-making efficiency as speed and quality.

The graphic part contains 12 posters and drawings illustrating the results of the work.

Keywords: control systems, identification methods, cluster analysis, data classification, decision-making.

ВІДГУК

керівника магістерської кваліфікаційної роботи

студентки Токаренко Вероніки Михайлівни
(прізвище, ім'я, по батькові)

на тему: Розробка і дослідження ефективних методів ідентифікації об'єктів в системах прийняття рішень

Робота присвячена актуальній задачі розробки ефективних методів ідентифікації об'єктів в комп'ютерно-інтегрованих системах управління за рахунок використання удосконалених методів машинного навчання для підвищення ефективності прийняття рішень.

Студенткою було проведено аналіз і порівняння існуючих методів ідентифікації об'єктів і обгрунтовано вибір класифікаційного методу. В якості новизни роботи слід відзначити, що магістрантка запропонувала реалізувати його модифікованим методом SVM з опорою на розроблений нею метод кластеризації, що використовує принцип пошуку опорних точок за методом поверхневого натягу.

Всі прийняті в роботі рішення обґрунтовані на необхідному рівні на основі проведеного багатоваріантного аналізу і коректними математичними викладками.

Під час проектування магістерської роботи дипломниця проявила рівень теоретичних і практичних знань, достатній для розв'язання поставлених інженерних задач.

Більшість результатів магістерської кваліфікаційної роботи дипломниця отримала в результаті самостійного вирішення поставлених завдань

Етапи календарного плану виконання роботи виконувались в заплановані терміни.

До недоліків потрібно віднести те, що тестування розробленого програмного забезпечення було проведено на більшій кількості навчальних вибірок.

Висновок: магістерська кваліфікаційна робота Токаренко Вероніки Михайлівни *відповідає* спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку "А" (92 бали), а Токаренко В.М. автор *заслуговує* присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні комп'ютерні системи».

Керівник магістерської кваліфікаційної роботи

проф. каф. КСУ, к.т.н., доцент _____ Микола БИКОВ
(посада, науковий ступінь, вчене звання) (підпис) *Ім'я ПРІЗВИЩЕ*)

ВІДГУК

опонента на магістерську кваліфікаційну роботу

студентки Токаренко Вероніки Михайлівни
(прізвище, ім'я, по батькові)

на тему: Розробка і дослідження ефективних методів ідентифікації об'єктів в системах прийняття рішень

Подана на рецензію магістерська кваліфікаційна робота відповідає індивідуальному завданню і виконана у повному обсязі та у встановлений термін. Робота складається з наступних частин: аналіз стану проблеми ідентифікації об'єктів в системах автоматизації, математичне обгрунтування обраних методів ідентифікації об'єктів, програмна реалізація, економічний розділ, висновки, список використаних джерел, додатки.

Актуальність роботи викликана необхідністю підвищення ефективності ідентифікації виробничих ситуацій і об'єктів в комп'ютерно-інтегрованих системах управління, що дає можливість покращити прийняття управлінських рішень шляхом використання оптимальних стандартних алгоритмів управління для відомих ідентифікованих ситуацій і об'єктів.

В роботі запропоновано покращений метод кластерного аналізу даних про об'єкти, що ґрунтується на вперше запропонованому принципі поверхневого натягу, який дозволяє фіксувати граничні точки кластерів, а також модифіковано класифікатор типу SVM, орієнтований на використання знайдених точок в якості опорних.

Дослідження автором запропонованого методу за допомогою машинного експерименту дозволили визначити оптимальне значення порогу поверхневого натягу як умови закінчення процедури кластеризації.

Практична цінність роботи полягає в тому, що розроблений програмний продукт більш ефективний, оскільки розроблений кластеризатор знаходить поверхневі опорні точки кластерів для програми класифікації за алгоритмом SVM, що усунуло необхідність використання обчислювально-затратної процедури пошуку опорних точок за методом множників Лагранжа.

Зауваження до роботи: машинний експеримент з ідентифікації об'єктів бажано було б провести на більшій кількості еталонних файлів даних, а не тільки на файлі "iris.dat". Дане зауваження не є критичним, робота виконана на професійному рівні та її якість не викликає сумніву.

Магістерська кваліфікаційна робота виконана у відповідності з встановленими вимогами. Робота заслуговує оцінки «А», а її автор Токаренко Вероніка Михайлівна– присвоєння кваліфікації: ступінь вищої освіти магістр, спеціальність 151 – «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні комп'ютерні системи».

Опонент

професор каф. АІІТ, д.т.н., проф. _____ Роман КВЄТНИЙ
(посада, науковий ступінь, вчене звання) (підпис) (*Ім'я ПРІЗВИЩЕ*)

Печатка установи, організації опонента

ЗМІСТ

ВСТУП

Сучасний розвиток науки і техніки поставив на порядок денний необхідність автоматизації процесів обробки інформації для ідентифікації об'єктів на технологічних дільницях та під час оперативного прийняття рішень на адміністративному рівні сучасних комп'ютерно-інтегрованих систем автоматизації управління виробництвом і бізнесом. В більшості робіт з даної проблематики задача ідентифікації розв'язувалась з використанням лінійної апроксимації і критерія мінімуму квадратичного відхилення. Але практика застосування такої методики до об'єктів різної природи показала, що не завжди такий вибір рівняння апроксимації і критерія якості призводить до задовільних результатів. На сьогодні розв'язання цієї задачі пов'язано з впровадженням інтелектуальних методів в комп'ютерні системи управління, до яких відносяться, зокрема, методи машинного навчання (Machine Learning - ML). Машинне навчання - це підрозділ науки штучного інтелекту, присвячений розробці методів і алгоритмів опрацювання інформації для прийняття рішень на адміністративному і технологічному рівнях АСУП, які покращують якість управління на основі сумісного аналізу навчальних даних і отриманих результатів від виконання прийнятих рішень на управління. Перевагою застосування машинного навчання в комп'ютерно-інтегрованих системах управління є звільнення від рутинної нецікавої роботи осіб, що приймають рішення, звільняючи їм простір для творчості і розвитку.

Методи ML дозволяють розробляти ефективні системи прийняття рішень на адміністративному рівні підприємства для управління бізнесом, логістикою і транспортними перевезеннями, проектуванням і моделюванням виробництва продукції, а на технологічному рівні - розпізнавання виробничих ситуацій, зображень об'єктів різного типу та осіб за їх голосом.

В задачах прийнятті маркетингових рішень і розпізнавання образів, які посідають значний обсяг в технологічних та бізнес процесах сучасних комп'ютерно-інтегрованих виробництв, важливу роль відіграють такі методи

ідентифікації об’єктів за їх параметрами шляхом аналізу і класифікації даних, як кластеризація і класифікація. Найбільш розповсюдженими на сьогодні є такі методи кластерного аналізу даних, як метод кластеризації на основі щільності (DBSCAN), метод *k*- внутрішньо групових середніх (*k*-means), кластеризації з використанням гаусових моделей сумішей (GMM), метод середнього зсуву (Mean-Shift), методи ієрархічної агломеративної кластеризації (HAC). На сьогодні найбільш досконалим методом класифікації даних є метод опорних векторів (SVM). Даний метод за рахунок використання узагальнених вирішувальних функцій здатний оперувати не тільки з лінійними, а й з нелінійно розподіленими зображеннями об’єктів в просторі ознак.

Недоліком існуючих методів ідентифікації об’єктів за їхніми параметрами шляхом класифікації та кластеризації є відсутність такого зв’язку між ними, що орієнтує класифікатор на використання граничних точок кластерів з метою підвищення його ефективності. Тому задача розробки покращених методів кластеризації і класифікації даних, які використовуються під час ідентифікації об’єктів для підтримки прийняття управлінських рішень, що вирішується в даній магістерській кваліфікаційній роботі (МКР), є актуальною.

Метою даної роботи є збільшення ефективності систем прийняття рішень шляхом розробки ефективних методів ідентифікації об’єктів за рахунок застосування покращених методів класифікації і кластеризації.

Об’єктом дослідження є процеси аналізу даних в інтелектуальних комп’ютерних системах управління.

Предметом дослідження є методи, моделі та алгоритми ідентифікації об’єктів шляхом класифікації і кластеризації даних.

Зв’язок роботи з науковими програмами, планами, темами. Робота виконана згідно плану науково-дослідної роботи кафедри КСУ № 46К7 «Розробка методів і технологій автоматизації процесів контролю і керування»”.

Методи дослідження. У процесі дослідження застосовувалися: теорія інформації; методи машинного навчання; теорія алгоритмів; теорія ймовірностей; комп'ютерне моделювання для оцінки адекватності отриманих теоретичних результатів.

Наукова новизна отриманих результатів.

1. Вперше запропоновано метод групування подібних об'єктів в кластери з використанням принципу поверхневого натягу рідин.

2. Вперше запропоновано ідентифікувати об'єкти за їх параметрами шляхом використання модифікованого методу опорних векторів, який використовує координати поверхневих точок кластерів, отриманих на етапі кластеризації методом поверхневого натягу.

3. За результатами дослідження розроблено новий ефективний алгоритм ідентифікації об'єктів в системах прийняття рішень.

Практична цінність. Розроблені методи. моделі та програмне забезпечення дозволили підвищити ефективність управлінських рішень в комп'ютерно-інтегрованих системах управління виробництвом.

Впровадження. Результати МКР використовуються в навчальному процесі і наукових дослідженнях кафедри КСУ.

Достовірність теоретичних положень магістерської роботи підтверджується коректним застосуванням математичних методів, строгістю постановки задач, порівнянням результатів, отриманих за допомогою розроблених у роботі методів, з відомими, та співпадінням результів, отриманих в машинному експерименті під час тестування програмного забезпечення, з результатами математичного моделювання.

Особистий внесок здобувача. Автором роботи більшість результатів отримана самостійно.

Апробація результатів роботи. Основні положення й результати роботи доповідалися на міжнародній науково-технічній конференції КУСС-2022. Опубліковані тези доповіді: [Електронний ресурс]. - Режим доступу: https://conferences.vntu.edu.ua/index.php/mccs/mccs2022/paper/view/16530

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ ІДЕНТИФІКАЦІЇ ОБ’ЄКТІВ В СИСТЕМАХ АВТОМАТИЗАЦІЇ

1.1 Сучасний стан проблеми ідентифікації об’єктів

1.1.1 Суть задачі ідентифікації об’єктів в системах управління

Під поняттям ідентифікації розуміють процедуру ототожнювання об'єктів, встановлення відповідності об'єкта, що розпізнається, своєму образу [1]. Висловлюючись більш стисло, ідентифікація — це дія встановлення ідентичності. У техніці під ідентифікацією розуміють встановлення відповідності розпізнаваного предмета чи явища своєму образу.

Проблема ідентифікації в сучасних комп’ютерно-інтегрованих системах управління виникає в зв’язку з необхідністю аналізу виробничих ситуацій на їх адміністративному та розпізнавання образів (зорових і сигнальних) на технологічному рівнях. На адміністративному рівні такий аналіз проводиться для моделювання і прийняття рішень з ефективного управління комерційною та організаційною діяльністю підприємств, а на технологічному – для реалізації оптимального виробництва продукції.

Проблемі ідентифікації присвячена велика кількість робіт, які відрізняються як об’єктами ідентифікації, так і алгоритмами і методами її виконання [2, 3]. Безпосередньо самі алгоритми ідентифікації об’єктів залежать від їх природи, тому методи, які лежать в їх основі, пов’язані з вибором апроксимаційного рівняння, що використовує спостережувані дані, вибором критерія якості апроксимації, і відповідного йому методу оптимізації. Внаслідок цього виникли питання: як вибирати ефективний метод ідентифікації для об’єктів різної природи, який би забезпечував гарантоване отримання найкращого в заданих умовах рішення задачі ідентифікації. Для вибору методу ідентифікації об’єктів в інтегрованих системах управління виробництвом розглянемо їх класифікацію.

1.1.2 Огляд методів і способів ідентифікації об’єктів

Термін «метод» використовується для позначення системи прийомів і правил, що використовуються під час вивчення явищ природи, суспільства, мислення [2].

Під способом розуміють дію чи систему дій, які використовуються під час виконання роботи для досягнення поставленої мети [2].

Під час ідентифікації об'єктів застостовуються в більшості такі способи:

- уніфікованих найменувань;
- числових номерів;
- штрихового кодування;
- умовних позначок;
- класифікаційний;
- посилальний;
- описовий;
- описово-посилальний.

Найдавнішим із вказаних способів є **спосіб уніфікованих найменувань**. Для прикладу, найменування планет, морів, рік, озер, гір є унікальними і застосовуються в сполученні "ім'я — об'єкт" чи "об'єкт — ім'я" : Чорне море, Маріанська впадина, планета Марс, ріка Буг, озеро Свитязь, гора Арарат тощо. Ці надані унікальні найменування є ідентифікаторами і широко застосовуються в системах обробки інформації про ці об'єкти та у системах керування ними, наприклад: Новодністровська ГЕС, міжнародна космічна станція "Мир", товариство "Люстдорф".

Використання синтагми "об'єкт - ім'я" є обов'язковою умовою ідентифікації, оскільки одне і те ж ім'я може бути надане різним об'єктам: річка Дніпро, кафе "Дніпро", місто "Дніпро" і таке інше.

Досить широко для ідентифікації застосовується **спосіб числових номерів**, які надаються різним об'єктам. Наданий об’єкту номер у сполученні з його найменуванням дозволяє однозначно ідентифікувати об'єкт. На

практиці застосовуються два основних способи ідентифікації за допомогою цифрового номера: серійно-порядковий і порядковий.

Поширеним прикладом серійно-порядкового способу є ідентифікація дат. Наприклад, дата 17.12.2022 ідентифікує 17-те число грудня місяця 2022 року. Серійно-порядковий спосіб широко застосовується у вітчизняній паспортній системі для ідентифікації паспортів. В ньому кожен паспорт має у межах виділеної серії уніфікований шестизначний номер, наприклад, паспорт має серію АА і порядковий номер 628 284.

Порядковий номер, що привласнюється об'єкту, встановлює орган, який здійснює нумерацію. Наприклад, нумерацію будинків на вулиці встановлюють комунальні служби, автобусів — керівництво автотранспортним рухом, залізничних потягів - встановлює керівництво залізничним транспортом. Номери об'єктам часто надають у міру їх появи, обліку чи реєстрації: номер автомобіля в черзі на заправці, номер поштомату в «Новій пошті», нумерація шкіл у міру їх появи, табельний номер працюючого на деякому підприємстві по мірі його реєстрації, тощо.

Перевага порядкового номерного способу полягає в забезпеченні короткої і простої (за кількістю знаків) ідентифікації об'єкта. До недоліків можна віднести неінформативність, викликану відсутністю ознак, що характеризують об'єкти з наданими порядковими номерами. В якійсь мірі цей недолік усувається при використанні серійно-порядкового ідентифікаційного номера. Наприклад, використання цього методу ідентифікації для нумерації кімнат шляхом доповнення номера кімнати номером поверху: номери 2.13 і 4.13 означають кімнату номер 13 на другому поверсі та кімнату номер 13 на четвертому поверсі.

Інформаційно більш насиченою є ідентифікація способом з використанням порядкових номерів у поєднанні з класифікаційними угрупованнями, яка прийнята у ЗКП - Загальнодержавному класифікаторі продукції. Наприклад, для ідентифікації автомобіля вантажного ГАЗ 53А служить код ЗКП 45 1113 2111. В ньому є класифікаційна частина коду -перші

шість знаків 45 1113, яка дозволяє з класу 45 "Вироби автомобільної промисловості", до якого відносять також велосипеди, мотоцикли та автобуси, виділити конкретно автомобілі 451, вантажні 4511, загального призначення 45111, вантажопідйомністю від 1,5 т до 3 т (451113). Кінцеві чотири знаки коду 2111 включають номер, що надається в автомобільній галузі автомобілю з врахуванням числа ведучих коліс.

Спосіб штрихового кодування згідно з ДСТУ 3144—95 описується як присвоєння товарам умовних позначень у вигляді цифрового, літерно-цифрового та штрихового коду з метою їх ідентифікації.

Найбільшою цариною ефективного використання технологій штрихового кодування є споживчий ринок товарів і послуг. Основне функціональне призначення таких технологій полягає в ідентифікації товарів з використання в оперативному режимі актуальної інформації.

Під час ідентифікації продукції та документів широко використовується **спосіб умовних позначок**. В переважній мірі застсовуються такі три способи: мнемонічний, класифікаційний і мнемо-класифікаційний.

Мнемонічний спосіб умовних позначок дозволяє полегшує людині запам'ятовування та розуміння інформації про продукцію або документи. Так позначення "Електронасос ГНОМ 100—25" містить поряд з назвою об'єкта ("Електронасос") мнемонічне позначення, в якому літера Г означає для брудної води, літера Н — насос, літера О — одноступінчастий, літера М — моноблочний, число 100 — продуктивність 100 $м^3$/годину і число 25 — з тиском 25 мм.

Класифікаційний спосіб застосовується тоді, коли інформація опрацьовується в комп'ютерних системах. Наприклад, з його використанням побудована унікальна класифікаційна система позначення конструкторських документів і виробів, яка виглядає таким чином.

Чотиризначний літерний код організації-виробника (XXXX) надається згідно спеціального кодифікатора конструкторських організацій; код класифікаційної характеристики, (наприклад 303 125), який є інформаційною

частиною повідомлення, видається згідно класифікатора виробів і конструкторських документів машинобудування та приладобудування (Класифікатор Єдиної Системи Конструкторської Документації - ЕСКД). За такою класифікаційною характеристикою можна знайти, наприклад, інформацію про "групу однорідних виробів, що розроблювалися різними конструкторськими організаціями з метою запозичення", оскільки в коді класифікаційної характеристики міститься наступна інформація:

- 30 — вироби загального машинобудування; 303 — пристрої, що передають рух; 3031 — редуктори;

- 30311 — циліндричні одноступінчасті;

- 303115 — з міжосьовою відстанню пари зубчастих коліс від 63 до 315 мм включно.

Порядковий реєстраційний номер забезпечує ідентифікацію конкретного редуктора і надається згідно класифікаційної характеристики у межах коду організації-виробника від 001 до 999.

Класифікаційний спосіб зручно використовувати у випадках, в яких ідентифікуються групи однорідних, схожих за своїми описовими параметрами, об'єктів. Його перевага полягає в інформативності — він дозволяє зі значної кількості об'єктів виділяти саме ті, що мають певні ознаки. Він досить широко застосовується в багатьох сферах діяльності, оскільки забезпечує систематизацію об'єктів. Цей метод особливо ефективний для обробки інформації в системах управління, наприклад, якщо є необхідність зібрати інформацію про вантажні автомобілі в логістиці підприємства, види продукції, що користується найбільшим попитом або фактори шкідливих умов виробництва тощо.

Посилальний спосіб застосовується у тих випадках з метою ідентифікації об'єктів, в яких інформація про конкретні характеристики міситься в технічних чи нормативних документах. Він здебільшого використовується для визначення конкретної продукції під час її замовлення.

У ньому ідентифікація містить найменування виробу, умовне його позначення та посилання на документ, в якому описані визначені вимоги до виробу, наприклад:

- маргарин за ДЕСТ 240-85;
- соляна кислота за ДЕСТ 3118—77;
- бутилкаучук БК-1675У за ТУ 2294-010-17187505-95.

Якщо документ не надає ідентифікації якогось виробу, то в класифікацію додається інформація про виробника продукції, для прикладу: миючий синтетичний порошкоподібний засіб "Віка" згідно ТУ В6 - 00209651 Вінницькогонського ЗАТ "Реактив". Спосіб має той недолік, що не розкриває використанні основних характеристик та особливостей продукції.

У випадках, коли потрібноно ідентифікувати заданий об'єкт шляхом опису його характеристик (показників, властивостей, параметрів), як правило, використовується **описовий спосіб ідентифікації**. В такому способі однорідні об'єкти, які мають однакові найменування, близьку номенклатуру показників і сферу застосування можуть відрізнятися між собою значеннями цих показників. Даний спосіб ідентифікації передбачає використання усіх головних характеристик об'єкта і з їх допомогою відрізняє його від всіх інших однорідних об'єктів.

Описові методи ідентифікації широко застосовуються в закладах медицини у вигляді медичної карти пацієнта, у криміналістиці у вигляді опису злочинця та характеру злочину, у геології як опис мінералу тощо.

До однієї з головних переваг описового методу є можливість повести порівняльний аналіз об'єктів однорідного типу шляхом порівняння їх характеристик, які використовуються під час їхньої ідентифікації. При цьому порівняння можна проводити різними способами, зокрема, автоматизованим. При порівнянні встановлюється ступінь одинаковості чи відмінності об'єктів за їх основними характеристиками. При цьому приймається рішення про вибір, звичайно того об'єкта, який має найкращі показники для даних умов

застосування. Даний спосіб також використовується під час вибору однорідних об'єктів з метою їхньої стандартизації та систематизації.

Описово-посилальний спосіб на відміну від описового для ідентифікації використовує тільки частину характеристик об'єкта, при цьому посилається на документ, де наведені всі його основні характеристики.

Цей метод найбільш широко застосовується при створенні баз даних про різні об'єкти, наприклад, про інформаційні видання у вигляді каталогів, кадастрів, покажчиків, тощо. Він дозволяє суттєво скоротити обсяг інформації, потрібний для ідентифікації об'єктів, що істотно економить комп'ютерну пам'ять та скорочую обсяги видання. У каталогах приводять найменування продукції, її призначення та сферу використання, умовні позначення, що використовуються під час замовлення, позначення тих документів, які містять усі вимоги до даної продукції, адресу та найменування виробника, та основні характеристики з їх встановленими значеннями.

Як випливає з проведеного огляду методів ідентифікації об’єктів в найбільшій мірі для розв’язання вказаних раніще задач прийняття рішень на управління в комп’ютерно-інтегрованій системі управління підходять класифікаційний і описовий способи. Описовий спосіб дозволяє шляхом опису його характеристик (ознак) ідентифікувати заданий об'єкт, визначивши його схожість до інших об’єктів і сформувати з них групу (кластер) однорідних. Така процедура називається кластеризацією даних, а групи схожих між собою об’єктів – кластерами (іноді таксонами). При цьому віднесення нового об’єкта до одного з встановлених кластерів здійснюється класифікаційним способом за допомогою процедури класифікації, яка визначає, до якої групи об’єктів він найбільш похожий згідно встановленого критерію близькості.

Таким чином, для вирішення проблеми розробки ефективних методів ідентифікації об’єктів в системах управління проведемо вибір оптимальних для умов автоматизованого виробництва методів кластеризації та класифікації. Оскільки вони відносяться сучасною наукою про штучний

інтелект до методів машинного навчання, то проведемо аналіз сучасного стану цих методів.

