Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДІВ ВІДБОРУ ОЗНАК ДЛЯ КЛАСИФІКАЦІЇ ТА КЛАСТЕРИЗАЦІЇ ДАНИХ

(тема)

Виконала:
студент 2 курсу, групи ІНФм-23-1
Сотникова А.В.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Машталір В.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Сотниковій Анастасії Віталіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів відбору ознак для класифікації та кластеризації даних

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 03 січня 2025 р.

3. Вихідні дані до роботи математичні методи відбору ознак для класифікації та кластеризації даних, перелік використовуваних програмних засобів, теоретичні відомості про існуючі методи відбору ознак для класифікації та кластеризації та принципи їх роботи.

4. Перелік питань, що потрібно опрацювати в роботі

1. Огляд існуючих методів та підходів до відбору ознак для кластеризації та класифікації даних.

2. Опис теоретичних відомостей обраних для порівняння методів відбору ознак, а саме створення аналізу принципу дії фільтрових, обгорткових та вбудованих методів відбору ознак для класифікації та кластеризації.

3. Опис принципів створення порівняння переліку обраних методів відбору ознак для задач класифікації та кластеризації.

4. Створення програмної реалізації для обраних методів відбору ознак для задач класифікації та кластеризації.

5. Створення аналізу та підсумків за результатами тестування фільтрових, вбудованих та обгорткових методів відбору ознак для класифікації та кластеризації даних.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) схеми принципу дії фільтрових, обгорткових та вбудованих методів відбору ознак; представлення коду та результатів виконання коду, отримані матриці помилок класифікації для розроблених методів відбору ознак.

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	26.11.24-27.11.24	
3	Аналіз літератури з досліджуваної проблеми	28.11.24-29.11.24	
4	Аналіз існуючих методів відбору ознак для класифікації та кластеризації	30.11.24-02.12.24	
5	Розробка обраних методів відбору ознак та способів їх порівняння	03.12.24-05.12.24	
6	Програмна реалізація обраних методів відбору ознак та способів їх порівняння	06.12.24-10.12.24	
7	Оформлення пояснювальної записки	11.12.24-21.12.24	
8	Перевірка на плагіат	22.12.2024	
9	Рецензування	23.12.2024	
10	Підготовка презентації та доповіді	24.12.2024	
11	Занесення роботи в електронний архів	04.01.2025	
12	Попередній захист кваліфікаційної роботи	13.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____Сотникова А.В._____
(підпис)

Керівник роботи _____ _____проф. Машталір В.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 99 с., 1 табл., 47 рисунку,
41 джерело.

ГІБРИДНІ МЕТОДИ, ВБУДОВАНІ МЕТОДИ, КЛАСИФІКАЦІЯ ДАНИХ, КЛАСТЕРИЗАЦІЯ ДАНИХ, ОБГОРТКОВІ МЕТОДИ, РЕЛЕВАНТНА ОЗНАКА, ТОЧНІСТЬ МОДЕЛІ, ФІЛЬТРОВІ МЕТОДИ.

Об’єктом дослідження є масиви даних, які містять різнорідні ознаки, які необхідно відібрати для кластеризації або класифікації.

Метою дослідження є дослідження підходів до відбору ознак, що підвищують ефективність роботи методів та дозволяють точно групувати об’єкти для кластеризації та дають найвищу точність класифікації даних.

У ході проведення дослідження було використано методи числового моделювання та аналітичного обґрунтування. Проведено дослідження методів відбору релевантних ознак на основі принципів фільтрових, вбудованих та обгорткових методів відбору ознак, результати роботи обраних для дослідження методів було оцінено за допомогою ряду релевантних метрик.

У результаті дослідження було створено програмну реалізацію кожного методу на обраному високонавантаженому наборі даних.

HYBRIG METHODS, EMBEDDED METHODS, DATA CLASSIFICATION, DATA CLUSTERING, WRAPPER METHODS, RELEVANT FEATURE, MODEL ACCURACY, FILTER METHODS

The object of the research consists of data arrays containing heterogeneous features that need to be selected for clustering or classification.