1.2 Сучасний стан проблеми аналізу даних методами машинного навчання

1.2.1 Визначення і класифікація методів машинного навчання

В сучасних інтелектуальних системах управління переважним напрямком аналізу і опрацювання інформації для підтримки прийняття рішень є використання методів машинного навчання [3,4]. Переваги використання методів машинного навчання для підтримки прийняття рішень, а також основні області їх застосування в системах управління соціально-економічними системами та технічними уже були вказані у Вступі до даної роботи. Охарактеризуємо сучасний стан проблеми аналізу даних методами машинного навчання та їх особливості. Зокрема, розглянемо особливості існуючих на сьогодні методів кластеризації і класифікації даних з метою вирішення в магістерській роботі завдання покращення процедури ідентифікації об’єктів для підвищення ефективності прийняття рішень в комп’ютерно-інтегрованих системах управління.

Machine Learning, ML, або машинне навчання є розділом теорії штучного інтелекту, в якому здійснюється пошук методів вирішення різноманітних задач шляхом використання навчання під час вирішення подібних задач [5,6,7].

Побудова цих методів здійснюється з використанням засобів алгебри, дискретної математики, математичної статистики, чисельних методів, теорії оптимізації та інших розділів математики.

Машинне навчання останнім часом є головним напрямком теперішніх комп'ютерних технологій. Для фахівця з інтелектуальних технологій (ІТ) якраз знання основ і практичного застосування методів ML, окрім уміння програмувати, стає стандартом.

Терміном Mashine learning на разі визначають клас тих методів штучного інтелекту, для яких характерно навчання в процесі застосування розв’язків множини схожих задач, а не пряме рішення задачі [8,9,10]. Таким чином, машинне навчання полягає в отриманні знань з даних. Це область науки, що знаходиться на перетині штучного інтелекту, комп'ютерних наук та статистики, вона також відома як статистичне навчання або прогнозна аналітика . Оскільки поняття штучного інтелекту здебільшого застосовується під час розв’язання задачі розробки інтелектуальних роботів, то термін «машинне навчання» (ML) на сьогодні є більш поширеним, ніж словосполучення «штучний інтелект». Застосування методів машинного навчання стало звичним явищем не тільки в комп’ютерних інтелектуальних системах управління, а й в повсякденному житті. Велика кількість сучасних веб-застосунків і пристроїв широко використовують алгоритми машинного навчання, починаючи з простого випадку автоматичних рекомендацій з покупки продуктів чи замовлення, перегляду фільмів, і закінчуючи складними процедурами розпізнаванням осіб на фотографіях і персоналізованими онлайн-радіотрансляціями. В роботі таких складних сайтів як Facebook, Netflix або Amazon можна побачити, наявність якихось моделей машинного навчання в їхніх розділах.

На сьогодні методи машинного навчання використовують не тільки в управлінській діяльності підприємств і комерційних за стосунках, а й в наукових дослідженнях, пов’язаних з обробкою даних [11,12] До таких наукових проблем можна віднести дослідження космосу, відкриття нових частинок, аналіз ДНК, діагностику захворювань в медицинській практиці.

Навчання можна розділити два типи:

1. Дедуктивне, що використоіує формалізацію експертних знань та їх занесення в комп'ютерну систему прийняття рішень у формі бази знань.

2. Індуктивне навчання, або навчання по прецедентах, яке грунтується на знаходженні закономірностей, що існують в даних.

Перший тип навчання використовується в експертних системах, в зв’язку з чим під терміном Machine Learning на разі розуміють навчання на прецедентах.

Сформулюємо проблему навчання з використанням прецедентів таким чином. Існує набір об'єктів чи ситуацій або явищ та низка альтернативних відповідей (реакцій) [13]. Прогонозується, що між об'єктами та відповідями існує залежність, яка апріорі є невідомою. Встановлено тільки визначей набір прецедентів - пар «об'єкт, реакція», названий навчальною вибіркою. Грунтуючись на цих даних треба встановити неявну залежність за допомогою побудови алгоритму, здатного достатньо точно прийняти рішення про його класифікацію для всякого можливого вхідного об'єкта. Методи ML дозволяють реалізувати емпірично сформоване рішення там, де залежність аналітично не обов'язково виражається. Тут є важливою властивість навченої системи до узагальнення, тобто правильної реакції на ті дані, які до навчальної вибірки не належать. Для визначення адекватності розроблених моделей відповідей використовується вибраний критерій якості.

З однієї сторони розділ Machine Learning історично увібрав в себе методи математичної статистики, а з іншої сторони завдячує поділу нейромережевих технологій на два напрямки: топології архітектури нейромереж та методів їх навчання [14]. На сьогодні відомі способи класифікації методів навчання випливають лише зі способу використання нейромереж і на теперішній час є застарілими. Класифікацію метолів ML потрібно виконати для того, щоб мати можливість орієнтуватися у виборі потрібного методу для розв’язання конкретно поставленої задачі. На рис. 1.1 наведено один із альтернативних варіантів класифікації відомих на теперішній час методів машинного навчання. Зі схеми рис. 1.1 витікає, що поставлені в роботі задачі розробки ефективних методів і кластеризації та класифікації даних відносяться до класичного навчання.

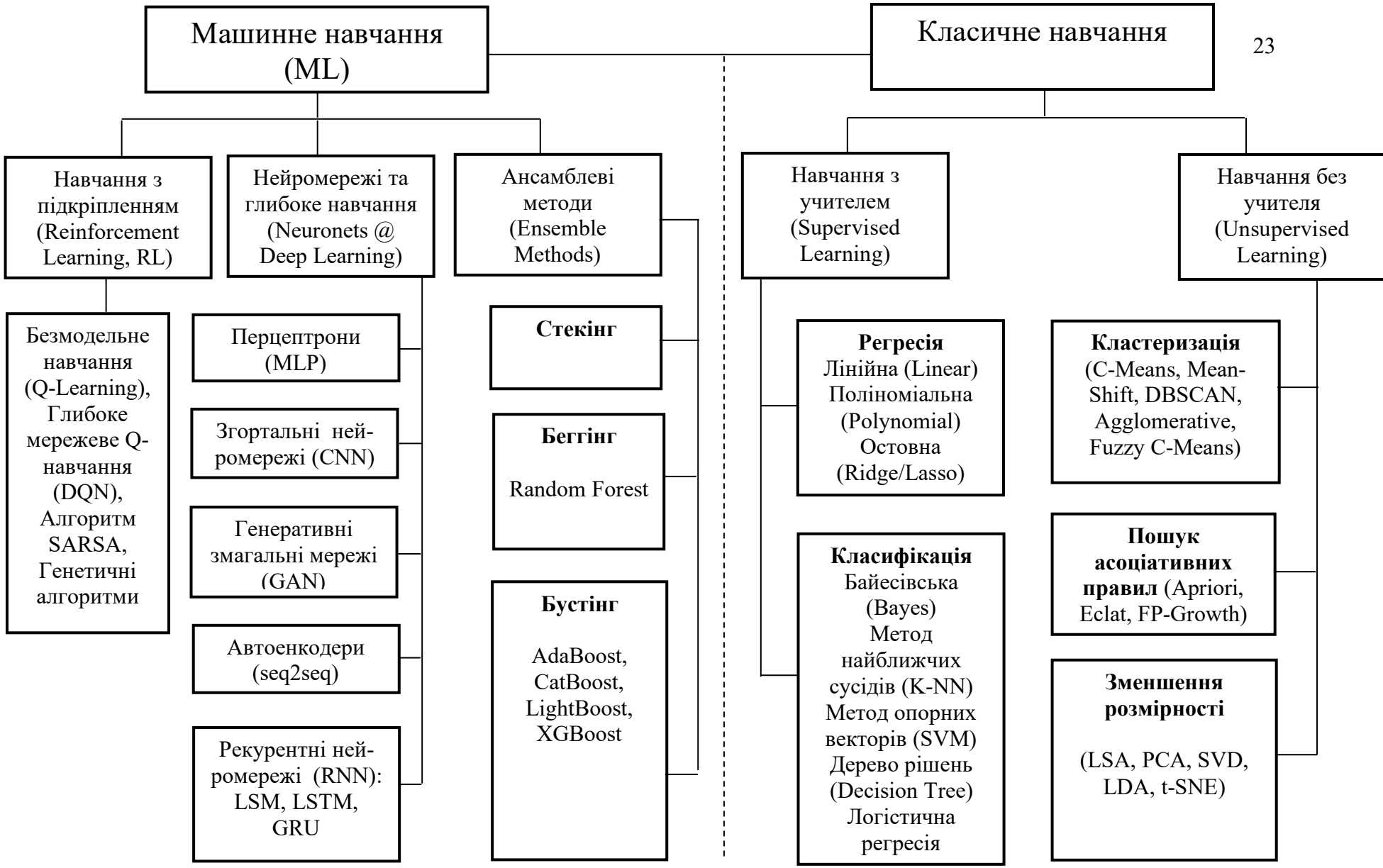


Рисунок 1.1 – Класифікаційна схема методів машинного навчання

Методи класичного навчання відрізняються від методів суто машинного навчання тим, що для їх реалізації потребують традиційне програмування. Розглянемо відмінності між ними більш детально.

1.2.2 Аналіз відмінностей між класичним і машинним навчанням

Для уяснення того, як здійснити реалізацію ефективної ідентифікації об'єктів в системах управління за допомогою запропонованих методів кластеризації і класифікації наявних в розробника даних про ці об'єкти, проаналізуємо їх відмінності від методів суто машинного навчання [15,16].

Основна відмінність класичного навчання від суто машинного полягає в необхідності використання тля його реалізації традиційного програмування. При цьому слід відзначити, що машинне навчання не завжди може замінити традиційне програмування. Наприклад, створити сайт інженеру веб-розробнику за допомогою алгоритмів машинного навчння неможливо. Тому машинне навчання і штучний інтелект використовують в якості доповнення до стандартних інструментів програмування. Для прикладу, для системи трейдингу прогнозний алгоритм можна розробити з використанням машинного навчання, а інтерфейс, візуалізацію даних та інші частини реалізовують однією з мові програмування (Java, Python, R та інш).

Тому машинне навчання як правило використовують там, де стандартні методи програмування для вирішення задачі не є ефективними.

Щоб отримати рішення задачі шляхом традиційного програмування, інженеру потрібно розробити алгоритм і написати програму. Після цього він задає початкові дані програми і отримує результат. Отже під час рішення задачі традиційним програмуванням інженер-розробник сам створює алгоритм розв'язання, який вводить набір параметрів і на підставі вхідних даних має можливість спрогнозувати результат роботи моделі.

Оскільки людині-розробнику дуже складно працювати з великим набором параметрів, тоді вона може за допомогою традиційного програмування побудувати тільки дуже просту модель.

Під час використання суто методу машинного навчання для вирішення тієї ж задачі інженер-розробник використовує інший підхід. Алгоритм вирішення задачі при цьому ним самостійно не розробляється, а спочатку збирається навчальна вибірка апріорних даних, за тим її використовують для побудови моделі машинного навчання.

Зібрана представницька вибірка навчальних даних завантажується в різні алгоритми машинного навчання і використовуються різні «регулятори», щоб налагодити алгоритм навчання на отримання різних моделей. З отриманих моделей вибирається та, яка володіє узагальнюючими властивостями, тобто може створювати адекватні прогнози при поданні на вхід уже нових даних, які не входили до навчальної вибірки. Використання готових моделей у випадку машинного навчання подібне до процесу, що має місце в стандартному програмуванні, але замість алгоритму використовується готова розроблена модель. Модель, отримавши вхідні дані, виробляє результат. Якраз використання готової відомої моделі є недоліком машинного навчання в тому випадку, коли потрібно реалізувати суттєво нову ідею в моделі обробки даних. Оскільки в нашій магістерській кваліфікаційній роботі поставлена задача розробки більш ефективних, порівняно з існуючими, методів ідентифікації об’єктів, за рахунок нової ідеї кластерного аналізу методом поверхневого натягу і відповідної до нього модифікації методу класифікації типу SVM, то в даній роботі ми обираємо шлях вирішення цієї задачі покращеними методами класичного навчання з використанням традиційного програмування.

1.3 Аналіз методів класифікації образів

1.3.1 Класифікація об’єктів за принципом мінімуму відстані

Класифікація образів за допомогою функцій відстані — одна з перших ідей автоматичного розпізнавання образів. Цей простий метод класифікації

виявляється дуже ефективним інструментом при рішенні таких задач, у яких класи характеризуються ступенем мінливості, обмеженої в розумних межах.

У ситуаціях, коли вектори вимірів (образи) у кожнім класі будуть майже ідентичні, класифікатори, що діють за принципом мінімальної відстані, можуть виявитися надзвичайно ефективним засобом рішення задачі класифікації [17, 18, 19]. Найбільш наглядним способом введення міри подібності для векторів об‘єктів, які зображуються точками в просторі ознак, є визначення відстані між ними. Наприклад, на рис.1.2,а) видно, що інтуїтивно точку $\vec{x}$ можна віднести до класу Ω_1 виключно з тих міркувань, що цей вектор знаходиться ближче до точок цього класу.

Використання відстані для класифікації дає задовільні практичні результати тільки в випадку компактного розміщення кластерів (рис.1.2,а). В іншому випадку (рис. 1.2,б) зарахування зображень в один із класів на грунті оцінки його близькості до об‘єктів відповідного класу викликає ускладнення.

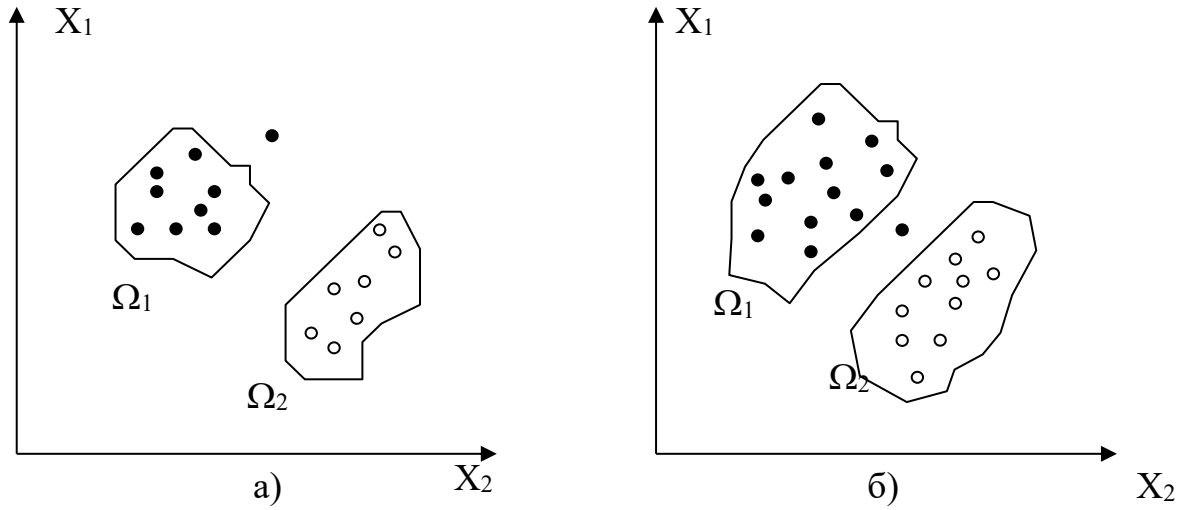


Рисунок 1.2 – Класифікація об‘єктів за принципом мінімуму відстані: а – сприятливе розміщення об’єктів *x*; б – несприятливе

1.3.2 Байесівська класифікація образів

Байесівський підхід до класифікації образів передбачає побудову класифікатора на основі статистичних властивостей класів [20, 21, 22]]. Статистичний аналіз надає можливість побудувати правило класифікації, яке

є оптимальним з точки зору гарантування в середньому найменшої імовірності здійснення помилки класифікації. Це оптимальне в статистичному сенсі правило класифікації слугує зазвичай стандартом, з яким почасти здійснюється порівняння якості інших класифікаційних алгоритмів.

Байєсівський класифікатор забезпечує віднесення образа ***x*** до класу з найменшим значенням середніх втрат ***r***. Тому образ ***x*** зараховується в клас ω_i , якщо виконується умова $r_1(x) < r_2(x)$; це повинно означати, що

$$L_{11}p(x|\omega_1)p(\omega_1)+L_{21}p(x|\omega_2)p(\omega_2)<L_{12}p(x|\omega_1)p(\omega_1)+L_{22}p(x|\omega_2)p(\omega_2), \qquad (1.1)$$

чи, що те ж саме,

$$(L_{21}-L_{22})p(x|\omega_2)p(\omega_2) < (L_{12}-L_{11})p(x|\omega_1)p(\omega_1). \qquad (1.2)$$

У формулах (1.1) і (1.2) елемент матриці L_{ij} визначає втрати від прийнятого рішення на класифікацію *i*-го об'єкта як *j*, p(x|ω_i) - функція правдоподібності для класу ω_i, $r_i(x)$ - середні втрати для пред'явленого образа *x*, $r_j(x)=\sum_{i=1}^{M}L_{ij}p(x|\omega_i)p(\omega_i)$. $p(\omega_i)$ – ймовірність появи образів класу ω_i.

Перевагою байесівського методу є те, що є можливість оцінки ймовірності помилки для кожного рішення. До недоліком потрібно віднести потребу в деякій попередній інформації, а саме знання платіжної матриці втрат від прийняття відповідних рішень та апріорних ймовірностей класів і функцій щільностей ймовірності розподілу ознак для кожного з них.

1.3.3 Класифікація а допомогою дерева рішень

Одним із широко розповсюджених методів прийняття рішень є використання моделі дерева рішень. Під "деревом рішень" розуміють графічне зображення послідовності рішень відносно станів навколишнього

середовища із зазначенням на кожному кроці (у вузлах дерева) умов і прийнятих відповідних рішень і іноді і виграшів для них (на гілках дерева, що виходять з вузлів для всіх можливих комбінацій альтернатив і станів середовищ. Приклад дерева рішень наведено на рис. 1.3.

Одним із поширени методів прийняття рішень є використання моделі дерева рішень. «Дерево рішень» розуміється як графічне представлення серії рішень, пов’язаних із станом навколишнього середовища, із зазначенням на кожному кроці (у вузлах дерева) умов і відповідних прийнятих рішень, а іноді і результатів для них (на гілках дерева). Рисунок 1.3 показує приклад структури дерева, що містить вузли і гілки для усіх можливих комбінацій альтернатив.

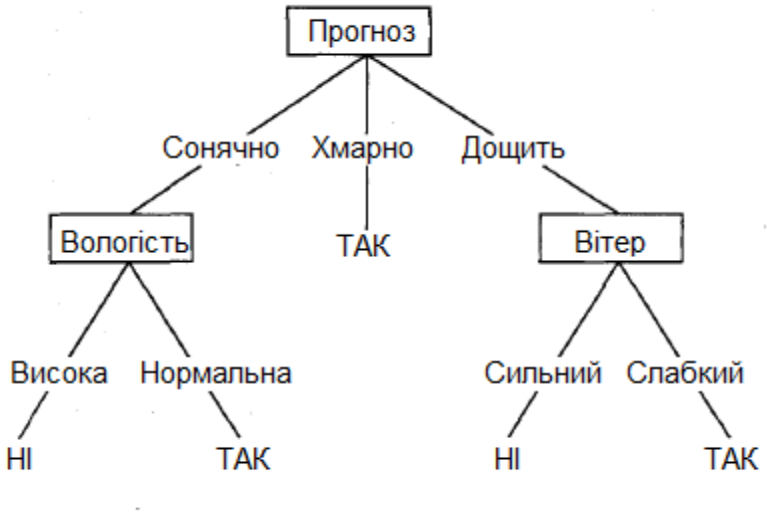


Рисунок 1.3 – Дерево прийняття рішення “Грати в теніс”

Під час побудови вказаної моделі дерева використовують його навчання на навчальній вибірці даних, опис яких залежить від природи предметної області, для якої вирішується задача. Отримана модель навченого дерева рішень відповідає деякій дискретній цільовій функції, а сам процес навчання реалізується відповідн визначенню Т. Мітчелла [23]:

[*"Definition:* A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E* "].

Іншими словами: кажуть, що комп'ютерна програма навчається при розв'язанні задачі, вибраної з набору класів задач *T*, якщо її ефективність *P* підвищується з накопиченням досвіду *E* відповідно до обраного критерію *P*.

До найпопулярніших класів задач *T* у машинному навчанні відносяться вже розглянуті в попередніх розділах:

- Класифікація - віднесення об'єктів до однієї з груп (класів) на основі їх характеристик;

- Регресія - прогнозування кількісних характеристик об'єкта на основі інших характеристик об'єкта;

- Кластеризація – поділяє набір об'єктів на групи на основі характеристик цих об'єктів таким чином, щоб об'єкти всередині групи були схожі один на одного, а об'єкти за межами тієї самої групи менш подібні.

Окрім названих, поширеним класом задач є знаходження аномалій в наборах даних, тобто об'єктів або їх груп, сильно несхожих на інші в цьому наборі (це так званий клас методів "Data Mining"), та багато інших [24].

На додаток доописаних вище, розповсюдженим класом завдань є виявлення аномалій у наборах даних, тобто об'єктів чи груп об'єктів, які дуже відрізняються від інших у цій колекції (це так звані методи «інтелектуального аналізу даних Data Mining»), і багато інших [24].

Досвід *E* алгоритмом навчання отримує шляхом аналізу впливу даних на одержаний результат навчання. Алгоритми машинного навчання як було попередньо визначено можна поділені на навчання з учителем (контрольоване навчання) і навчання без учителя (неконтрольоване навчання) в залежності від використанняу навчальній вибірці маркованих чи немаркованих даних відповідно. Тобто в контрольованому навчанні (supervised lerning) належність до того чи іншого класу для кожного об'єкта навчальної вибірки відома, в неконтрольованому навчанні (unsupervised learning) така інформація невідома.

В практичному прикладі в завданні прогнозу неповернення чи повернення кредиту деяким позичальником під досвідом *E* будеє отримана з попередньої кредитної історії навчальна вибірка з таких даних: набір позичальників, описаних таким набором ознак як тип кредиту, вік, зарплата і інші, а також цільовим класом, який описується фактом неповернення (0) або повернення (1) кредиту. В даному прикладі вирішується задача бінарної класифікації. У випадку наявності інформації про час відтермінування клієнтом повернення кредиту на основі чого теж саме треба спрогнозувати для нових клієнтів, вирішується задааі регресії.

Ще однією, третьою складовою у наведеному визначенні машинного навчання є критерій (метрика) *P* оцінки ефективності алгоритму. Різні задачі і відповіднїм алгоритми використовують різні критерії, вигляд яких залежить від опису вирішуваної проблеми. З них найпростішою метрикою якості класифікаційного алгоритму точність (accuracy), тобто частка правильних прогнозів даного алгоритму на тестовій вибірці.

Оцінка якості класифікаційної моделі у вигляді дерева рішень, характеризується основними двома показниками: точністю розпізнавання і помилкою.

Точність розпізнавання визначається відношенням правильно класифікованих в процесі навчання об'єктів до їх загальної кількості у навчальній вибірці..

Помилка розраховується відношенням неправильно класифікованих в процесі навчання об'єктів до їх загальної кількості у навчальній вибірці.

Як правило, навчання моделі дерева прийняття рішень підходить найкраще для проблем з такимми характеристиками:

- Об’єкти, що класифікуються, описані у вигляді ознака-значення, при цьому використовується фіксований набір ознак (наприклад, на рис. 1.3 Вологість) і їх значеннями (наприклад, Висока). В найпростішому випадку навчання дерева рішень кожна ознака має невелику кількість розбіжних можливих значень (наприклад, Висока, Нормальна, Середня). При цьому є

можливість використати ті розширення базового алгоритму, що дозволяють працювати з реальними величиниами ознак (для прикладу, числове представлення вологості).

- Функція цілі чи критерій мають дискретні вихідні значення. На рис. 1.3 дерево рішень надає логічну класифікацію Так чи Ні кожному прикладу. Моделі дерева рішень також зручно розповсюдити на функції з більш ніж двома дискретними вихідними значеннями. Суто більше розширення дає можливість навчати цільові функції з реальними величинами, хоча застосування дерев рішень у такому випадку є менш поширеним.

- В тих випадках, де потрібне використання диз'юнктивних ознакових описів. Таким чином, дерева рішень можна представляти диз'юнктивними логічними виразами.

- Для випадків, коли дані для навчання містять помилки, методи навчання дерева рішень дають можливість усунути помилки у класифікації навчальних об’єктів і помилки в значеннях їх ознак.

- В тих випадках, в яких навчальні дані мають відсутніми значення деяких ознак, використання моделі дерева рішень не приводить до негативних результатів.

Багато практичних завдань відповідають вказаним характеристикам. Навчання за допомогою моделі дерева прийняття рішень використовується до розв’язання таких проблем, як визначення несправності обладнання за відомою причиною, класифікації медичних пацієнтів за історіями їх захворювань, визначення позичальників кредитів за ймовірністю сплати платежів. Проблеми, в яких треба класифікувати приклади в одну з можливих категорій дискретного набору, називають проблемами дискретної класифікації. Оскільки в системах управління ідентифікуються здебільшого об’єкти з неперервними значеннями параметрів, що їх описують, то даний метод для вирішення нашої проблеми не підходить.

1.3.4 Класифікація образів з використанням опорних векторів

Машину опорних векторів (Support Vector Mashine - SVM) на сьогодні визнана одним із найкращих класифікаторів, що дозволяє реалізувати і лінійну і нелінійну класифікацію [25, 26].

Метод опорних векторів (SVM) є алгоритмом контрольованого навчання, і він використовується в більшості для завдань класифікації. У цьому алгоритмі кожен елемент даних зображається точкою в *n*-вимірному просторі (де *n* - кількість ознак), при цьому величина кожної ознаки задає значення відповідної координати. SVM здійснює класифікацію, використовуючи гіперплощину, що розділяє класи оптимальним способом. *Опорними векторами* називаються ті об'єкти вибірки, які лежать на границях класів [27, 28, 29]. Розподіл між класами буде оптимальним у тому випадку коли відстань між граничними гіперплощинами, які лежать на опорних точках, буде максимальною (рис.1.4).

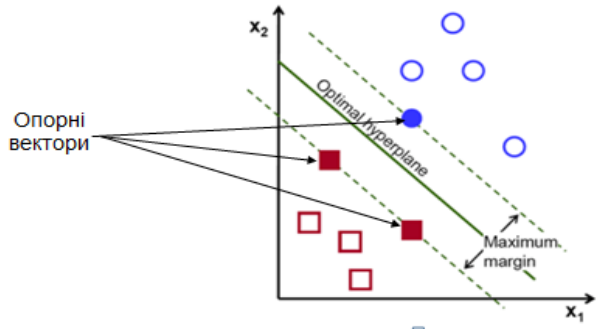


Рисунок 1.4 – Геометрична ілюстрація методу опорних векторів

Навчання класифікатора методом опорних векторів зводиться до задачі оптимізації, що розв’язується застосуванням евристичних алгоритмів. Недоліком цього методу є обчислювальна складність знаходження опорних точок, що є кінцями тих векторів, через які проводяться граничні для класів гіперплощини, для яких будується розподільна площина. Пошук опорних точок здійснюється шляхо розв’язанням оптимізаційної задачі з використанням Лагранжіана такого вигляду [30, 31].

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j \qquad (1.3)$$

за умов $\sum_i \alpha_i y_i = 0$,

де α_i - множники Лагранжа, x_i - вектори навчальної вибірки, y_i=±1 – індекси належності точок навчальної вибірки до першого чи другого класів.

Пошуку оптимального значення функції L_D шляхом багатократного обчислення скалярного добутку векторів породжує його недолік, що полягає в обчислювальній складність методу SVM.

Проведений в даному розділі огляд методів класифікації, що використовуються для ідентифікації об’єктів в системах прийняття управлінських рішень, показав, що на сьогодні найбільш досконалим є класифікація за алгоритмом опорних векторів SVM. Оскільки для навчання класифікатора даного типу потрібно реалізувати складну оптимізаційну процедуру для виявлення граничних точок в кластерах, яка є неефективною з точки зору обчислювальних затрат, то усунути цей недолік можна шляхом розробки методів кластерного аналізу, які фіксують поверхневі точки кластера під час навчання [32]. Проаналізуємо особливості найкращих методів кластерного аналізу, відомих на сьогодні.

1.4 Огляд методів кластерного аналізу

Задача кластерного аналізу даних полягає в об’єднанні деякої множини об’єктів в групи (кластери) так, аби об’єкти одного кластера були більш похожі згідно визначеної міри подібності один на іншого, ніж на об’єкти в інших кластерах. Завдання кластерного аналізу відносяться до “навчання без учителя”, тобто до неконтрольованого навчання.

Завдання кластеризації можна сформулювати наступним чином.

Задана навчальна вибірка X і міра відстані ρ між об'єктами. Треба розподілити задану вибірку X на групи об’єктів (кластерів) таким чином, щоб

до кожного кластера входили об'єкти, близьких по метриці ρ, а об'єкти в різних кластерах істотно відрізнялися. Кожен об’єкт при цьому маркується міткою класу, тобто для всякого $x_i \in X$ приписується номер кластера y_i.

До найбільш популярних на сьогодні алгоритмів кластеризації, відноситься такі, як метод кластеризації за середнім зсувом (*k*-medians), метод *k*-внутрішньогрупових середніх, просторова кластеризація даних з шумом на основі щільності **(** DBSCAN **)**, кластеризація з використанням моделей Гаусової суміші (GMM), агломеративна ієрархічна кластеризація [33, 34].

Проаналізуємо переваги і недоліки на сьогодні найбільш популярних методів кластеризації [35, 36, 37].

1.4.1 Кластеризації даних за середнім зсувом (*k*-medians)

Кластеризація середніх зсувів - це алгоритм, який на основі ковзного вікна намагається визначити щільні області точок даних. Подібно до описаного нижче методу *k*- means, він грунтується на центроїді, і має за мету знаходження центральних точок кожного классу. Він працює шляхом послідовної зміни кандидатів на центральні точки таким чином, щоб вони були середнім значенням точок у розширюваному вікні. Затим утворені вікна-кандидати на етапі подальшої обробки фільтруються для усунення близьких дублікатів, і утворюють кінцевий набір центральних точок та відповідні їм групи близьких до них точок.

Реалізація методу здійснюється алгоритмом методу в такий спосіб.

1. Вибирається за ядро з радіусом *r* зпочаткове кругове розширюване вікно з центром у деякій точці *C* (яка вибирається випадковим чином). Середній зсув реалізується як алгоритм сходження по схилу, що передбачає послідовнене переміщення даного ядра в напрямку більш високої щільності на кожному ітераційному кроці до злиття.

2. В кожному ітераційному кроці ковзне вікно зміщується у напрямку областей з більш високою щільністю, при цьому розраховуючи центральну точку як середнє значення точок в даному вікні. Щільність всередині

роширюваного вікна є пропорційною кількості точок в ньому. Зміщуючись до середнього значення точок, вікно поступово рухатиметься в напрямку областей точок більшої щільності.

3. Зсув розширюваного вікна здійснюється згідно з середнім значенням до тих пір, поки не буде визначено напрямку, в якому зміщення може вмістити всередині ядра більше точок.

4. Крок з 1 по 3 виконуються пробно з кількома розширюваними вікнами, аж поки всі точки не увійдуть до вікна. З декількох розсувних вікон, що перекриваються зберігається одне вікно, що містить найбільшу кількість точок. Точки даних групуються відповідно до того розсувного вікна, де вони знахояться.

Перевагою методу порівняно з методом *k*-середніх, є те, що в ньому немає необхідності задавати кількість кластерів, оскільки їх автоматично виявляє середній зсув. Недоліком є початкова невизначеність у виборі розміру радіуса *r* вікна.

1.4.2 Метод *k*– внутрішньо групових середніх

Алгоритм методу *k*-means можна описати таким чином:

1. На першому кроці задається кількість кластерів і випадковим чином вибирають їхні центральні точки. Кількість потрібних кластерів визначається з умов задачі, або шляхом застосування нескладного алгоритму кластеризації, наприклад, методу максимінної відстані.

2. На другому етапі для кожної точки з заданого набору обчислюється відстань до кожного з вибраних центрів кластерів. Точка відносять до того центру, до якого вона найближча.

3. На третьому кроці всі точки, зараховані до визначених центрів, використовують для обчислення координат нового центру, знаходячи середнє значення координат цих точок.

4. Вказані кроки повторюють до тих пір, поки не буде отримана задана необхідна точність, або доки центри кластерів не перестануть змінюватися в черговій ітерації.

Перевагою даного методу є простота його реалізації. Основними недоліками є те, що необхідно знати наперед кількість кластерів і те, що він не фіксує поверхневі точки кластеру, координати яких можуть бути корисними в алгоритмах класифікації.

1.4.3 Просторова кластеризація даних з шумом з використанням щільності (DBSCAN)

1. Алгоритм методу DBSCAN починає з довільної точки даних. До цієї точки задають за допомогою відстані епсилон ε деякий окіл. Точки, що знаходяться в заданому околі ε, є точками сусідства.

2. Якщо в цьому околі є достатня відповідно до заданого параметру minPoints кількість точок, то алгоритм розпочинає процес кластеризації, і першою точкою в новому кластері стає поточна точка даних. В протилежному випадку ця точка буде позначена як шум. І в тому і в іншому випадках точка позначається як “відвідана”.

3. Для знайденої першої точки в новому скупченні точки, що лежать в межах відстані ε, теж стають частиною цього скупчення. Після цього процедура, згідно з якою всі точки в околі є входять до одного кластера, повторюється для всіх тих нових точок, які щойно були додані до групи кластерів.

4. Процес для кроків 2 і 3 повторюється до тих пір, доки всі точки в кластері не будуть визначені, що означає, що всі точки в межах околу є кластера були відвідані та позначені.

5. Після закінчення роботи з поточним кластером, береться нова невідвідана точка і обробляється, і алгоритм виявляє наступний кластер чи шум. Процес повторюється до тих пір, доки всі точки не будуть позначені як

відвідані. Оскільки після того, як всі точки будуть відвідані, кожна з них позначаєься як та, що належить до кластеру, або як шум.

Алгоритм DBSCAN має ряд великих переваг. Перша полягає в тому, що не потрібно задавати певну кількість кластерів. Наступна полягає в тому, що він також приймає викиди за шуми, що відрізняє його від алгоритму від середнього зсуву, який відносить їх до кластера, навіть якщо ця точка даних сильно відрізняється. Ще одна перевага в тому, що він здатний досить добре знаходити кластери довільної величини та різної форми.

Головним недоліком алгоритму є те, що він працює гірше інших для кластерів з різною щільність. Це пояснюється тим, що вибір порогу відстані ε та параметра minPoints кількості точок для визначення точок сусідства буде змінюватися залежно від зміни щільності точок в кластері [38,39].

1.5 Постановка задач

Проведенийв даному розділі аналіз використання сучасних методів машинного навчання для вирішення задачі ідентифікації об'єктів шляхом класифікації і кластеризації даних в системах підтримки прийняття управлінських рішень показав, що їхня ефективність може бути підвищена. Для розв'язання цієї проблеми в даній магістерській кваліфікаційній роботі ставляться і вирішуються наступні задачі:

1. Розробити для ідентифікації об'єктів ефективний метод кластеризації даних, який здатний під час кластеризації фіксувати граничні точки кластера.

2. Модифікувати обраний в результаті попереднього аналізу алгоритм класифікації даних типу SVM для підвищення ефективності його роботи шляхом використання в ньому опорних точок шляхом їх визначення серед граничних точок кластерів.

2 МАТЕМАТИЧНЕ ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ

2.1 Математичне обґрунтування методу класифікації

2.1.1 Опис класифікації даних методом SVM

В результаті попередньо проведеного аналізу в якості класифікатора даних, що використовується для ідентифікації об'єктів, було обрано метод опорних векторів SVM. Він широко використовує при розв'язанні завдань регресійного аналізу даних та розпізнавання образів. В інтелектуальних комп'ютерних системах управління розпізнавання образів використовується для ідентифікації спостережуваного об'єкта чи технологічної ситуації з метою виконання алгоритмівд, призначених для обробки відомих категорій ситуацій чи об'єктів. В даному застосуванні завдання класифікації образів є центральним, для його вирішення використовується цілий ряд методів і алгоритмів. Як було вказано в розділі 1 обраний метод опорних векторів є одним з найдосконаліших сучасних алгоритмів класифікації, оскільки може працювати як лінійний, так і нелінійний класифікатор при умові застосування відповідних функцій ядра для нелінійного випадку [25, 26].

Алгоритм реалізації методу опорних векторів SVM є дискримінантним класифікатором, робота якого математично ґрунтується на понятті розподільної гіперплощини [27, 28]. Для отриманої навчальної вибірки маркованих даних він будує оптимальну гіперплощину, яка класифікує найкращим способом з точки зору точності нові прецеденти об'єктів. Простий приклад, показаний на рис. 2.1 надає наглядну можливість розуміння оптимальності побудованого у вигляді гіперплощини класифікатора. Для випадку лінійно розподільної множини точок на площині показано принцип пошуку прямої, яка розподіляє два класи найкращим чином [29, 30].

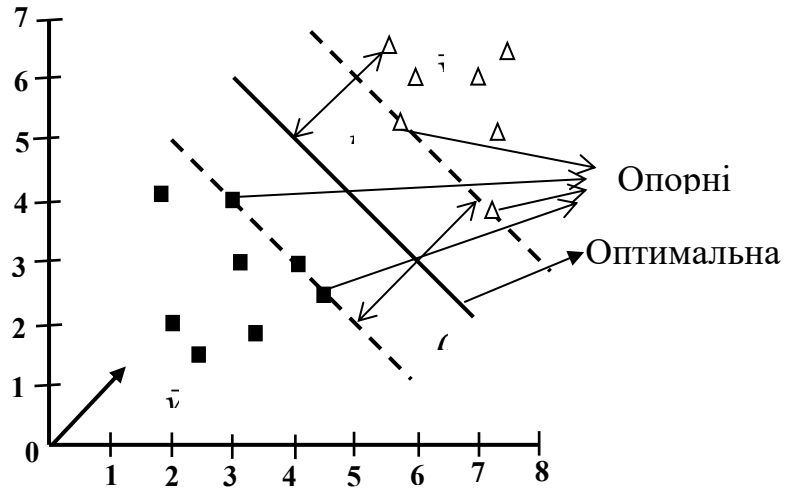


Рисунок 2.1 – Приклад побудови методом SVM оптимальної гіперплощини: r – відстань до точки $\vec{x}$, $\vec{w}$ - вектор нормалі до гіперплощини, ρ - маржа.

На рис. 2.1 для спрощення розуміння проблеми показано опис об’єктів двох класів у двохвимірному просторі ознак (на площині) у вигляді точок,що є кінцями векторів, що є зображеннями даних об’єктів. В цьому простому випадку представлений прямими лініями лінійний класифікатор спрощує наглядність уяви про розділення об’єктів на два класи, тому що тут замість векторів і гіперплощин у просторі вищої розмірності ми маємо справу з точками і прямими. Дане спрощення не виключає можливості застосування таких самих понять до задач класифікації у просторах вищої розмірності [30].

З наведеного прикладу зрозуміло, що є декілька прямих, які дозволяють провести класифікацію. Для виішення питання, яка з них найкраща, можна інтуїтивно критерій оцінки якості дискримінантних прямих вибрати таким чином: пряма не якісна, якщо проходить занадто близько до точок, оскільки віона буде чутливою до впливу завад і не буде мати узагальнюючого характеру під час класифікації нових точок, яких не було у навчальній вибірці, тобто адекватно їх класифікувати не буде. Тому потрібно знайти такий спосіб побудови, який знайде лінію, яка буде проходити якнайдалі від усіх точок.

В такій постановці, робота алгоритму SVM, по-перше, грунтується на пошуку гіперплощини, що має найбільшу мінімальну відстань ρ до опорних

точок. По-друге, в теорії SVM дана відстань отримує назву маржі. Звідси розподільна гіперплощина, що є оптимальною, повинна максимізувати запас маржі для навчальних даних [31, 32].

2.1.2 Математичні засади класифікатора SVM

Для *n*-мірного простору узагальнений математичний вираз лінійної дискримінантної (вирішувальної) функції має такий вигляд:

$$f(\vec{x}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n = w_0 + \vec{w}_a^T \vec{x}, \quad (2.1)$$

де: $\vec{w}_a = (w_1, w_2, ..., w_n)^T$ - вектор нормалі, він же ваговий вектор;

w_0 - зміщення відносно початку координат розподільної гіперплощини;

$\vec{x} = (x_1, x_2, ..., x_n)^T$ - вектор зображення об'єкта.

Проекцію розподільної гіперплощини (2.1) на двовимірний простір зображено на рис. 2.2.

На ній показано:

$\vec{u}$ - додатноорієнтована одинична нормаль;

$\vec{p}$ і $\vec{x}$ - деякі точки гіперплощини і простору;

0 – початок координат.

З геометричної точки зору рівняння гіперплощини записується я:

$$\vec{u}^T (\vec{x} - \vec{p}) = 0, \quad (2.2)$$

або

$$\vec{u}^T \vec{x} = \vec{u}^T \vec{p} \quad (2.3).$$

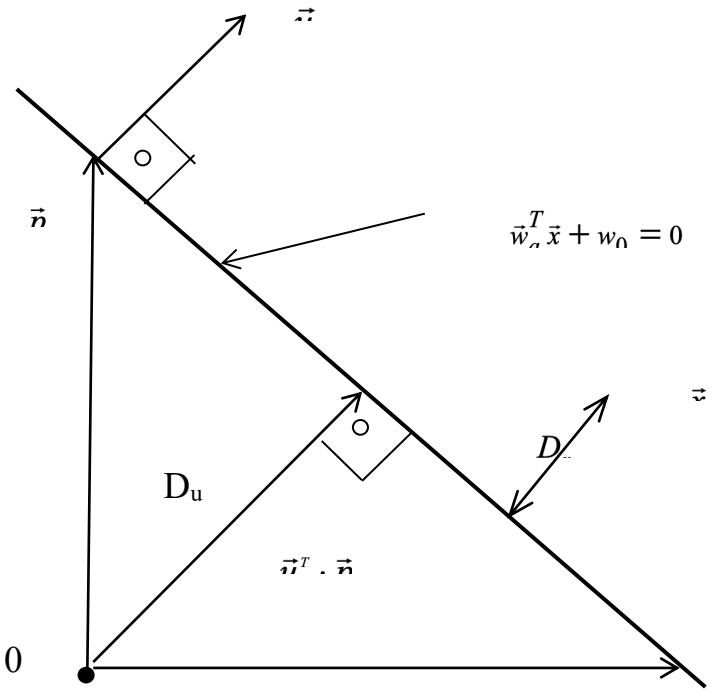


Рисунок 2.2 – Зображення розподільної гіперплощини у вигляді проекції на двовимірний простір

З рівняння (2.1) рівняння гіперплощини можна записати як:

$$\vec{w}_a^T\vec{x}+\vec{w}_0=0, \qquad (2.4),$$

або

$$\vec{w}_a^T\vec{x}=-\vec{w}_0\ . \qquad (2.5).$$

Якщо пронормувати ваговий вектор діленням (2.5) на його норму (довжину) $\left\|\vec{w}_a^T\right\|=\sqrt{w_1^2+w_2^2+...+w_n^2}$, то отримаємо:

$$\frac{\vec{w}_a^T\vec{x}}{\left\|\vec{w}_a^T\right\|}=-\frac{w_0}{\left\|\vec{w}_a^T\right\|}. \qquad (2.6)$$

Шляхом порівняння (2.3) і (2.6) отримаємо вираз одиничної нормалі - $\vec{u} = \frac{\vec{w}_a}{\|\vec{w}_a\|}$, вираз відстані від початку координат до гіперплощини - $D_u = |\vec{u}^T \cdot \vec{p}| = \frac{|w_0|}{\|\vec{w}_a\|}$, вираз відстані від точки простору до гіперплощини - $D_x = |\vec{u}^T \vec{x}^* - \vec{u}^T \vec{p}| = \left| \frac{\vec{w}_a^T \cdot \vec{x}^* + w_0}{\|\vec{w}_a\|} \right|$. Ці визначення будуть використані під час побудови принципів оптимізації процедури навчання класифікатора типу SVM.

2.1.3 Постановка задачі навчання класифікатора за методом SVM

Задачу навчання класифікатора SVM математичну сформулюємо таким чином [31].

Існують вхід дані для навчання:

$$U = \{(\vec{x}_i, y_i)\}, \quad \vec{x}_i \in R^n, \quad y_i \in \{-1, +1\}, \quad i = 1,2,...,m, \quad k = 1,2,...,n, \qquad (2.7)$$

де через U позначена навчальна вибірка векторів об'єктів $\vec{x}_i$, для яких відомі марки y_i цих векторів, що визначаються наступним чином:

якщо $\vec{x}_i \in \Omega_1$ то $y_i = 1$, а якщо $\vec{x}_i \in \Omega_2$ то $y_i = -1$,

отже y_i визначає, до якого класу входить вектор $\vec{x}_i$;

R^n - простір параметрів (ознак) розмірності n;

m – число векторів у навчальній вибірці.