The aim of the research is to study feature selection approaches for classification and clustering that enhance method efficiency and allow for accurate grouping of objects for clustering, while achieving the highest classification accuracy.

During the research, numerical simulations and analytical reasoning were applied. The research included exploring relevant feature selection methods based on the principles of filter, embedded and wrapper feature selection techniques The performance of the selected methods was evaluated using a set of relevant metrics.

As a result of the research, a software implementation of each feature selection method based on the identified principles was created, and an evaluation of the selected methods was conducted on a high-load dataset.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ARD – Automatic Relevance Determination (метод виділення ознак шляхом регресії автоматичного виділення релевантності)

Accuracy – Метрика точності класифікації, частка правильно класифікованих об’єктів від їх загальної частки

DataFrame – Таблична структура, представлення даних у Python

Chi-squared Score – Метод відбору ознак за критерієм Хі-квадрат

CFS – Correlation-based feature selection (метод відбору ознак на основі кореляції)

Elastic Net – Метод відбору ознак шляхом еластичної сітки

Laplacian Score – Метод відбору ознак за критерієм Лапласа

LASSO – Least Absolute Shrinkage and Selection Operator (метод відбору релевантних ознак за допомогою оператора найменшого абсолютного стиснення та відбору)

MAD – Mean Absolute Difference (метод відбору ознак на основі значення середньої абсолютної різниці)

MCFS – Multi-Cluster Feature selection (метод мультикластерного відбору ознак)

Precision – точність передбачення у ході класифікації, частка правильно передбачених позитивних результатів від загальної кількості позитивних результатів

Recall – повнота результатів класифікації, частка правильно передбачених позитивних об’єктів серед усіх фактичних позитивних об’єктів

RFE – Recursive Feature Elimination (метод відбору ознак шляхом рекурсивного виключення)

Ridge regression – Метод відбору ознак за допомогою гребневої регресії

Relief – Метод відбору ознак із випадковим вибором примірників

ROC-AUC – Area Under the ROC Curve (площа під кривою помилок)

SMLR – Sparse Multinomial Logistic Regression (метод відбору ознак за допомогою розрідженої мультиноміальної логістичної регресії)

Variance Threshold – Метод відбору ознак на основі значення дисперсії

Fisher Score – Метод відбору ознак за критерієм Фішера

F1-Score – метрика балансування відношення між точністю передбачення та повнотою передбаченого результату

ВСТУП

Будь-який процес обробки даних включає у себе етапи виділення значущих та менш значущих факторів, або ознак даних, які впливають на розглянуте питання. Не винятком є задачі класифікації та кластеризації даних.

Процес класифікації даних – це процес, під час якого дані розподіляються до відповідних виділених груп, відповідно до їх наявних характеристик. Даний процес широко використовується наразі у всіх сферах життя. Прикладами можуть бути алгоритми визначення спаму за вмістом листа, або системи, які дозволяють визначати ризики захворювань на основі даних медичних обстежень.

Процес кластеризації даних – це процес, під час якого дані розподіляються на групи відповідно їх характеристик. Відмінністю є те, що для задач кластеризації групи, на які поділяються дані, не відомі, а дані відносяться до груп, кластерів, за сукупністю спільних або схожих значень критеріїв. Прикладами задач класифікації у реальному житті є виділення сегментів ринку реалізації певного продукту або виділення цільових груп користувачів за критеріями.

Незважаючи на кардинально різні приклади застосування класифікації та кластеризації даних, дані процеси мають спільну частину, а саме попередню підготовку даних. Для того, щоб алгоритми кластеризації та класифікації дали належний результат, кількість ознак повинна бути мінімальною та максимально інформативною для задачі.

Таким чином, постановку задачі підбору ознак для класифікації або кластеризації даних є формування такого переліку ознак, який є достатнім для надання моделлю найкращого результату роботи.

Розглянуте питання є особливо актуальним наразі, оскільки зі зростаючими об’ємами даних, які необхідно обробити зростає також потреба у сучасних та ефективних методах виділення характеристик, або ознак даних, які мають найбільший вплив на розглянуте питання.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ВІДБОРУ ОЗНАК ДЛЯ КЛАСИФІКАЦІЇ ТА КЛАСТЕРИЗАЦІЇ ДАНИХ

1.1 Аналіз предметної області, яка визначає специфіку дослідження методів відбору ознак для класифікації та кластеризації даних

З огляду на те, що в умовах сучасності дані надходять не тільки із мережі Інтернет, а й з різноманітних пристроїв, сенсорів, сервісів та інших джерел, тож отримані дані мають не тільки високу вимірність та характеризуються різноманітними типами даних, що відповідають вмісту ознак, а також мають високу ймовірність того, що не всі ознаки є інформативними та релевантними для розв’язання поставлених задач. Відповідно, перелічені фактори, пов’язані із сучасними даними значною мірою ускладнюють їх обробку.

Так, для того щоб обробити велику кількість різновимірних даних, необхідно розробляти та використовувати новітні ефективні методи виділення ознак даних, які дозволяють отримати найбільшу точність та продуктивність роботи методів машинного навчання для обробки даних, що відповідає поставленій задачі дослідження обраних даних. Під час такої обробки переліку ознак також необхідно враховувати також такі фактори як структурованість даних кожної окремо взятої ознаки, формату збереження даних ознаки з позиції того, яким чином необхідно перетворити зазначені дані для отримання необхідного результату, відповідного поставленій задачі дослідження даних, а також слід враховувати достовірність джерел інформації, звідки було отримано наведені дані.

Оскільки у сучасному світі обсяги даних невпинно зростають, і їхня структура стає дедалі складнішою, що створює додаткові труднощі під час їх обробки та аналізу. Дана проблема особливо актуальна для сфери машинного навчання, оскільки велика кількість ознак може ускладнювати процес навчання

моделей, подовжувати час навчання та впливати на ефективність роботи системи. Також під ударом стають результати роботи моделі, оскільки надмірна кількість ознак негативно впливає на легкість інтерпретації результатів моделі. Важко інтерпретовані результаті складно правильно зрозуміти. Ще навіть більшу складність має створення правильних та коректних висновків за отриманими даними. У зв'язку із зазначеними проблемами чітко постає потреба у створенні або виділенні методів відбору ознак, які були б здатні зменшити розмірність даних, без втрати їхньої інформативності у рамках розглянутого питання, поставленого під час аналізу наведеного набору даних.

Приймаючи до уваги усе перелічене вище, можна зробити висновок, що в сучасних умовах великої розмірності та різноманітності даних використання методів для ефективного відбору значущих та релевантних ознак з огляду на поставлену задачу дослідження даних стає критичним для досягнення найбільш точних та релевантних результатів, які можуть бути без значних ускладнень інтерпретовані людиною.

Загалом, розв'язання задачі відбору ознак для класифікації та кластеризації є важливим аспектом машинного навчання, через те, що саме цей процес дозволяє знаходити ключові тенденції та закономірності у виділених даних. Проведення досліджень у сфері алгоритмів та способів відбору ознак для класифікації та кластеризації охоплює виконання аналізу алгоритмів і методів, які дають змогу виділити лише ті ознаки, які мають найбільшу інформативність для розглянутого питання навіть із надзвичайно великих наборів даних. Відповідно до того, що у ході роботи даних алгоритмів або методів, кількість та якість відібраних ознак оптимізується, це, у свою чергу, напряму впливає на точність проведеної класифікації або кластеризації виділених даних. Також оптимізація кількості ознак для класифікації або кластеризації позитивно впливає на продуктивність роботи моделей та задіяні ресурси, потрібні для побудови та роботи моделей машинного навчання.