За таких умов задачу навчання сформулюємо таким чином:

за прецедентами навчальної вибірки побудувати вирішувальну функцію $f(\vec{x})$ таким чином, щоб вона відповідала наступним вимогам:

$$\begin{cases} f(\vec{x}_i) > 0, & \vec{x}_i \in \Omega_1 \\ f(\vec{x}_i) < 0, \vec{x}_i \in \Omega_2 \end{cases}, \text{ або } \quad y_i \cdot f(\vec{x}_i) > 0, \quad \vec{x}_i \in U. \qquad (2.8)$$

Одне узагальнене для двох рівняння у виразі (2.8), дозволяє записати лінійний класифікатор на 2 класи таким чином:

$$f(\vec{x}_i) = sign(\vec{w}^T \vec{x}_i + w_0). \qquad (2.9)$$

Рівність +1 значення виразу (2.9) означає належність об'єкта до першого класу, а якщо -1, тоді до другого. Так як під час навчання значення +1 і -1 для векторів $\vec{x}_i$ навчальної вибірки є відомими та задаються значеннями y_i, то вираз (2.9) можна записати:

$$f(\vec{x}_i) = y_i \cdot (\vec{w}^T \vec{x}_i + w_0). \qquad (2.10)$$

Останній вираз визначає відстань (ненормовану) від точок 1-го і 2-го класу до гіперплощини $\vec{w}^T \vec{x} = 0$. Цю відстань назвемо маржою, або функціональним зазором. Як було показано в попередньому розділі, нормалізація цього зазору діленням на норму вектора нормалі $\vec{w}$ дозволяє усунути в різних задачах проблему масштабування даних з точки зору відстані. При цьому геометрична маржа буде незалежаною від масштабування. Для прикладу, при заміні параметрів $\vec{w}$ на $5\vec{w}$ і $\vec{w}_0$ на $5\vec{w}_0$ геометричний зазор не зміниться. При встановленому обмеженні $\|w\| = 1$ геометричний зазор збігається з функціональним.

Домножвши для масштабування $f(\vec{x})$ на деяке додатне число з того факту, що $f(\vec{x})$ - лінійна функція, з виразу (2.10) отримаємо:

$$y_i \cdot (\vec{w}^T \vec{x}_i + w_0) \geq 1, \;\; i = 1,...,m. \qquad (2.11)$$

З цього витікає, що промасштабована відстань між граничними гіперплощинами $\vec{w}^T\vec{x}_i + w_0 - 1 = 0$ і $\vec{w}^T\vec{x}_i + w_0 + 1 = 0$, які проходять через опорні точки, дорівнює:

$$r = \frac{2}{\|w\|}. \qquad (2.12)$$

Останній вираз показує, що мінізація величини $\|w\|$ приводить до максимізації маржі. Оскільки норма $\|w\|$ включає в себе квадратний корінь, то вирішити в такому вигляді задачу оптимізації такої функції важко. Процедуру оптимізації можна спростити, якщо замінити $\|w\|$ на $\|w\|^2$ та домножити на коефіцієнт 1/2, що не приведе до зміни результату. В такому вигляді маємо квадратичну задачу оптимізації [32,33].

Задача оптимізації, таким чином, зведеться до пошуку вагового вектора, який максимізує r :

$$\frac{1}{2}\|w\|^2 \to \min \text{ при обмеженнях } y_i \cdot (\vec{w}^T\vec{x}_i + w_0) \geq 1, \quad i = 1,...,m. \qquad (2.13)$$

Задача з обмеженнями можна представити у вигляді задачі без обмежень шляхом введення множників Лагранжа,:

$$\min_{w,w_0} \max_{\alpha \geq 0} \left\{ \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\vec{w}^T\vec{x}_i + w_0) - 1) \right\} \qquad (2.14)$$

Задачу пошуку оптимальних опорних точок розв’язують шляхом введення функції-лагранжіана:

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j^T \qquad (2.15)$$

при умовах

$$\sum_{\text{i}} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0 \quad \underset{i}{\forall}(i = \overline{1,m}),$$

де α_i - множники Лагранжа.

Розв'язок поставленої задачі має такий вигляд:

$$\vec{w} = \sum \alpha_i y_i \vec{x}_i \,. \tag{2.16}$$

З (2.16) витікає, що ваговий вектор для оптимальної гіперплощини можна отримати з виразу:

$$\vec{w} = \sum_{i=1}^{k} \alpha_i y_i \vec{x}_i \,, \tag{2.17}$$

де i – номер опорної точки (вектора), i=1…k; k – число шуканих опорних точок, α_i - множники Лагранжа, визначені для опорних точок процедурою оптимізації.

2.1.3 Алгоритмізація процесу навчання класифікатора SVM

В обчислювальному плані алгоритм пошуку величин множників Лагранжа для побудови оптимальної гіперплощини за виразом (2.17) зводиться до розв'язку системи лінійних алгебраїчних рівнянь, яку отримують шляхом визначення частинних похідних для заданої функції Лагранжа по вибраних множниках α_i при відомих значеннях координат x_{ji} опорних точок $\vec{x}_i$ та значень марок y_i:

$$\begin{vmatrix} n & \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{2i} & \dots & \sum_{i=1}^{n} x_{mi} \\ \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{1i}x_{1i} & \sum_{i=1}^{n} x_{1i}x_{2i} & \dots & \sum_{i=1}^{n} x_{1i}x_{mi} \\ \sum_{i=1}^{n} x_{2i} & \sum_{i=1}^{n} x_{2i}x_{1i} & \sum_{i=1}^{n} x_{2i}x_{2i} & \dots & \sum_{i=1}^{n} x_{2i}x_{mi} \\ \dots & \dots & \dots & \dots & \\ \sum_{i=1}^{n} x_{mi} & \sum_{i=1}^{n} x_{mi}x_{1i} & \sum_{i=1}^{n} x_{mi}x_{2i} & \dots & \sum_{i=1}^{n} x_{mi}x_{1m\cdot} \end{vmatrix} \cdot \begin{vmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{vmatrix} = \begin{vmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i x_{1i} \\ \sum_{i=1}^{n} y_i x_{2i} \\ \dots \\ \sum_{i=1}^{n} y_i x_{mi} \end{vmatrix} . \tag{2.18}$$

Алгоритм навчання класифікатора типу SVM наведений на рис. 2.3

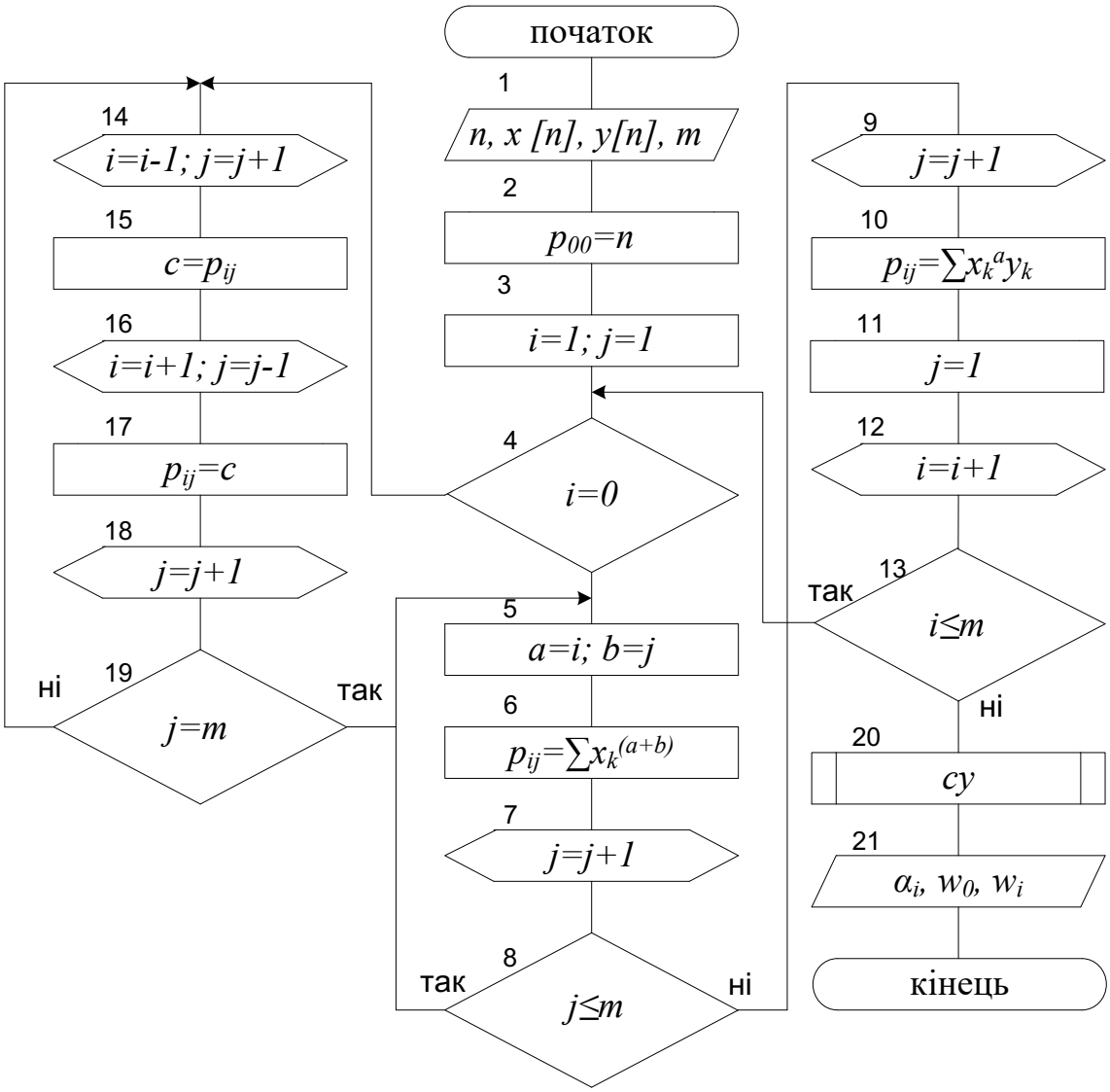


Рисунок 2.3 - Схема алгоритму навчання класифікатора типу SVM

Вхідними даними до алгоритму навчання є навчальна вибірка, що складається з масивів векторів об'єктів $\vec{x}[n]$ і їх мітокк $y[n]$, де n задає кількість елементів в навчальній вибірці, а m – розмірність ознакового простору.

Алгоритм складається з чотирьох основних кроків: введення даних вхідних даних; обчислення сумм, що визначають коефіцієнти перед невідомим в системі (2.18); розв'язання одержаної системи та друк результатів, якими є числові значення множників Лагранжа α_i та коефіцієнтів w вагового вектора. Детальний вигляд алгоритма показаний на рис. 2.3. Під час обчислення

коефіцієнтів p_{ij} при невідомих в блоці 6 алгоритму використовується властива для систем (2.18) закономірність:

$$p_{ij} = \sum_{k=0}^{n} x_k^{(i+j)}.$$

Вільні члени рівнянь обчислюються в блоці 10 алгоритму за формулою

$$p_{ij} = \sum_{k=0}^{n} x_k^i y_k.$$

Скороченяо кількості обчислень досягнуто за рахунок того, що повністю обчислюються: 1-ша стрічка, остання і передостання колонки матриці системи, що показано в блоки 4 і 13. В блоках 1, 6 і 10 дії наведені в спрощеному вигляді. В програмі вони реалізуються у вигляді циклічних процесів. В алгоритмі використанії наступні позначення:

i,j – індекси, якими позначаються номер стрічки и колонки відповідно; *k* – індекс, що позначаєт номер елемента в масиві *x* або *y*; p_y – коефіцієнти, які замінюють відповідні суми в системі рівнянь (2.18).

Після знаходження можників α_i Лагранжа алгоритмом обчислюється за формулою (2.17) ваговий вектор $\vec{w}$, що визначає в просторі ознак кут нахилу гіперплощини. Величина зміщення w_0 оптимальної гіперплощини обчислюється за всіма граничними опорними векторами як медіана з:

$$w_0 = med\left\{(\vec{w}, \vec{x}_i) - y_i : \alpha_i > 0, i = \overline{1,k}\right\}. \quad (2.19)$$

2.2 Розробка методу кластеризації даних

2.2.1 Завдання кластерного аналізу даних

Модель класифікації за алгоритмом SVM, що описана в попередньому розділі, так само як і моделі інших класифікаційних методів, можна будовати

при наявності навчальних вибірок, що представляють зразки відомих класів. В зв'язку з цим перед машинним навчаннм моделі класифікатора потрібно виконати процедуру кластеризації даних. Її реалізують шляхом неконтрольоване навчання .

Як попередньо уже було визначено, під кластерним аналізом в загальному випадку розуміють групування деякої множини об'єктів таким чином, щоб об'єкти однієї групи (яку називають кластером) були більш похожі (у визначеному сенсі) один до одного, ніж до об'єктів в інших групах (кластерах) [30]. Якщо для заданої множини об'єктів відсутня будь-яка інформація відносно груп (класів), на які вони поділяються, задачу ідентифікації класів можна сформулювати як задачу неконтрольованого навчання, або "навчання без учителя".

Вказану задачу навчання без учителя вирішують шляхом застосування до заданої вибірки даних алгоритмів кластеризації. Отримані при цьому кластери можна вважати класами зображень. Наявність цих класів дає можливість використати їх для отримання дискримінантних (вирішувальних) функцій шляхом використання алгоритмів контрольованого навчання ("навчання з учителем"). Також отримані на стадії навчання без учителя центри кластерів можна безпосередньо використати для побудови класифікатора, що функціонує за принципом мінімуму відстані [32].

Оскільки більшість алгоритмів кластеризації базуються на апріорному досвіді розробника і його інтуїції, отриманих в результаті проведених раніше експериментів, то їх можна назвати неконтрольованим навчанням лише умовно.

2.2.2 Розробка алгоритму кластеризації за принципом поверхневого натягу

На разі існує значнаа кількість методів і відповідних їм алгоритмів кластеризації даних (більше 100). До них можна віднести як класичні методи, які базуються на узагальненому алгоритмі К. С. Фу [24], так і новітні, що

зумовлені структурою і природою даних і ціллю їх використання [25, 26]. Всім цим методам і алгоритмам кластерного аналізу властиві свої недоліки і переваги, які зумовлені орієнтацією на розв'язання вибраного класу задач. До загального недоліку можна віднести те, що вони у просторі ознак формують кластери точок, але їхні границі не фіксують. Остання характеристика Наявність маркованих граничних точок кластера є дуже корисною під час побудови класифікатора. В роботі [34] авторами запропоновано метод пошуку опорних точок для класифікатора SVM шляхом знаходження граничних точок кластера за аналогією з фізичним принципом поверхневого натягу, властивого рідинам.

Сутність казаного явища ілюструється на рис. 2.4, на якому зображена фізична природа натягу в поверхневому шарі рідини, зумовленого нескомпенсованим притяганням внутрішніми молекулами поверхневих [35].

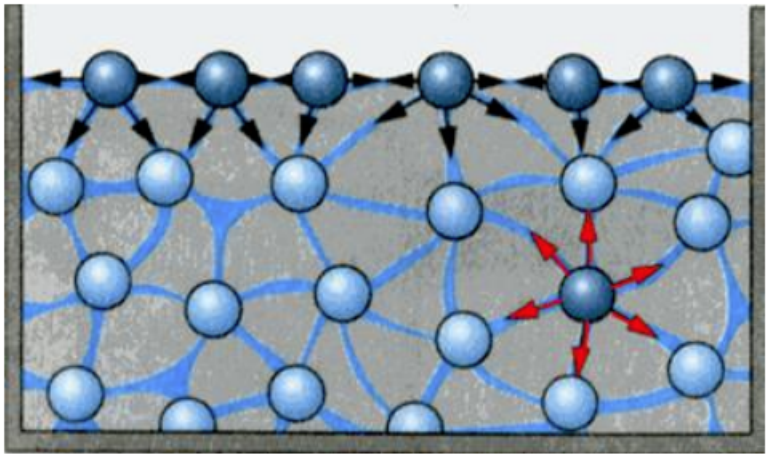


Рисунок 2.4 – Демонстрація поверхневого натягу молекул в рідині

В рідині на кожну молекулу діють сили притягання від сусідніх молекул. Для молекул, що знаходяться всередині рідини, ці сили взаємно скомпенсовані. Рівнодійняка сил притягання, що діє на молекули, що знаходяться на поверхні розділення середовищ рідина-повітря, напрямлена усередину рідини. В поверхневому шарі, товщина якого дорівнює радіусу дії

молекулярних сил, молекули чинять тиск на рідину, стягуючи її поверхню до мінімуму. Це явище називається поверхневим натягом.

Ідея запропонованого авторами підходу до кластерного аналізу даних полягає в використанні аналогії з описаним явищем повеневого натягу рідини. В якості поверхневих точок кластера вибираються ті, що неврівноважені силами взаємного впливу оточуючими точками (оскільки вони на поверхні кластера). Із цих знайдених точок вибирають точки сусідніх класів, між якими відстані найменшіи, вони і будуть опорними. У вказаному підході приймаються такіі припущення:

а) вважається, що модулі сил взаємодії точок простору обернено пропорціональні відстані між ними:

$$f(\vec{x}_i, \vec{x}_j) = \frac{p}{d}, \tag{2.20}$$

де p є ваговим коефіцієнтом; d - відстань між точками, яка обчислюється за формулою

$$d = \sqrt{\sum_{k=1}^{n} x_{ik}^2} \ . \tag{2.21}$$

У виразі (2.21) через x_{ik} позначео k-у координату i-ої точки вибірки.

б) на вибрану окрему точку діють сили притягання тільки від найближчих точок.

Проведений в роботі [32] аналіз результатів машинного експерименту, показав, що запропонований підхід можна використовувати як для оптимізації процедури навчання класифікатора, так і для кластерного аналізу даних шляхом фіксації границі кластерів поверхневими точками.

В даній магістерській кваліфікаційній роботі проведена модифікація запропонованого у згаданій роботі алгоритму з метою його використання для

програмної реалізації побудови кластерів шляхом використання методом поверхневого натягу для визначення поверхнвих точок.

Задачу кластеризації даних шляхом фіксації граничних точок кластера в загальному випадку можна сформулювати таким чином [40]:

Відомі масив об'єктів X і масив номерів (міток) кластерів Y. Задана у просторі ознак $\Re^n$ вибірка зображень об'єктів деякою підмножиною точок $X^m = \{x_1,...,x_m\} \subset X$, де n – розмірність простору ознак, m – кількість точок у вибірці,. Також задана метрика $D(x_1, x_2)$ для відстані міру відстані між точками і задана функція натягу (стрес) $\Delta(x_k, z_k)$ між вибраною поточною точкою x_k і групою найближчих до неї точок, де z_k - центр вибраної групи точок. По цій заданій інформації треба розділити задану вибірку об'єктів на непересічні кластери таким чином, щоб в кожомун кластері були тільки близькі за метрикою D точки. Також потрібно при цьому фіксувати в масиві S_{Y_i} точки, які лежать на поверхні кластера (тобто є найбільш віддалені від його центру), де Y_i - кластер з номером i.

Алгоритм класифікації розробляється у вигляді вирішувальної функції $f: X \to Y$, яка ставить у відповідність номер кластера $y \in Y$ будь-якому об'єкту $x \in X$, причому для кожного кластера фіксується підмножина поверхневих точок $S_{Y_i} \in X$. В більшості випадків множина кластерів Y наперед не є відомою, при цьому потрібно вирішити задачу пошуку оптимальної кількості кластерів.

Побудова такого алгоритму при узагальненому підході вимагає знаходження для кожного кластеру множини поверхневих точок застосуванням ітераційної процедури градієнтного спуску, оптимальної за критеріїєм максимізації поверхневого натягу. Задача в такій узагальненій постановці не розглядається за мету даної магістерської кваліфікаційної роботи. Вона в даній роботі зведена до розробки алгоритму ітераційного пошуку оптимального натягу (стресу) для визначення поверхневих точок тільки для одного з побудованих кластерів. Отримані результати дають

можливість розширити роботу алгоритма також на випадок декількох кластерів.

Наведемо словесний опис розробленого в роботі алгоритму, Для заданого масиву даних $MAS(M) = \{\vec{X}_1, \vec{X}_2, ..., \vec{X}_m, ..., \vec{X}_M\}$ N-мірного простору ознак він описується таким чином.

1) Для всіх різних пар точок обчислюємо масив відстаней за виразом:

$$D(\vec{X}_i, \vec{X}_j) = \sqrt{\sum_{n=1}^{N} (x_{in} - x_{jn})^2}, \qquad (2.22)$$

в якому кількість різних пар відстаней дорівнює $C_M^2 = \frac{M(M-1)}{2}$.

2) Переглядаємо послідовно всі точки масиву $MAS(M)$ шляхом зміни індексу точок m від 1 до M, і виконуємо для кожної з них такі дії:

а) Використовуємо масив відстаней, в ньому вибираємо 5 найближчих точок до поточної k-ої. На вході алгоритму передбачена можливість зміни кількості найближчих точок. Досліджувалися випадки при 3, 4, 5 і 6 найближчих точках, що відповідало разом з взятою k-ою точкою 4, 5, 6 і 7 точкам в ядрі, при цьому оптимальною виявилася кількість з 5 найближчих точок

б) Для вибраної групи точок визначаємо середню відстань за формулою:

$$d_m = \frac{\sum_{p=1}^{5} D(\vec{X}_m, \vec{X}_p)}{5}, \qquad (2.23)$$

де p – індекси чотирьох точок, найближчих до k-ої.

в) з урахуванням координат вибраної поточної точки $\vec{X}_m$ обчислюємо координати (z_{m1}, z_{m2}, …, z_{mn},…, z_{mN}) центра $\vec{Z}_m$ даної сукупності точок:

$$z_{m1} = \frac{\sum_{p=1}^{6} x_{p1}}{6}, z_{m2} = \frac{\sum_{p=1}^{6} x_{p2}}{6}, ..., z_{mn} = \frac{\sum_{p=1}^{6} x_{pn}}{6}, ..., z_{mN} = \frac{\sum_{p=1}^{6} x_{pN}}{6} \quad , \qquad (2.24)$$

де p – номери точок, які належать поточній сукупність точок.

г) Відносне зміщення точки $\vec{X}_m$ відносно центру $\vec{Z}_m$ визначаємо з виразу:

$$\Delta_m = d(\vec{X}_m, \vec{Z}_m) = \frac{\sqrt{\sum_{n=1}^{N} (x_{mn} - z_{mn})^2}}{d_m}, \qquad (2.25)$$

де n – номери координат точок $\vec{X}_m$ і $\vec{Z}_m$, для заданого масиву $MAS(M)$ точок.

д) Перевіряємо умову $\Delta_k > \sigma$; якщо вона виконується, то точка $\vec{X}_k$ є поверхневою в кластері.

Ітеративно повторюєм процедуру для різних значень поверхневого натягу $\Delta(x_k, z_k)$ і порогу σ до тих пір, поки не отримаємо для чергового набору поверхневих точок максимального значення стресу, який використовується за умову припинення пошуку граничних (поверхневих) точок кластера.

Блок-схема алгоритму кластеризації даних запропонованим методом поверхневого натягу зображена на рис. 2.5.

Для реалізації описаного алгоритму розроблено програми мовами Python і Java, лістинги яких наведені в Додатку Б.

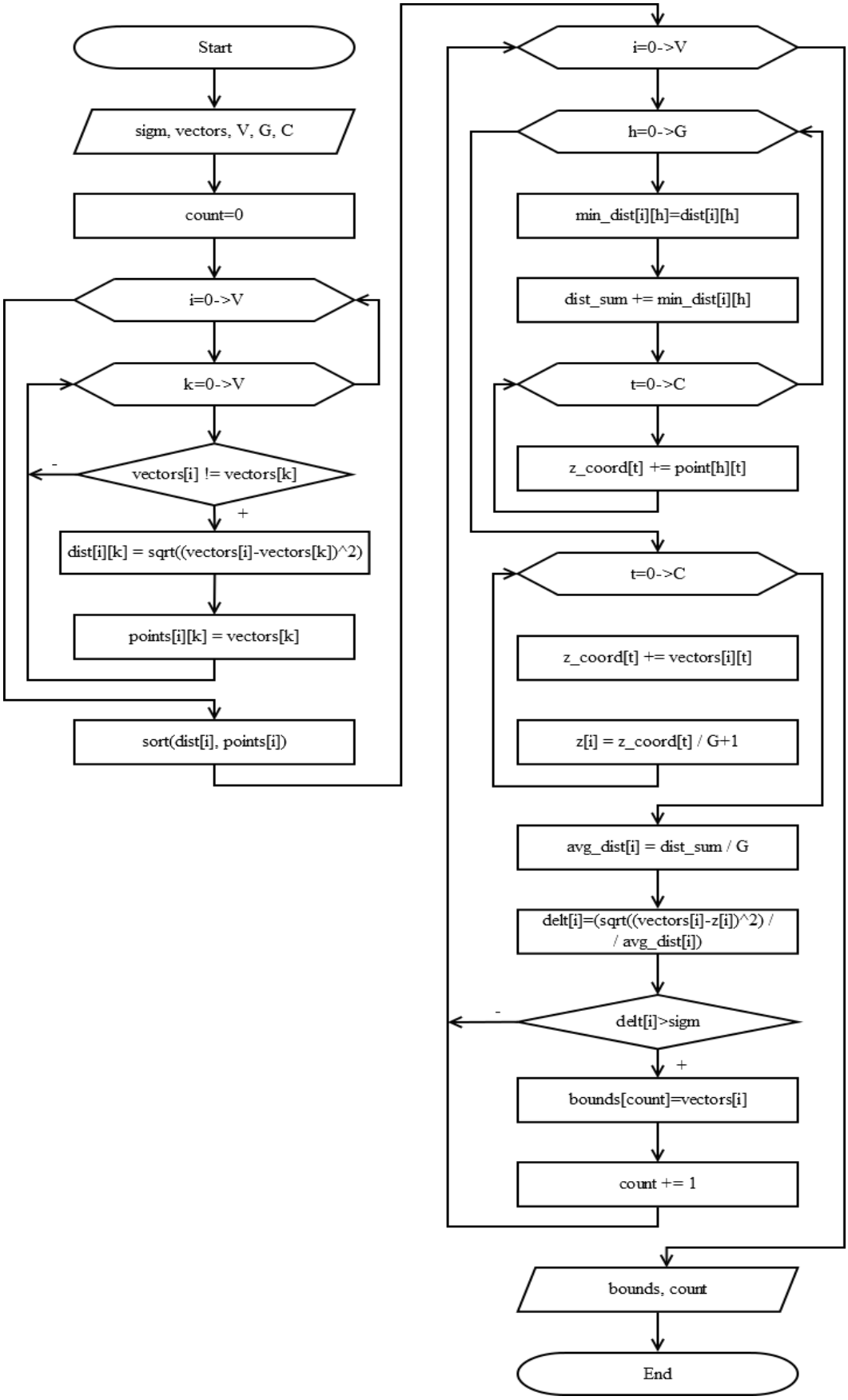


Рисунок 2.5 – Схема алгоритму знаходження поверхневих точок кластера

2.3 Висновки до розділу

В даному розділі роботи розглянуто принцип роботи класифікатора SVM і наведено математичні засади і постановка задачі машинного навчання даного класифікатора. Показано, що в відомій постановці його недоліком є необхідність проведення складних оптимізаційних обчислень для знаходження опорних векторів методом множників Лагранжа. Для усунення виявленого недоліку запропоновано метод кластеризації даних, що фіксує поверхневі точки кластера за принципом поверхневого натягу. Цей метод використовується для зменшення обчислювальних затрат під час навчання класифікатора. Розроблено схеми алгоритмів навчання модифікованого алгоритму SVM і кластеризації даних методом поверхневого натягу.

3 ПРОГРАМНОГО РЕАЛІЗАЦІЯ

3.1 Обгрунтування вибору програмної платформи

При розробці програмних продуктів ефективність процесу програмування і якість його результатів в великій мірі визначається вибором програмної платформи. Під час вирішення даної проблеми впливають такі основні фактори :

1. Можливість комерційної реалізації розробленого програмного продукту.

2. Наявність в середовищі програмування засобів для реалізації в алгоритмах стандартних підпрограм, що дозволяє розробнику пришвидшити процес програмування, а також тестування адекватності розроблених алгоритмів та якості програм шляхом проведення машинного експерименту.

Для визначення програмних середовищ, які задовольняють вказаним вимогам, проведемо аналіз програмних платформ, які використовуються для реалізації задач машинного навчання.

На сьогоднішній день існує декілька поширених мов програмування і значна кількість бібліотек до них, що використовуються для реалізації задач машинного навчання [41]. До найбільш відомих мов такого застосування можна віднести мови Python та R.

Мова R є досить популярною в середовищі дата-аналітиків. Початково вона розроблювалася для розв’язання задач статистичного аналізу даних, а на разі широко застосовується в областях Big Data і Data Mining. Основним її недоліком є складнощі її застосування для вирішення задач, які не пов'язані з аналізом і візуалізацією даних.

Мова Python, на відміну від мови R, є мовою загального призначення і може бути успішно використана під час вирішення завдань різного характеру, в тому числі і задач машинного навчання. Через це ця мова в останні кілька років набрала великої популярності серед фахівців з ML і стала головною в

реалізації задач даного напрямку в IT. Мова Python є дуже просто і легкою для вивчення. Її застосування в машинному навчанні не вимагає знання всі тонкощів програмування, оскільки підтримується значною кількістю бібліотек, призначених для реалізації з їх допомогою функцій окремих задач і методів машинного навчання, зокрема, регресійного аналізу, зменшення розмірності даних, кластеризації і класифікації об'єктів, графічної візуалізації результатів розрахунків, генерації випадкових величин та інш [42].

Більшість сучасних бібліотек глибокого (DL) і машинного (ML) навчання надають мові Python додатки API. До найбільш популярних бібліотек відносяться [43,44]:

- NumPy: задачі обробки багатовимірних масивів і лінійної алгебри; використовується в якості допоміжної бібліотеки під час виконання різних математичних операцій зі структурами даних.

- Scikit-learn: найкраща бібліотека для класичних ML алгоритмів; підтримує більшість алгоритмів навчання, таких як: лінійна і логістична регресія, метод SVM, Байесовський класифікатор, бустінг, кластеризація, метод найближчого сусіда, k-внутрішньогрупових середніх та багато інш. Jupyter Notebook: робота з даними, кодом і графіками; дозволяє писати код на Python і інших мовах, додавати текстові описи, вбудовувати графіки та діаграми безпосередньо в інтерактивну веб-сторінку.

- Tensorflow і Keras: бібліотеки глибокого навчання; містить такі компоненти, як багатовимірні масиви (тензори), оператори лінійної алгебри та обчислення похідних.

- Pandas: витяг і підготовка даних; дозволяє завантажувати дані з різних джерел, таких як бази даних SQL, у вигляді CSV, Excel, JSON файлів і інших форматів. Для завантажених в пам'ять даних дозволяє виконувати безліч різних операцій для аналізу, трансформації, заповнення відсутніх значень і очищення набору даних.

З урахуванням вказаних можливостей мови Python її було вибрано для реалізації і тестування запропонованих в роботі ефективних методів ідентифікації об’єкті з використанням класифікації і кластерного аналізу.

З урахуванням 1-го фактора, з огляду на заплановану комерційну реалізацію розроблених методів і алгоритмів, частина алгоритмів також реалізована мовою програмування Java, яка на сьогодні найчастіше застосовується для розробки комерційних додатків.

3.2 Розробка структури програмного забезпечення

Для вирішення поставлених в індивідуальному завданні на магістерську роботу завдань підвищення ефективності ідентифікації об’єктів шляхом покращення методів кластеризації і класифікації даних, необхідних для підтримки прийняття рішень в інтелектуальних комп’ютерних системах управління, в роботі були розроблені алгоритм кластеризації даних з фіксацією граничних точок кластера, модифікований алгоритм класифікації даних SVM з використанням знайдених алгоритмом кластеризації поверхневих точок, і алгоритм візуалізації отриманих результатів обробки даних у двовимірному просторі шляхом зменшення розмірності описового простору ознак. Вказані алгоритми реалізовані у вигляді одного програмного додатку ClusterBoundsSearcher_v0.0.2. На рис. 3.1 зображена UML-діаграма, в якій показана схема взаємодії програм в додатку. Тексти програм мовою Python і Java наведені в Додатку Г. Програмний код розробленого за стосунку з урахуванням його можливого комерційного використання в майбутньому, надається по запиту.

На рис.3.1 розроблена в магістерській роботі UML-діаграма відображує етапи використання розроблених програми кластеризації, на вхід подається навчальна вибірка заданої розмірності, програми класифікації отриманих даних удосконаленим класифікатором SVM та програми графічної

візуалізації результатів шляхом пониження їхньої розмірності для кращого розуміння до двомірного простору (площини).

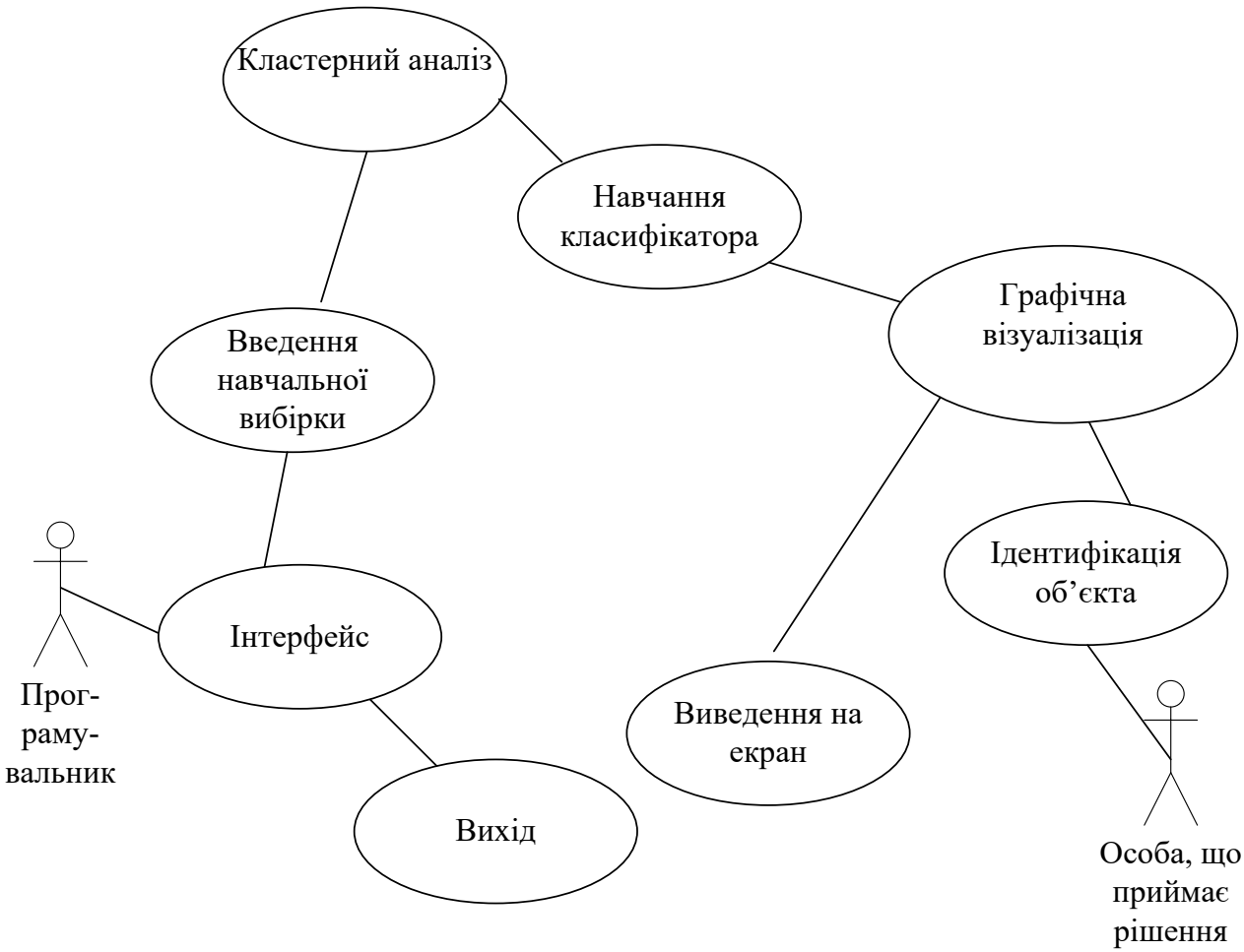


Рисунок 3.1 – UML-діаграма використання програмного забезпечення

Склад програмного додатку ClusterBoundsSearcher_v0.0.2 показано на рис. 3.2.

Рисунок 3.2 – Програмний додаток ClusterBoundsSearcher_v0.0.2

В програмний додаток входить папка “scripts” з скриптами модулів мовою Python для кластерного аналізу даних, класифікації даних

(ідентифікації) та їх візуалізації. Екран частини тексту програмного модуля ChartBuilder.py для класифікації та кластеризації даних мовою Python показано на рис.3.3.

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.figure
import numpy as np
import csv
from sklearn.manifold import MDS
from sklearn.decomposition import PCA
from sklearn import svm
import argparse


parser = argparse.ArgumentParser(description='Process data from File')
parser.add_argument('path')
parser.add_argument('svm_gamma')
args = parser.parse_args()
path = args.path
svm_gamma = float(args.svm_gamma)


def transform_to_form(list, dimension:int):
    transformed = [0]*len(list[0])
    for idx in range(0, len(list[0])):
```

Рисунок 3.3 - Скрі екрана з частиною тексту програми мовою Python

Рис. 3.3. показує, що в заголовках (хідерах) програми розміщені 4 оператори імпорту бібліотек Matpotlib, необхідних для виконання алгоритму по створенню графіків та візуалізації даних; бібліотеки Numpy, необхідної для обробки масивів та розв'язання систем лінійних алгебраїчних рівнянь в процедурі знаходження множників Лагранжа під час побудови оптимального класифікатора; функції CSV, що використовується для завантаження введених в графічному інтерфейсі вибіркових даних; бібліотеки Scicit-leran, необхідної для завантаження функцій багатовимірного метричного масштабування MDS, та методу PCA аналізу головних компонент, що реалізує зниження розмірності простору ознак перед візуалізацією даних.

Як було попередньо зумовлено в Розділі 3.1, в програмному блоці вставлено модуль ClusterBoundsSearcher.jar мовою Java, призначеного для

реалізації розроблених алгоритмів кластеризного аналізу і класифікації, з метою комерціалізації розробленого додатку в майбутньому. Запуск модуля на мові Java здійснюється файлом запуску run_additional.bat, а модуля на мові Java – файлом запуску run_additional.bat Python.

3.3 Інструкція користувачу програми

Розроблений програмний додаток ClusterBoundsSearcher_v0.0.2 дозволяє реалізувати наступні функції:

- вводити вибірку навчальних даних на кроції навчання та масиви даних, які підлягають класифікації під час етапу експлуатації;

- виконувати кластер ний аналіз даних з фіксацією граничних точок кластера;

- виконувати класифікацію даних за методом SVM з використанням в якості опорних знайдених поверхневих точок знайдених кластерів;

- за необхідності проводитити багатомірне масштабування та здійснювати візуалізацію точок кластерів шляхом проектування на площину.

Результати кластеризації записуються у файли “cluster_bounds_result.txt” та “cluster_bounds.dat”. Перший містить координати усіх точок зі вхідних даних, але вони розбиті на дві групи: точки, які знаходяться в середині кластеру та точки, які формують контур кластера, відповідно є поверхневими точками. Другий містить поверхневі точки кластера у форматі вхідного файлу. Формати даних у файлах описані більш детально нижче в пункті “Формати”.

Графічний інтерфейс програми зображено на рис. 3.4. Він містить п’ять пунктів меню, що відносяться до підготовки даних і параметрів роботи, та три пункти, що відносяться до керування роботою програми:

1 – “Choose a file” - в поле потрібно задати повний шлях до файлу чи натиснути клавішу Open та обрати необхідний файл. Формат запису векторів

кластеру в файлі повинен відповідати формату, який наведено нижче в пункті “Формат файлу”.

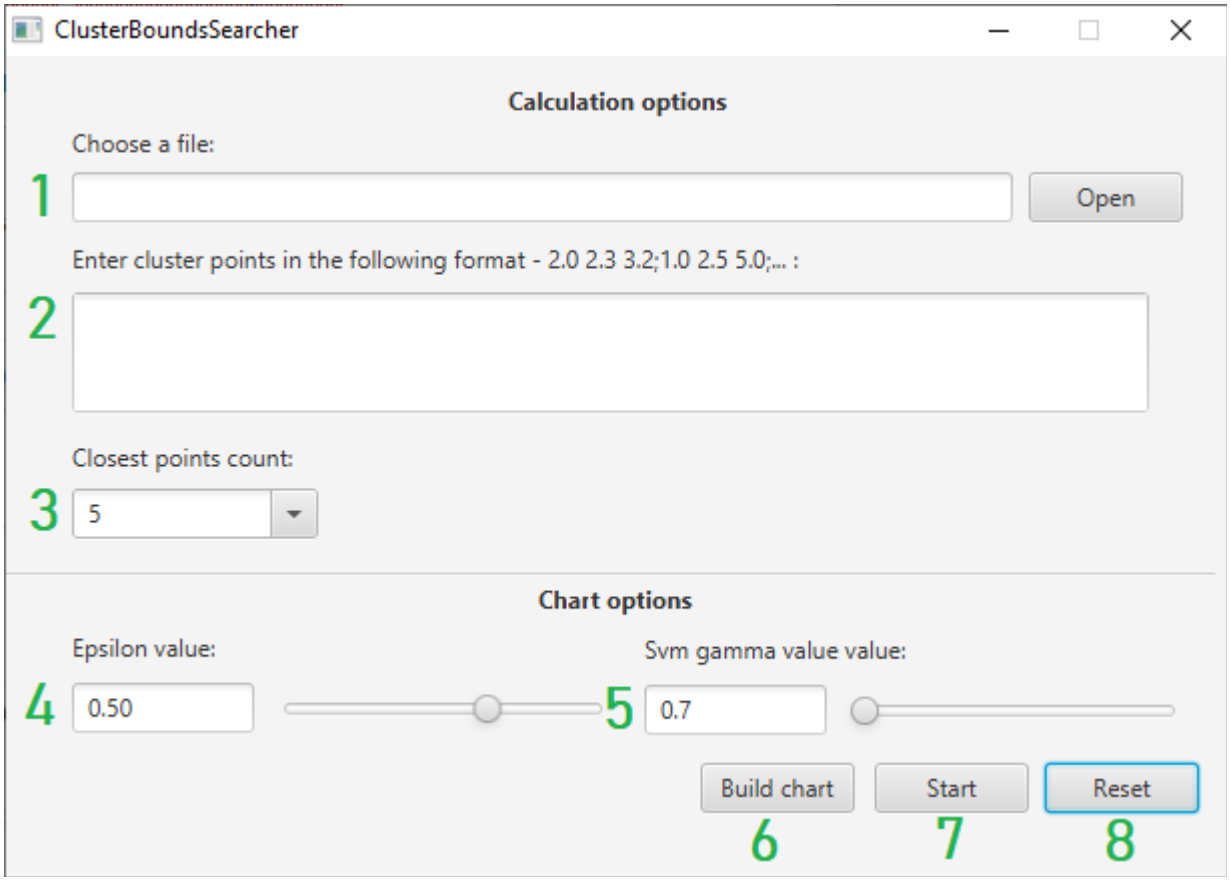


Рисунок 3.4 – Графічний інтерфейс розробленої програми

2 - Введення даних з клавіатури “Enter clusters point in the following format”- в поле потрібно ввести точки кластеру згідно формату, який наведено в нижче в пункті “Формат даних введених з клавіатури”.

3 –Вибір кількостінайближчих точок для розрахунку центрів сукупностей точок під час кластеризації “Closest point count”- можна обрати значення зі випадаючого списку чи виконати введення числового значення з клавіатури.

4 - Порогове значення відхилення “Epsilon value” поверхневого натягу (відносного зміщення точки відносно розрахованого центру) - значення обирається від 0 до 1 зі кроком 0.01.

5 - Введення параметру впливу точок кластера на результати класифікації “SVM gamma value”для методу SVM з типом ядра RBF - значення обирається від 0 до 100 зі кроком 0.05.

Керування:

6 - Клавіша Build chart - запускає процес побудови графіків.

7 - Клавіша Start - запускає процес кластеризації і класифікації та запис результатів у відповідні файли.

8 - Клавіша Reset - повертає усі значення полів графічного інтерфейсу на значення за замовченням.

Примітка: при заповненні обох полів вхідних даних (п. 1 та 2), пріоритет надаватиметься полю з файлом.

Вимоги до форматів даних, з якими працює програма, такі:

1. Формат файлу

Кожен рядок містить координати однієї точки кластеру. Координати розділені символом пробілу (‘ ’) та представляють числове значення зі плаваючою точкою. Приклад формату файлу зі координатами в трьох вимірному просторі:

3.0 5.0 3.0

3.0 6.0 3.0

4.0 3.0 4.0

4.0 7.0 4.0

2. Формат даних, введених з клавіатури

Координати точок потрібно задати у вигляді одного рядку, де координати точок розділені сиволом крапки з комою, координати однієї точки розділені символом пробілу (‘ ’). Значення координати представляє числове значення зі плаваючою точкою. Приклад формату даних введених з клавіатури зі координатами в трьох вимірному просторі:

3.0 5.0 3.0;3.0 6.0 3.0;4.0 3.0 4.0;4.0 7.0 4.0

3. Формат файлу з поверхневими точками (“cluster_bounds.dat”)

Даний файл міститиме масив координат точок, які є поверхневими. Формат відповідає формату вхідного файлу. (п.1 див. Формати).

3.0 5.0 3.0

3.0 6.0 3.0

4.0 3.0 4.0

4. Формат файлу зі результатом (“cluster_bounds_result.txt”)

Даний файл потрібен для побудови графіків. Перший рядок відповідає значенню розмірності простору координат. Далі в залежності від розмірності простору координат записані рядки, де кожен рядок містить координату точки. Записані дві групи точок: точки, які знаходяться в середині кластеру та точки, які формують контур кластера, відповідно є поверхневими точками. Приклад формату файлу з результатом з координатами в трьохвимірному просторі:

3.0

4.0,5.0,5.0,6.0,7.0

7.0,4.0,6.0,5.0,6.0

4.0,5.0,5.0,6.0,7.0

3.0,4.0,7.0,8.0,3.0,6.0,9.0

6.0,3.0,3.0,4.0,5.0,7.0,6.0

3.0,4.0,7.0,8.0,3.0,6.0,9.0

Підготовка програми до запуску:

1. Заповнити один блок зі вхідних даних (п. 1 чи 2 див. Меню ПЗ);

2. Опціонально змінити додаткові параметри (п. 3-4 див. Меню ПЗ);

3. Натиснути клавішу Start для запуску процесу розрахунку відносного зміщення точок відносно відповідних розрахованих центрів.

4. Для побудови графіків необхідно натиснути клавішу Build charts. Дана клавіша стає активною після виконання розрахунку.

Примітка: значення відхилення (епсілон) та коефіцієнта ядра (гамма) можна змінити та повторно натиснути клавішу Build charts, для того щоб дослідити її вплив на кінцеву картину розташування поверхневих точок

кластеру. Дана опція не вимагає повного перерахунку вхідних даних на відміну від параметру значення найближчих точок (п. 3 меню).

3.4 Результати машинного експерименту

Для оцінки адекватності та роботоздатностіі програмного забезпечення на етапі кластеризації був проведений експеримент з застосуванням випадкової вибірки навчальних даних, що вводилася з клавіатури. На етапі класифікації з цією метою проводився експеримент з стандартним файлом “iris.dat”. На рис. 3.5 показано результати роботи програми кластеризації навчальної вибірки з двомірних векторів з фіксацією граничних точок кластера.

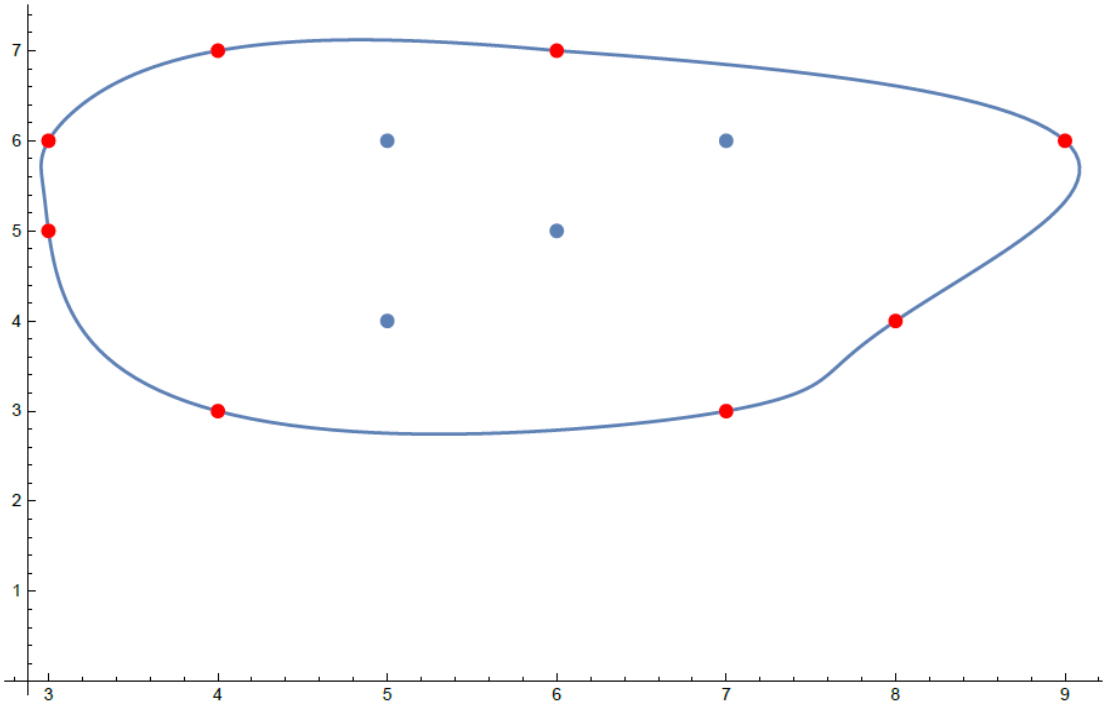


Рисунок 3.5 – Результати кластеризації даних у двомірному просторі ознак при значеннях closest point count=5 і epsilon=0,35

На рисунку червоними крапками показані поверхневі точки кластера. Проведені на різних навчальних вибірках з використанням даної програми

дослідження продемонстрували, що розроблений алгоритм кластеризації правильно фіксує поверхневі точки кластера при заданні порогу натягу epsilon в діапазоні 0,35÷0,45 і кількості точок в ядрі Closest point=5.

Результати класифікації векторів файлу “iris.dat”, спроектованих на двомірну площину, наведені на рис. 3.6.

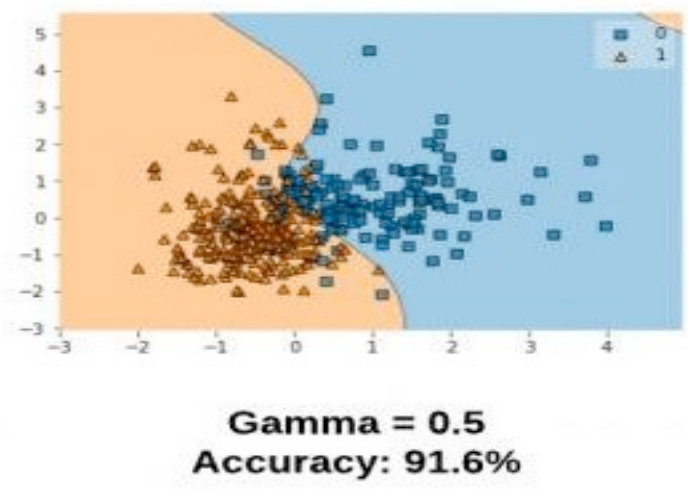


Рисунок 3.6 – Проекція класифікованих 4-х мірних векторів файлу “iris.dat” на площину

Проведений машинний експеримент показав, що класифікатор SVM, модифікований за рахунок використанням поверхневих точок кластерів, отриманих на кроції кластеризації, дає найвищу точність класифікації за умови використаннї функціонального ядра RBF та параметра Gamma=0,5. Ця точність співрозмірна з точністю класичного алгоритму SVM, але використання знайдених на етапі кластернизації поверхневих точок процес навчання здійснюється в 3 рази швидше. Такий результат підтверджує ефективність розроблених алгоритмів кластеризації і класифікації об’єктів.

Більш значна кількість результатів досліджень, що проведилися в машинному тестуванні, показана на графіках в ілюстративній частині Додатка В.

3.5 Висновки до розділу

В даному розділі проведена розробка програмного забезпечення для реалізації запропонованих в роботі модифікованих алгоритмів кластеризації і класифікації об’єктів. Програмне забезпечення рзробленр у вигляді додатку ClusterBoundsSearcher_v0.0.2, який містить такі програмні модулі: модуль кластеризації введених даних з фіксацією граничних точок кластера, модуль вдосконаленого класифікатора SVM, модуль зменшення розмірності простору ознак до двохвимірного, який дозволяє візуалізувати отримані результат кластерного аналізу і класифікації. Шляхом проведення машинного експерименту були вибрані оптимальні параметри покращених алгоритмів кластерного аналізу і класифікації. Проведений експеримент засвідчив підвищення ефективності розроблених методів ідентифікації порівняно з існуючими.

4 ЕКОНОМІЧНА ЧАСТИНА

4.4 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку розробки і дослідження ефективних методів ідентифікації об’єктів в системах прийняття рішень. Особливістю програми є те, що розроблений програмний продукт більш ефективний, оскільки розроблений кластеризатор за методом поверхневого натягу находить поверхневі опорні точки кластерів для модифікованої програми класифікації SVM, в якій не треба тепер використовувати обчислювально-затратну процедуру пошуку поверхневих точок за методом множників Лагранжа.

На сьогодні аналогами програмного продукту для ідентифікації об’єктів в системах прийняття рішень є програма кластеризації методом C-means і програма-класифікатор методом опорних векторів SVM. Програма є безкоштовною, проте з дуже низьким рівнем точності ідентифікації об’єктів.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздат-ність в

Продовження табл. 4.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	4	4
Наявність аналогів на ринку	3	3	3
Цінова політика	1	1	1
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	2	2	2
Ринок збуту	4	3	4
Конкурентоспроможність	2	2	2
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	3	3	4
Сума	38	35	38
Середньоарифметична сума балів	(38+35+38) / 3 = 37		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 1 0	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є вище середньої, що досягається за рахунок того, що програмний продукт відрізняється від існуючих тим, що дана технологія Також слід зауважити, що розроблений програмний продукт побавлений основного недоліка конкурентів, а саме відсутності взаємозв'язку між ними, орієнтованого на використання класифікатором поверхневих точок кластерів для покращення класифікації. Розроблений програмний продукт більш ефективний, оскільки розроблений кластеризатор за методом поверхневого натягу находить поверхневі опорні точки кластерів для модифікованої програми класифікації SVM, в якій не треба тепер використовувати обчислювально-затратну процедуру пошуку поверхневих точок за методом множників Лагранжа.

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

4.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{M}{T_р} \cdot t, \quad (4.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$Т_р$ – число робочих днів в місяці, 22 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.1.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	44000	2000,00	30	60000,000
Програміст	40000	1818,18	30	54545,455
Всього				114545,45

Оскільки в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

4.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання

Додаткова заробітна плата прийнято розраховувати як 10 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 10\ \% / 100\ \% \quad (4.2)$$

$З_д = (114545,45 \cdot 10\ \% / 100\ \%) = 11454,55$ (грн.)

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\% \quad (4.3)$$

$Н_з = (114545,45 + 11454,55) \cdot 22\ \% / 100\ \% = 27720,00$ (грн.)

4.2.4 Витрати на матеріали і комплектуючі

Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючи, то витрати на матеріали і комплектуючі дорівнюють нулю.

4.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12} \text{ [грн.].} \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

Т – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 20000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,36 міс.

$$A_{обл} = \frac{20000}{2} \times \frac{1{,}36}{12} = 1136{,}36 грн.$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.2. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * Н_а * \frac{t_{вик.}}{12} \quad (4.5)$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю. В даному випадку використовуються при розробці безкоштовно надані програмні продукти.

Таблиця 4.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахуван-ня, грн.
Комп'ютер та комп'ютерна переферія	20000	2	1,36	1136,364
Офісне обладнення	20000	4	1,36	568,182
Приміщення	650000	20	1,36	3693,182
Всього				5397,73

5.2.6 Тарифи на електроенергію

Тарифи непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \quad (4.6)$$

де В – вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 6,2 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,4 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,9.

$В_е = 0,9 \cdot 0,4 \cdot 8 \cdot 30 \cdot 6,2 = 535,68$ (грн.)

5.2.7 Інші витрати та загальновиробничі витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.7)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$I_в = 114545,45 * 55\% / 100\% = 63000$ (грн.)

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.8)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$Н_{нзв}$ = 114545,45 * 110 % / 100 % = 126000 (грн.)

5.2.9 Витрати на проведення науково-дослідної роботи

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$В_{заг}$ = 114545,45+11454,55+27720,00+5397,73+535,68+63000+126000=
= 348653,41 грн.

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються *ЗВ*, визначається за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} \text{ (грн)}, \quad (5.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7;

впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$ЗВ$ = 348653,41 / 0,5 = 697307 грн.

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб’єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (4.10)$$

де $\pm\Delta Ц_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_o = Ц_б \pm \Delta Ц_o$;

$Цб$ – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році ϑ =18%.

Припустимо, що при прогнозованій ціні 100 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на *20* грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 50000 шт., протягом другого року – на 45000 шт., протягом третього року на 35000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$\Delta П_1 = (0*20 + (100 + 20)*50000)* 0,8333* 0,35) * (1 - 0,18) = 1195833,286$ грн.

$\Delta П_2 = (0*20 + (100 + 20)*(50000+45000)* 0,8333* 0,35) * (1 - 0,18) = 2726499,891$ грн.

$\Delta П_3 = (0*20 + (100 + 20)*(50000+45000+35000)* 0,8333* 0,35) * (1 - 0,18) = 3730999,851$ грн.

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 7653333,03 грн.

4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розраховуємо приведену вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad 4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05...0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

ПП=(1195833,286/(1+0,1)^1)+(2726499,891/(1+0,1)^2)+(3730999,851/(1+0,1)^3)= 1087121,17 + 2253305,695 + 2803155,41 = 6143582,274 грн.

Далі розраховують величину початкових інвестицій *PV*, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} * ЗВ, \quad (4.12)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}=2...5$, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

PV = 2 * 697307 = 1394613,63 *грн.*

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід *(NPV, Net Present Value)* для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \tag{4.13}$$

$E_{абс}$ = 6143582,274 – 1394613,63 = 4748968,64 грн.

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої но-рми дохідності *(IRR, Internal Rate of Return)* вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.14}$$

$T_ж$ – життєвий цикл наукової розробки, роки.

$E_в = \sqrt[3]{(1 + 4748968,64/1394613,63} - 1 = 0,639$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{4.15}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_в > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \tag{4.16}$$

$$T_{ок} = 1 / 0,639 = 1,56 \text{ р.}$$

Оскільки $T_{ок}$ < 3-х років, а саме термін окупності рівний 1,56 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновки до розділу

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 697307 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат

для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 1,56 роки.

ВИСНОВКИ

В магістерській кваліфікаційній роботі розроблено покращені методи кластерного аналізу і класифікації даних, що використовуються під час ідентифікації об’єктів для підтримки прийняття рішень в комп’ютерно-інтегрованих системах управління.

В першому розділі проведено аналіз сучасного стану проблеми ідентифікації об’єктів методами машинного навчання (ML) і їх використання методів ML для підвищення ефективності прийняття рішень в системах управління. В результаті проведеного аналізу обґрунтовано метод ідентифікації з використанням кластеризації і класифікації.

В другому розділі проведено математичне обґрунтування запропонованого методу кластеризації даних, що використовує фізичний принцип поверхневого натягу, та класифікатора типу SVM, модифікованого на його основі.

В третьому розділі проведена програмна реалізація запопонованих в роботі модифікованих алгоритмів кластерного аналізу і класифікації даних, що використовуються для ідентифікації об’єктів. Розроблено програмний додаток ClusterBoundsSearcher_v0.0.2 у складі: програмного модуля для кластерного аналізу даних з фіксацією граничних точок кластера, модуля модифікованого класифікатора SVM, модуля пониження розмірності простору ознак до двохмірного з метою візуалізації отриманих результатів кластеризації і класифікації.

Проведений машинний експеримент підтвердив адекватність розроблених методів і програмного забезпечення.

Розроблені методи і програмне забезпечення для ідентифікації об’єктів в системах управління дозволили підвищити такі показники ефективності прийняття рішень, як швидкість і якість.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://uk.wikipedia.org/wiki/Ідентифікація
2. https://pidru4niki.com/20080215/ekonomika/sposobi_metodi_identifikatsiyi
3. Черноруцкий И. Г. Методы принятия решений / И.Г. Черноруцкий. — СПб.: БХВ-Петербург, 2005. — 416 с.
4. Березін Б. О. Підтримка прийняття рішень при побудові систем довготермінового зберігання інформаці / Б.О. Березін, Качанов П. Т., В.В. Циганок, О. В.Андрійчук // Проблеми розвитку інформаційного суспільства: матеріали Міжнародного форуму. -2009.- С.145-153.
5. 3. Rao C., Govindaraju V. Handbook of Statistics: Machine Learning: Theory and Applications / C. Rao, V. Govindaraju. – Cambridge University Press, - 2013. - 552 с.
6. Шлезингер М., Главач В. Десять лекций по статистическому и структурному распознаванию / М. Шлезингер, В. Главач. – К.: Наукова думка, 2004. – 545 с.
7. Nils J. Nilson. Introduction to Mashine Learning / - [Електронний ресурс]. – Режим доступу: http://ai.stanford.edu/people/nilsson/mlbook.html .
8. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. / Стюарт Рассел, Питер Норвиг. — М.: Издательский дом "Вильямс", 2006. — 1408 с.
9. Looney C.G. Pattern Recognition Using Neural Networks / C. G. Looney –New York: Oxford University Press, 2004. – 449 p.
10. MacKay D. Informatiom Theory, Inference and Learning Algorithms / D. MacKay. – Cambridge University Press, 2003. – 640 p.
11. Воронцов К.В. Машинное обучение. Курс лекцій. - [Електронний ресурс]. – Режим доступу: https://ru.coursera.org/learn/vvedenie-mashinnoe-obuchenie
12. Глибовець М.М., Олецький О.В. Системи штучного інтелекту. — К.: КМ Академія, 2002. — 366 с.

13. Люгер Дж. Искусственный интеллект. Стратегии и методы решения сложных проблем. — М.: Вильямс, 2003. — 864 с.

14. Smith L. An Introduction to Neural Networks / Leslie Smith – University of Stirling, 2008. – [Електронний ресурс]. - Режим доступу: http://www.cs.stir.ac.uk/~lss/NNIntro/ InvSlides.html

15. Mitchell T.M. Mashine Learning / T.M. Mitchell. – McGraw-Hill Science, 2008. – 414 p.

16. Смолин Д.В. Введение в искусственный интеллект: конспект лекций. — М.: ФИЗМАТЛИТ, 2004. — 208 с..

17. Фор А. Восприятие и распознавание образов / Пер. с фран. А.В. Серединского под ред. Г.П. Катыса. – М.: Машиностроение, 2001. – 272 с.

18. Гренандер У. Лекции по теории образов: Пер. с англ. / Под ред. Ю Журавлева. - М.: Мир, 1981. - 445 с.

19. Фу К. Структурные методы в распознавании образов К. Фу - М.: Мир, 1997. - 320 с.

20. Фомин Я.А., Тарловский Г. Р. Статистическая теория распознавания образов / Я. А. Фомин, Г.Р. Тарловский. - М.: Радио и связь, 2001. – 264 с.

21. Фукунага К. Введение в статистическую теорию распознавания образов / К. Фукунага.- М.: Наука, 2001. - 368 с.

22. Потапова Р.К. Речевое управление роботом / Р.К. Потапова – М.: Радио и связь, 2000. – 248 с.

23. Tom M. Mitchell. Mashine Learning.- McGraw-Hill: Science/ Engineering/ Math, 1997. – p. 432.

24. N.M. Bykov, I.V. Kuzmin, A.I. Yakovenko. Development of effective strategy of pattern recognition. - Proceedings of SPIE, 2001, Vol. 4225, pp.76-83.

25. Steinwart I., Christmann A. Support Vector Machines / Ingo Steinwart, Andreas Christmann. - Springer-Verlag, New York, 2008. — 602 p.

26. Vapnik V., Chapelle O. Bounds on error expectation for support vector machines // Neural Computation. — 2000. — Vol. 12, no. 9. — Pp. 2013–2036.

27 Steinwart I., Christmann A. Support Vector Machines / Ingo Steinwart, Andreas Christmann. - Springer-Verlag, New York, 2008. — 602 p.

28. Ray S. Understanding Support Vector Machine algorithm from examples / Sunil Ray. – [Електронний ресурс] . – Режим доступу: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

29. [Електронний ресурс].– Режим доступу: https://habr.com/ru/post/428503/

30. [Електронний ресурс]. – Режим доступу: http://datascientist.one/support-vector-machines/

31. Fine S., Scheinberg K. INCAS: An incremental active set method for SVM: Tech. rep.: 2002. http://citeseer.ist.psu.edu/fine02incas.html.

32. Биков М.М., Волоський Б.О. Розробка ефективного класифікатора даних в інтелектуальних системах управління [Електронний ресурс] / М.М. Биков, Б.О. Волоський // Матеіали XLIX науково-технічної конференції підрозділів ВНТУ, Вінниця, 27-28 квітня2020 р. – Електр. текст. дані. – 2020. – Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/9730 .

33. Филиппович Роман, Резник Сьюзен М., Davatzikos Christos (2011). «Полуконтролируемый кластерный анализ данных визуализации» // NeuroImage. -54(3): 2185–2197.- doi:10.1016 j.neuroimage.2010.09.074. PMC 3008313. PMID20933091.

34. Братко И. Алгоритмы искусственного интеллекта на языке Пролог. — М.: Вильямс, 2004. — 640 с.

35. Наследов А. Д. SPSS: Компьютерный анализ данных в психологии и социальных науках / А.Д. Наследов. – СПб.: Питер, 2004. – 416 с.

36. Бериков В.Б., Лбов Г.С. Современные тенденции в кластерном анализе. / В. Б. Бериков, Г. С. Лбов // Новосибирск: ин-т математики им. С. Л. Соболева. СО РАН, 2008. — 26 с.

37. Нейский, И.М. Классификация и сравнение методов кластеризации / И.М. Нейский // Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. – М.: Изд-во ООО “Эликс +”, 2008. – Выпуск 8. – С. 111-122.

38. Миркин Б.Г. Методы кластер-анализа для поддержки принятия решений: обзор / - серия WP7 «Математические методы анализа решений в экономике, бизнесе и политике» / Ф.Т. Алесеров, В.В. Подиновский, Б.Г. Миркин // НИУ «Высшая школа экономики», 2011. - 88 стр.

39. Болдак А. А. Определение количества кластеров в статистических данных / А. А. Болдак, Д. Л. Сухарев // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка : збірник наукових праць. – 2011. – № 53. – С. 118–122.

40. Биков М.М., Задачін А.С. Розробка програмного забезпечення кластеризації даних в інтелектуальних системах прийняття рішень. - Електронне наукове видання матеріалів конференції, м. Вінниця, 2021. – Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2021/paper/view/12437.

41. Rao C., Govindaraju V. Handbook of Statistics: Machine Learning: Theory and Applications, 2013. 552 c.

42. Введение в машинное обучение с помощью Python и Scikit-Learn. URL: https://habr.com/ru/company/ mlclass/blog/247751/

43. Классификация в Python с Scikit-Learn и Pandas. URL: https://stackabuse.com/classification-in-pythonwith-scikit-learn-and-pandas/ .

44. Обзор методов классификации в машинном обучении с помощью Scikit-Learn. URL: https://tproger. ru/translations/scikit-learn-in-python/s://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/

ДОДАТКИ

Додаток А

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Розробка і дослідження ефективних методів ідентифікації об'єктів в системах прийняття рішень»

Тип роботи: Магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ КСУ, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 81,6% Схожість 18,4%

Аналіз звіту подібності (відмітити потрібне)

☐ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Галущак А.В.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichek щодо роботи.

Автор роботи _____ Токаренко В.М.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Биков М.М.
(підпис) (прізвище, ініціали)

Додаток Б
(обов'язковий)

ВНТУ

ЗАТВЕРДЖЕНО
Завідувач кафедри КСУ
д.т.н., професор
_____ В.В. Ковтун
" 30 " жовтня 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

РОЗРОБКА І ДОСЛІДЖЕННЯ ЕФЕКТИВНИХ МЕТОДІВ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ В СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ

08-33.МКР.009.00.000 ТЗ

Студентка групи 2 АКІТ-21 м

Підпис Вероніка ТОКАРЕНКО
Ім'я ПРІЗВИЩЕ

Керівник к.т.н., доцент

Підпис Микола БИКОВ
Ім'я ПРІЗВИЩЕ

Вінниця 2022

1.Назва та галузь застосування

1.1. Назва – Розробка програмного забезпечення кластеризації і класифікації даних в системах прийняття рішень.

1.2. Галузь застосування – Комп'ютеризовані системи прийняття рішень і управління.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ від "14" вересня 2022 року №203

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є підвищення ефективності систем прийняття рішень за рахунок розробки покращених методів кластеризації і класифікації.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск / К.Д. Маннинг, П. Рагхаван, Х. Шютце — М.: Изд-во "Вильямс", 2011. – 528 с.
2. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python / Андреас Мюллер, Сара Гвидо. – Москва, 2017. – 393 с.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- введення даних;
- кластеризація даних з фіксацією поверхневих точок кластера;

-класифікацію даних за методом SVM з використанням поверхневих точок знайдених кластерів;

- обґрунтування доцільності розробки багатовимірне шкалювання і візуалізація точок кластерів на площині.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- Visual Studio 2020;
- Intellij IDEA Community Edition;
- Eclipse IDE.

5.2.2. Умови експлуатації системи:

- робота на стандартних ПЕОМ в приміщеннях зі стандартними умовами;
- можливість цілодобового функціонування системи;
- текст програмного забезпечення системи є цілком закритим.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Багатоваріантний аналіз методів машинного навчання, обґрунтування доцільності розробки «29» вересня 2022 р.
2. Розробка математичного обгрунтування досліджуваного методу ідентифікації «15»жовтня 2022 р.
3. Розробка алгоритмів функціонування системи «25»жовтня 2022 р.
4. Розробка програмного забезпечення системи «10» листопада 2022 р.
5. Тестування програмного забезпечення «20» листопада 2022 р.
6. Оформлення пояснювальної записки і графічного матеріалу «14» грудня2022 р.

6.2 Графічні матеріали:

1. Розробка схем алгоритмів «5 » грудня 2022 р.

2. Розробка плакатів демонстраційних «10» грудня 2022 р.
3. Розробка UML-діаграм системи «12»грудня 2022 р.

7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до 14» грудня 2022 р.

7.2. Атестація роботи здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до 17» грудня 2022 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК.

7.4. Захист магістерської кваліфікаційної роботи провести «22» грудня 2022 р.

Додаток В
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА І ДОСЛІДЖЕННЯ ЕФЕКТИВНИХ МЕТОДІВ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ В СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ

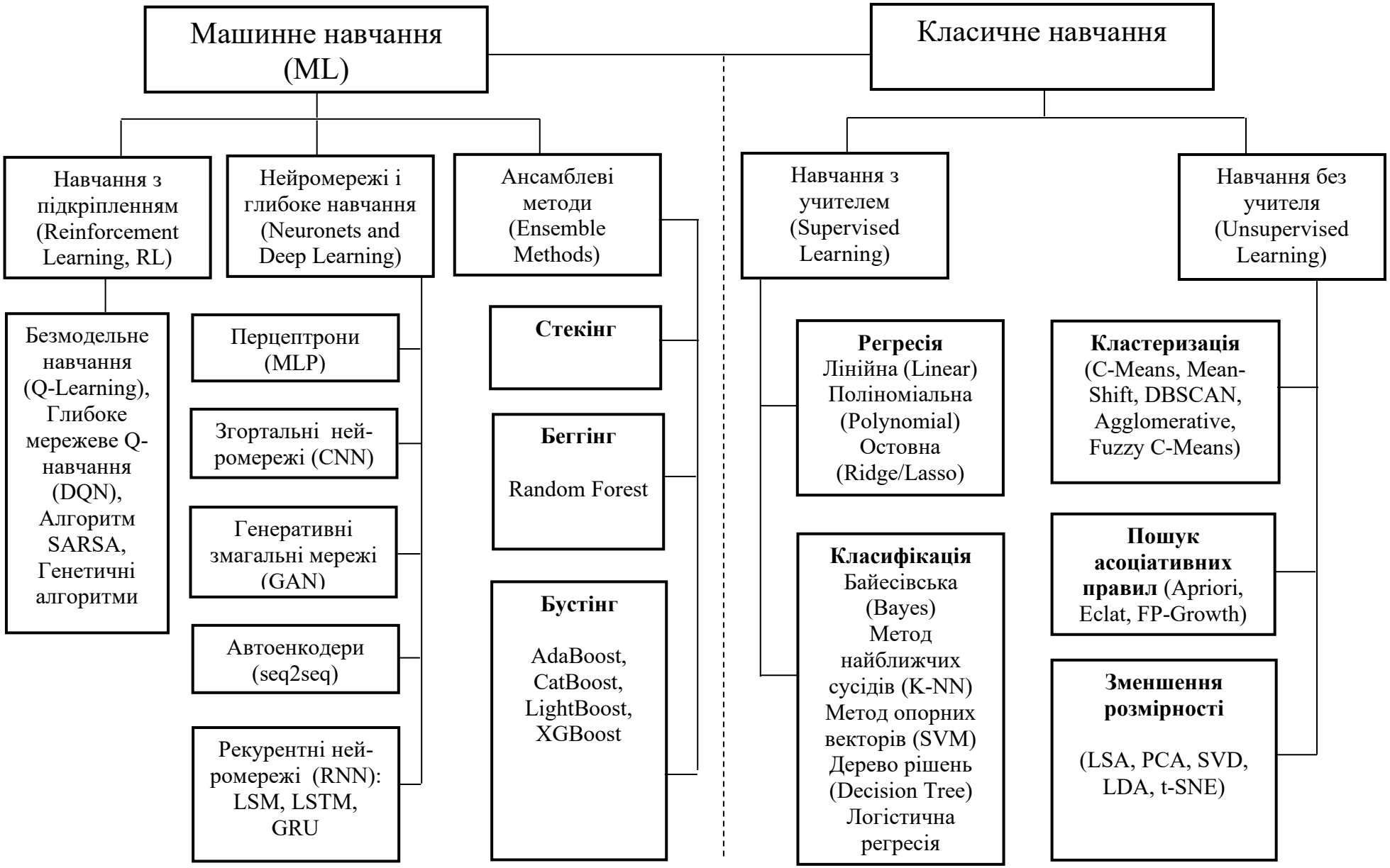
Студентка групи 2АКІТ-21м

Підпис Вероніка ТОКАРЕНКО
Ім'я ПРІЗВИЩЕ

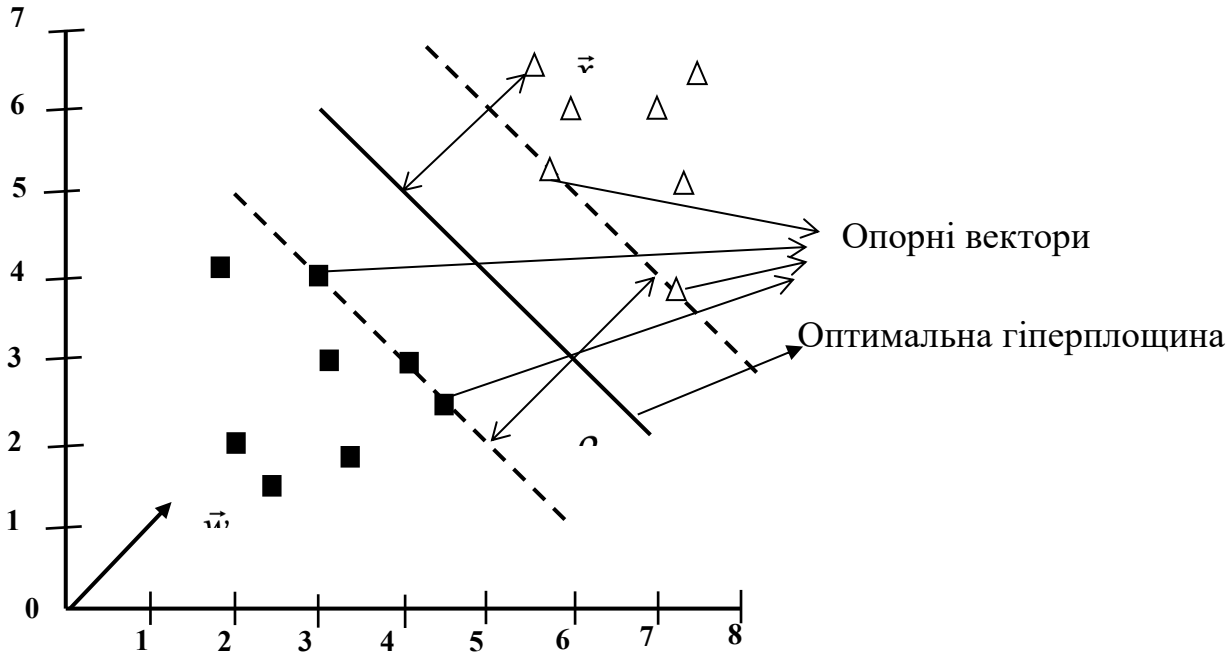
Керівник к.т.н., доцент, проф. кафедри КСУ

Підпис Микола БИКОВ
Ім'я ПРІЗВИЩЕ

КЛАСИФІКАЦІЙНА СХЕМА МЕТОДІВ МАШИННОГО НАВЧАННЯ



ГЕОМЕТРИЧНА ІНТЕРПРЕТАЦІЯ КЛАСИФІКАЦІЇ МЕТОДОМ ОПОРНИХ ВЕКТОРІВ

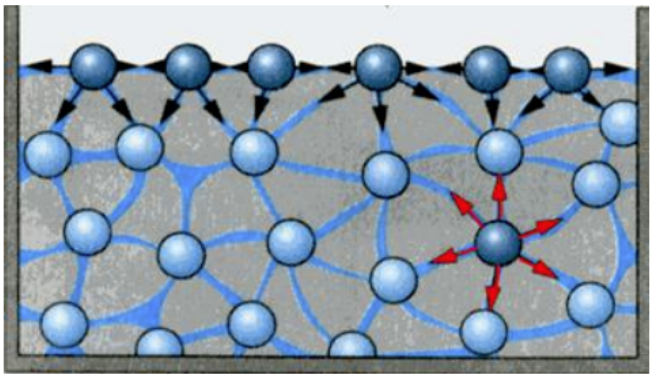


$$r = \frac{2}{\|w\|}$$

Задача оптимізації – пошук вагового вектора, що максимізує r :

$$\frac{1}{2}\|w\|^2 \to \min \text{ при обмеженнях: } y_i \cdot (\vec{w}^T \vec{x}_i + w_0) \geq 1, \quad i = 1,...,m.$$

ГЕОМЕТРИЧНА ІНТЕРПРЕТАЦІЯ МЕТОДУ ПОВЕРХНЕВОГО НАТЯГУ



$f(\vec{x}_i, \vec{x}_j) = \frac{p}{d}$ - сила взаємодії між точками,

p - ваговий коефіцієнт,

$d = \sqrt{\sum_{k=1}^{n} x_{ik}^2}$ - відстань між точками,

x_{ik} - k-а координата i-ої точки (вектора) вибірки.

$\Delta_m = d(\vec{X}_m, \vec{Z}_m) = \frac{\sqrt{\sum_{n=1}^{N} (x_{mn} - z_{mn})^2}}{d_m}$ - сила поверхневого натягу точок

якщо $\Delta_m > \sigma$, то точка $\vec{X}_m$ є крайньою в кластері

epsilon=σ – стрес, заданий поріг поверхневого натягу

СХЕМА АЛГОРИТМУ КЛАСИФІКАЦІЇ МОДИФІКОВАНИМ МЕТОДОМ SVM

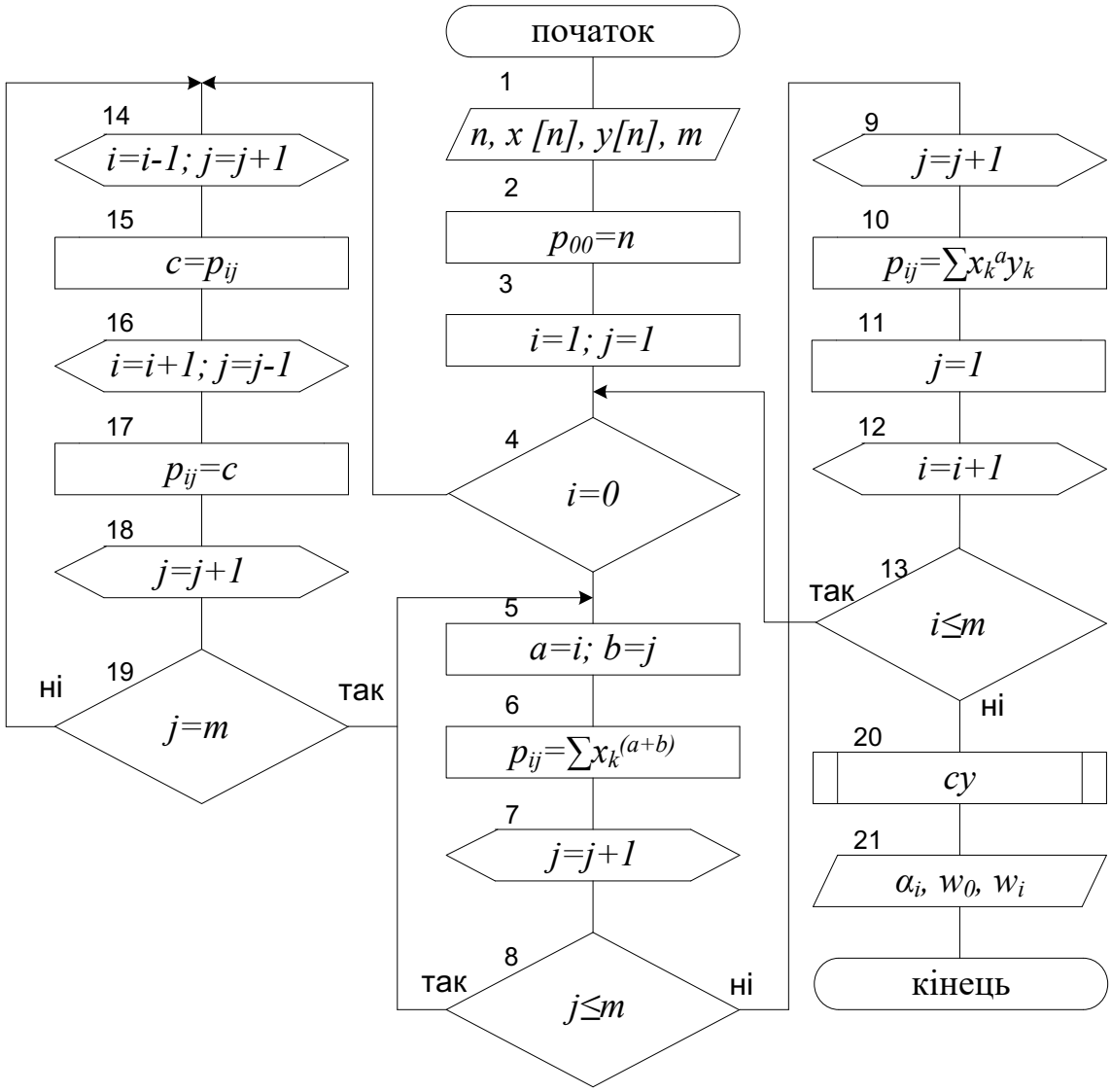
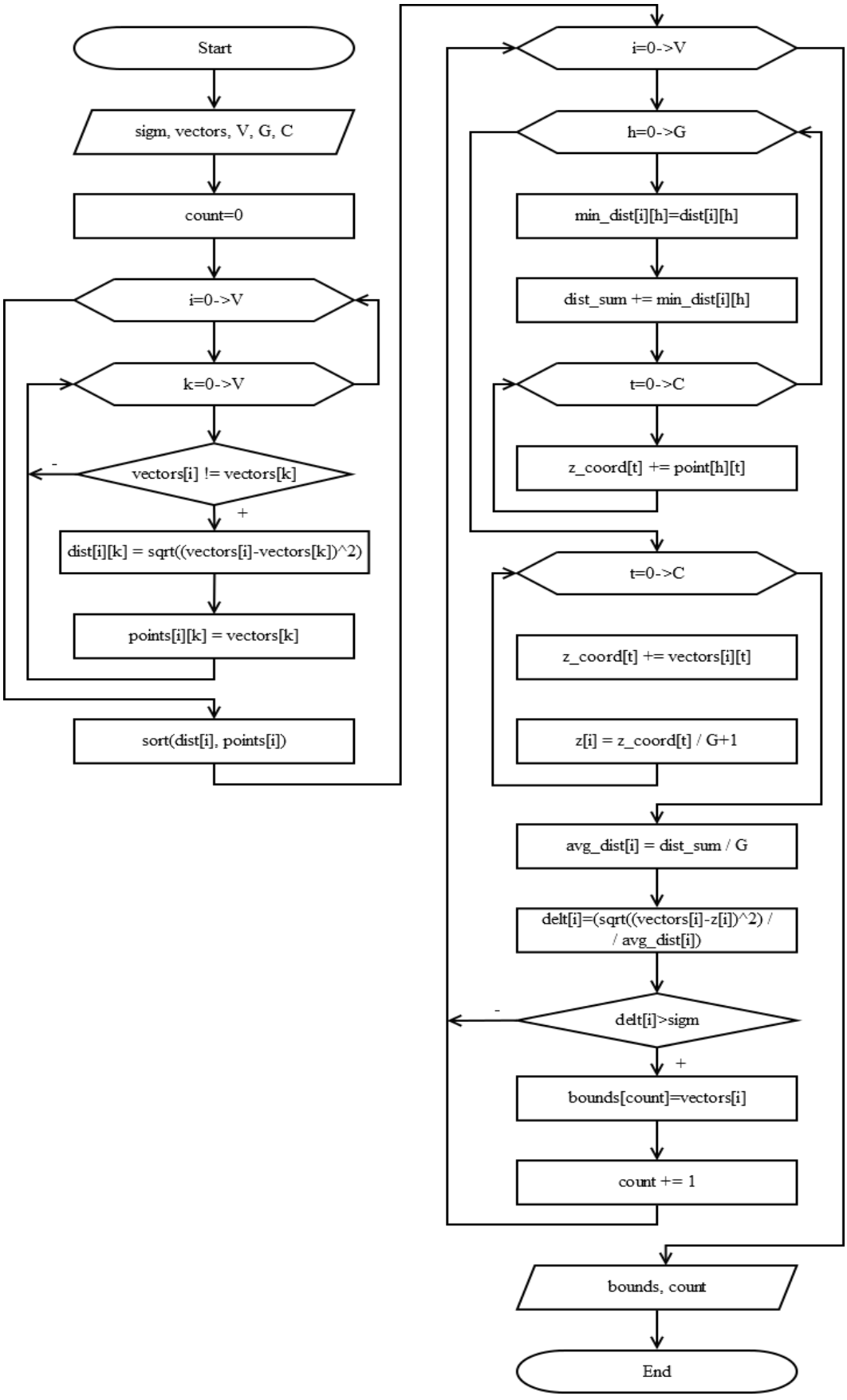
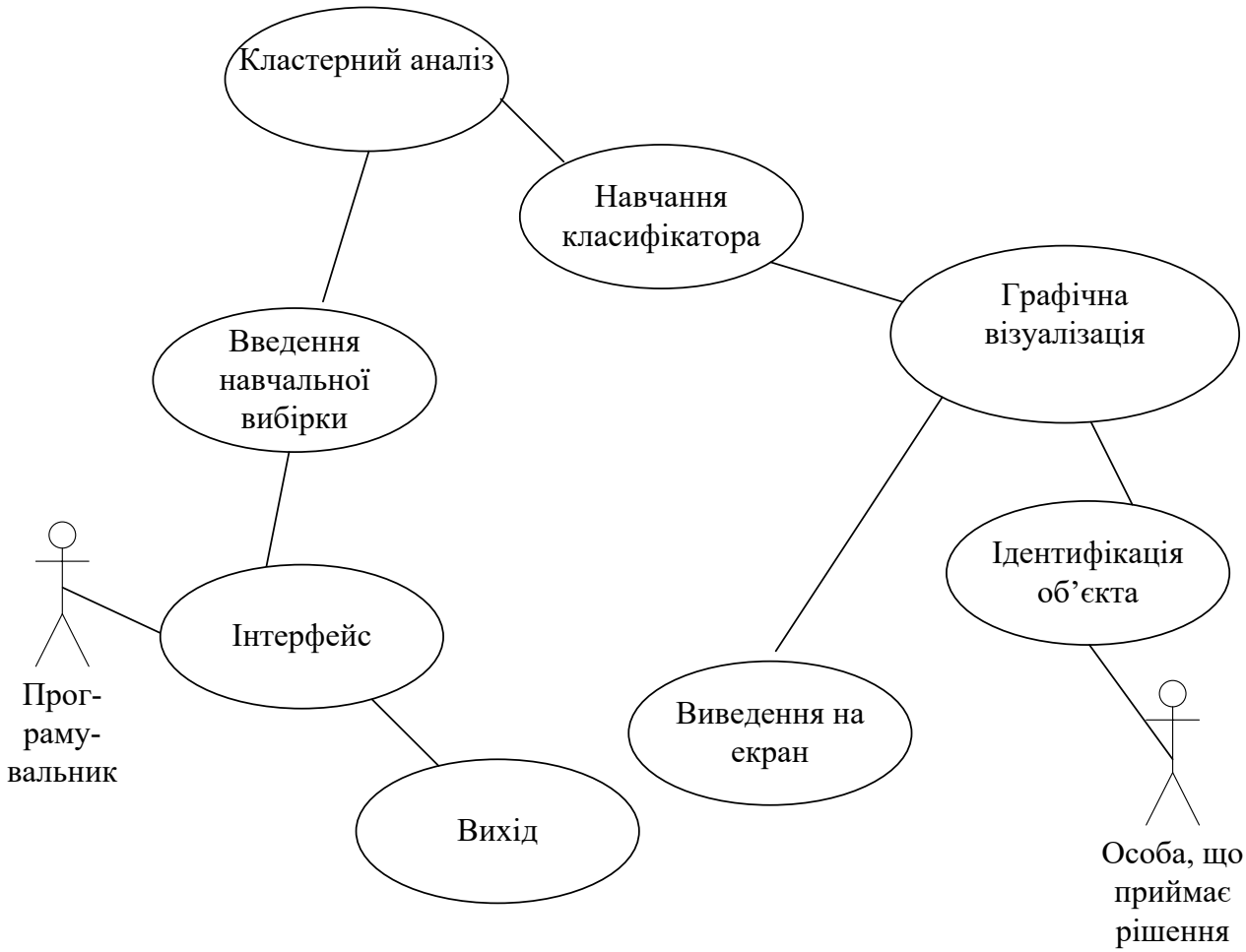


СХЕМА АЛГОРИТМУ ЗНАХОДЖЕННЯ ПОВЕРХНЕВИХ ТОЧОК КЛАСТЕРА



UML-ДІАГАРАМА ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



ЕКРАН З ВМІСТОМ ПРОГРАМНОГО МОДУЛЯ

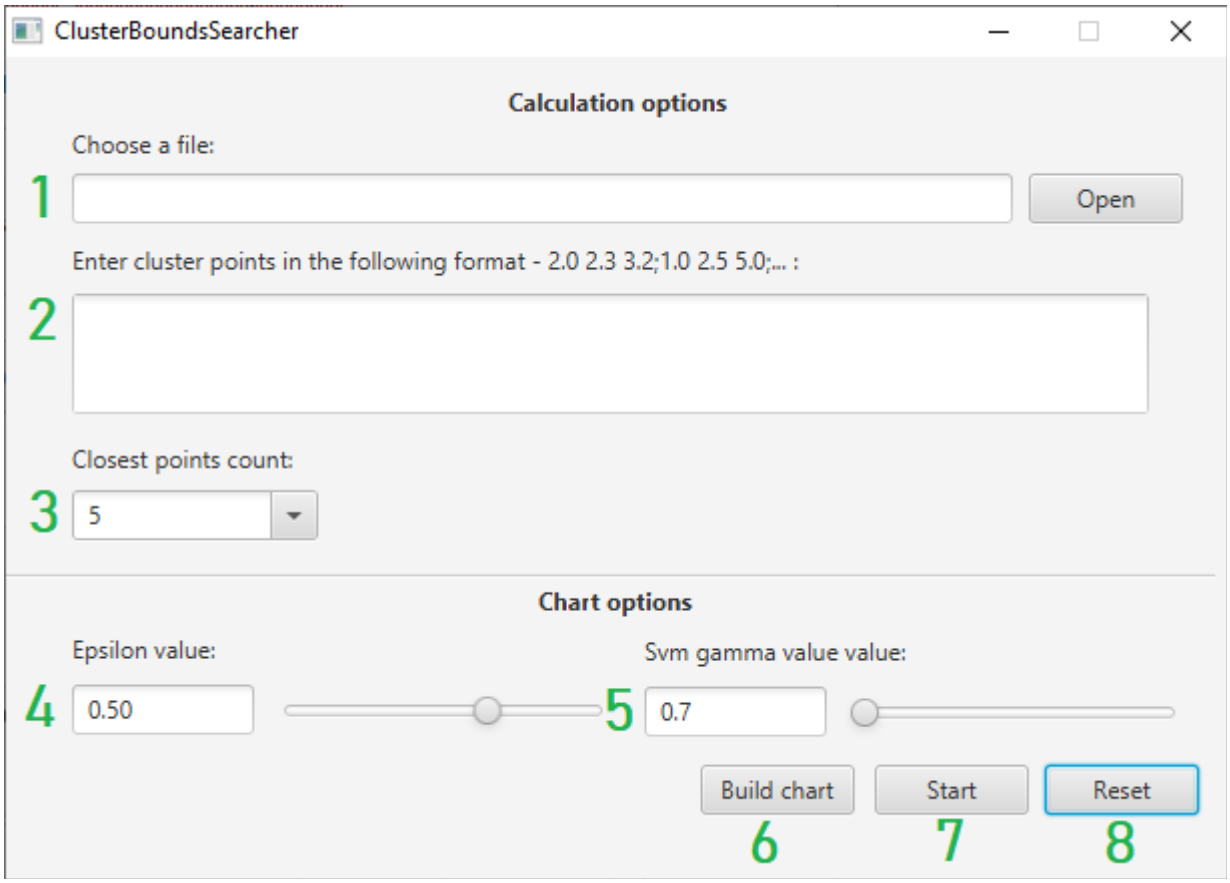
scripts		<Папка>	29.05.2021 14:47
ClusterBoundsSearcher	jar	26 133	29.05.2021 14:45
Instruction_Cluster_bound_points	pdf	149 472	29.05.2021 15:01
run	bat	94	30.04.2021 12:18
run_additional	bat	87	30.04.2021 12:20

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.figure
import numpy as np
import csv
from sklearn.manifold import MDS
from sklearn.decomposition import PCA
from sklearn import svm
import argparse


parser = argparse.ArgumentParser(description='Process data from File')
parser.add_argument('path')
parser.add_argument('svm_gamma')
args = parser.parse_args()
path = args.path
svm_gamma = float(args.svm_gamma)


def transform_to_form(list, dimension:int):
    transformed = [0]*len(list[0])
    for idx in range(0, len(list[0])):
```

ГРАФІЧНИЙ ІНТЕРФЕЙС ПРОГРАМИ



1. “Choose a file” - введення даних у вигляді файлу
2. “Enter clusters point in the following format” - введення даних з клавіатури
3. “Closest point count” - вибір кількості найближчих точок
4. “Epsilon value” – вибір порогового значення стресу
5. “SVM gamma value” – вибір параметра GAMMA для класифікатора SVM
6. Клавіша “Build chart” - запускає процес побудови графіків.
7. Клавіша “Start” - запускає процес кластеризації і класифікації та запис результатів у відповідні файли
8. Клавіша “Reset” - повертає усі значення полів графічного інтерфейсу на значення за замовченням

РЕЗУЛЬТАТИ МАШИННОГО ЕКСПЕРИМЕНТУ

Дослідження методу кластеризації

При значеннях epsilon менше 0,35 та більше 0,50 точки та їх кількість неправильні

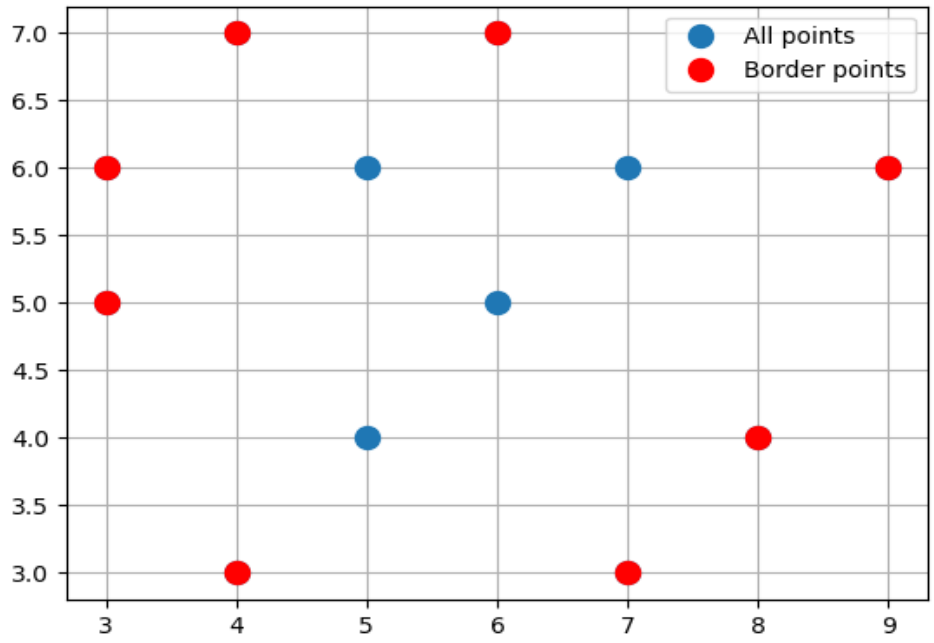


Рисунок 1 - Поверхневі точки при sigma = 0,35

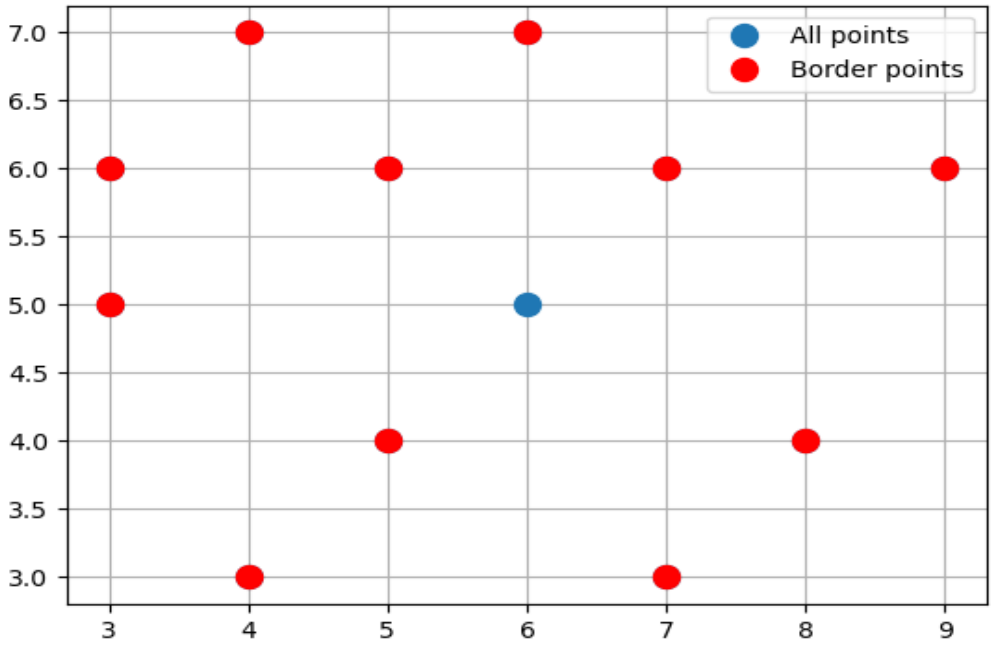


Рисунок 2 - Поверхневі точки при sigma = 0,15

Дослідження методу класифікації

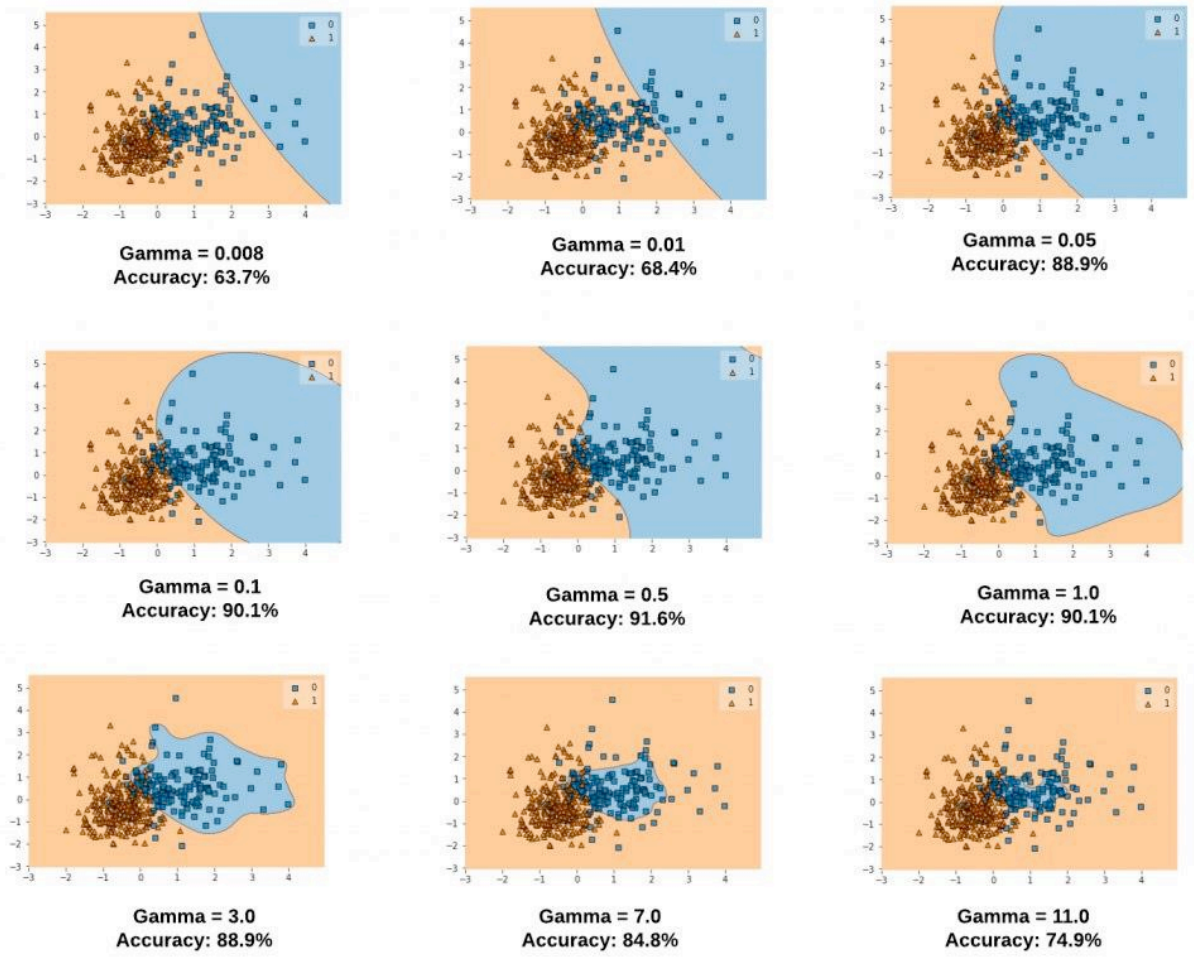


Рисунок 3 – Результати класифікації точок файлу “iris.dat” при різних значеннях парамету GAMMA

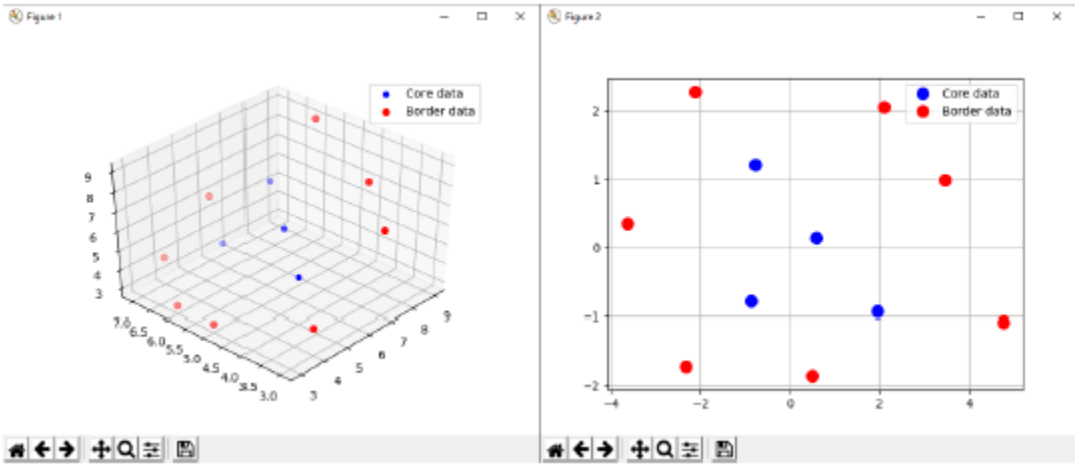


Рисунок 4 – Результати проектування точок на трьохмірний і двомірний простори

Додаток В
(довідниковий)

Лістинги програм

Лістинг програми кластеризації з фіксацією поверхневих точок кластера

```
public class ClusterBoundPointFinder {

    List<Point<Float>> inputVectors = Arrays.asList(
            new Point<>(Arrays.asList(3F, 5F)),
            new Point<>(Arrays.asList(3F, 6F)),
            new Point<>(Arrays.asList(4F, 3F)),
            new Point<>(Arrays.asList(4F, 7F)),
            new Point<>(Arrays.asList(5F, 4F)),
            new Point<>(Arrays.asList(5F, 6F)),
            new Point<>(Arrays.asList(7F, 3F)),
            new Point<>(Arrays.asList(6F, 5F)),
            new Point<>(Arrays.asList(6F, 7F)),
            new Point<>(Arrays.asList(8F, 4F)),
            new Point<>(Arrays.asList(7F, 6F)),
            new Point<>(Arrays.asList(9F, 6F))
    );
    private int corePointCount = 5;
    private final float deviation;
    private final int coordinatesCount;

    public ClusterBoundPointFinder(List<Point<Float>> inputVectors, float deviation) {
        this.inputVectors = inputVectors;
        this.deviation = deviation;
        this.coordinatesCount = inputVectors.get(0).getCoordinates().size();
    }

    public ClusterBoundPointFinder(float deviation) {
```

```
        this.deviation = deviation;
        this.coordinatesCount = this.inputVectors.get(0).getCoordinates().size();
    }

    private Point<Float> calculateCentre(Set<Point<Float>> points, Point<Float> keyPoint) {
        final ArrayList<Point<Float>> allPoints = new ArrayList<>(points);
        allPoints.add(keyPoint);

        return new Point<>(IntStream.range(0, coordinatesCount)
                .mapToDouble(i -> allPoints.stream()
                        .mapToDouble(point -> Double.valueOf(point.getCoordinates().get(i)))
                        .average().getAsDouble())
                .boxed()
                .map(Double::floatValue)
                .collect(Collectors.toList()));
    }

    private double calculatePointOffset(Point<Float> point, Point<Float> centre, Double
meanDistance) {
        return calcDistance(point, centre) / meanDistance;
    }

    protected double calcDistance(Point<Float> pointA, Point<Float> pointB) {
        return Math.sqrt(IntStream.range(0, pointA.getCoordinates().size())
                .mapToDouble(i -> pointA.getCoordinates().get(i) - pointB.getCoordinates().get(i))
                .map(val -> Math.pow(val, 2))
                .sum());
    }

    private void calculateBoundPoints() {
        Map<Point<Float>, Map<Point<Float>, Double>> distances = new HashMap<>();
        inputVectors.forEach(point -> distances.put(point, inputVectors.stream()
                        .filter(innerPoint -> !innerPoint.equals(point))
                        .collect(Collectors.toMap(innerPoint -> innerPoint, innerPoint ->
calcDistance(point, innerPoint)))));
```

```
        Map<Point<Float>, Map<Point<Float>, Double>> closestDistances =
distances.keySet().stream()
                .collect(Collectors.toMap(key -> key, key -> distances.get(key).entrySet().stream()
                        .sorted(Comparator.comparingDouble(Map.Entry::getValue))
                        .limit(corePointCount)
                        .collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue))));


        Map<Point<Float>, Double> meanDistances = closestDistances.keySet().stream()
                .collect(Collectors.toMap(key -> key, key ->
closestDistances.get(key).values().stream()
                        .mapToDouble(value -> value)
                        .average().getAsDouble()));


        Map<Point<Float>, Point<Float>> pointCentres =
closestDistances.entrySet().stream().collect(Collectors.toMap(Map.Entry::getKey, entry ->
calculateCentre(entry.getValue().keySet(), entry.getKey())));

        Map<Point<Float>, Double> pointOffsets =
pointCentres.entrySet().stream().collect(Collectors.toMap(Map.Entry::getKey, entry ->
calculatePointOffset(entry.getKey(), entry.getValue(), meanDistances.get(entry.getKey()))));

        Map<Point<Float>, Double> result = null;

        for (float i = deviation; i < 1; i+=0.05) {
            final float f = i;
            result = pointOffsets.entrySet().stream().filter(entry -> entry.getValue() >
f).collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue));
            System.out.println(String.format("RESULT(size = %d, deviation=%.2f) = %s",
result.size(), f, result));
        }
    }
```

```
    public static void main(String[] args) {
        ClusterBoundPointFinder clusterBoundPointFinder = new
ClusterBoundPointFinder(0.15F);
        clusterBoundPointFinder.calculateBoundPoints();
    }
}
```

Лістинг програми класифікації даних і графічної візуалізації

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.figure
import numpy as np
import csv
from sklearn.manifold import MDS
from sklearn.decomposition import PCA
from sklearn import svm
import argparse

parser = argparse.ArgumentParser(description='Process data from File')
parser.add_argument('path')
parser.add_argument('svm_gamma')
args = parser.parse_args()
path = args.path
svm_gamma = float(args.svm_gamma)

def transform_to_form(list, dimension:int):
    transformed = [0]*len(list[0])
    for idx in range(0, len(list[0])):
        tmp = [0]*dimension
        for kdx in range(0, dimension):
            tmp[kdx] = list[kdx][idx]
        transformed[idx] = tmp
    return transformed

with open(path, "r") as fd:
```

```
    reader = csv.reader(fd)
    i = 0
    k = 0
    dimensions = int(float(reader.__next__()[0]))
    coord = [[0]] * dimensions
    border_coord = [[0]]*dimensions
    for row in reader:
        print(f"row={row}")
        if i < dimensions:
            coord[i] = [float(str) for str in row]
            i += 1
        else:
            border_coord[k] = [float(str) for str in row]
            k += 1

print(f"coordinates={coord}")
print(f"border_coord={border_coord}")

if dimensions <= 3:
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.scatter(coord[0], coord[1], coord[2], s=30, label='Core data', c='blue')
    ax.scatter(border_coord[0], border_coord[1], border_coord[2], s=30, label='Border data',
        c='red')
    plt.legend()
    plt.grid(True)

# transform to point form
coord_form = transform_to_form(coord, dimensions)
border_coord_form = transform_to_form(border_coord, dimensions)
print(f"coordinates_form={coord_form} ({len(coord_form)})")
print(f"border_coord_form={border_coord_form} ({len(border_coord_form)})")
points = coord_form + border_coord_form
print(f"points={points}")

# dimensional reduction by pca
reduction_density = 2
```

```
pca = PCA(n_components=reduction_density)
coord_transformed_pca = pca.fit_transform(coord_form)
print(coord_transformed_pca.shape)
print(coord_transformed_pca)

border_coord_transformed_pca = pca.fit_transform(border_coord_form)
print(border_coord_transformed_pca.shape)
print(border_coord_transformed_pca)
points_pca = pca.fit_transform(points)
print(f"points_pca={points_pca.shape}")
print(f"points_pca={points_pca}")
fig1 = plt.figure()
plt.plot(points_pca[0:len(coord_form), 0], points_pca[0:len(coord_form), 1], 'bo', label='Core
    data', markersize=10)
plt.plot(points_pca[len(coord_form)+1:len(points_pca), 0],
    points_pca[len(coord_form)+1:len(points_pca), 1], 'ro', label='Border data',
    markersize=10)
plt.legend()
plt.grid(True)
points_pca_ = np.array(points_pca)
groups = np.array([0] * len(coord_form) + [1] * len(border_coord_form))
print(f"groups={groups}")

# building SVM graph
C = 1.0
# SVM regularization parameter
svc = svm.SVC(kernel='rbf', gamma=svm_gamma, C=C).fit(points_pca, groups)
# create a mesh to plot in
x_min, x_max = points_pca_[:, 0].min() - 1, points_pca_[:, 0].max() + 1
y_min, y_max = points_pca_[:, 1].min() - 1, points_pca_[:, 1].max() + 1
h = abs((x_max / x_min)/100)
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
np.arange(y_min, y_max, h))

fig_svm = plt.figure()
plt.subplot(1, 1, 1)
Z = svc.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
```

```
plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=0.8)
plt.scatter(points_pca_[:, 0], points_pca_[:, 1], c=groups, cmap=plt.cm.Paired)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.xlim(xx.min(), xx.max())
plt.title('SVC with rbf kernel')
plt.legend()
plt.grid(True)
plt.show()
```