Загалом, робота методів класифікації та кластеризації даних базуються на використанні вхідного набору ознак, що описують досліджувані об’єкти або характеристики процесів або явищ у рамках заданого питання для аналізу даних.

Наступним кроком дамо визначення поняттям класифікації та кластеризації даних.

Класифікацією даних називають таку задачу, під час розв’язання якої із використанням моделі машинного навчання визначається приналежність класифікованого об’єкта до тієї або іншої категорії на основі комплексного аналізу моделлю характеристик заданого об’єкта [1]. Для виконання класифікації на основі вхідних даних необхідно, щоб дані містили у собі перелік об’єктів та визначені категорії даних об’єктів. На основі відповідності «об’єкт-категорія» будується модель класифікації даних, що містить у собі ряд правил та закономірностей, відповідно характеристик об’єктів, щодо поділу об’єктів заданого набору даних на категорії.

Кластеризацією даних, у свою чергу, є виконання поділу заданої множини вхідних об’єктів на категорії. Поділ виконується на основі аналізу даних об’єктів, але у якості вхідних даних для роботи, модель машинного навчання не отримує даних ані про кількість категорій об’єктів, наявних на даному наборі даних, ані про властивості, характерні для кожної із визначених груп об’єктів [1]. Таким чином, поділ об’єктів на групи, або кластери, виконується на основі спільності виділених значущих характеристик даних у рамках розглянутого питання.

Окремі нюанси роботи та обробки даних має робота із великими наборами даних. Великі та надзвичайно великі набори даних можуть містити тисячі ознак та мільйони строк. Виходячи із даних показників, без застосування спеціальних інструментів для виділення статистично значущих і незалежних ознак, аналіз таких даних постає фактично неможливою задачею. Більш того, обробка великих даних у більшості випадків може бути ускладнена через надмірність, особливо характерну для багатьох бізнес-даних, оскільки бізнес часто фіксує усі дані своєї

роботи, незалежно від того, чи є наявні дані потрібними, чи ні. Такі дані називають шумами даних та представляють собою деяку несуттєву для обраного розглянутого питання аналізу даних інформацію, яка не несе змісту для розглянутого питання і є радше випадковою [2].

Для того, щоб обробити більшість великих наборів даних, необхідна наявність значні ресурсів, потрібна як із огляду на велику кількість та об’ємність даних, так і з огляду на ресурсоємність роботи більшості алгоритмів виділення значущих ознак [2]. Також, завжди потрібно приймати до уваги той факт, що значна частина ознак, як правило, не є інформативною для вирішення конкретної поставленої задачі аналізу даних. Крім цього, завжди потрібно пам’ятати й про те, що взаємозалежність ознак може негативно впливати на точність результатів моделі [3].

Загалом, процес підбору мінімального набору ознак для виконання задачі класифікації або кластеризації даних представляє собою пошук такої підмножини ознак, яка найкраще представляє задані, або вхідні, дані для досягнення розв’язку заданого питання дослідження [4]. Наприклад, у випадку виконання класифікації даних, розв’язком питання дослідження може постати розроблена модель для передбачення поведінки певної ознаки або групи ознак із найвищим ступенем точності отриманих передбачень. Іншим прикладом може бути розроблена модель для кластеризації даних із найвищою точністю визначення груп даних.

Для того, щоб провести вибір ознак даних для виконання класифікації або кластеризації найкращим чином, необхідно враховувати ряд факторів, які безпосередньо впливають на якість отриманих результатів у контексті поставленої задачі.

Першим фактором, який суттєво впливає на отримані результати роботи методів класифікації та кластеризації є поняття інформативності ознак. Дане поняття відіграє значну роль у побудові моделей машинного навчання, які

можуть надавати ефективні результати роботи. Інформативність кожної окремо взятої ознаки може бути різною з огляду на розглянуте питання дослідження та пов’язані із ним фактори [5]. Саме по собі, поняття інформативності ознак даних відповідає мірі того, наскільки окремо взята ознака даних несе інформативну цінність у рамках розглянутого питання і її ступінь впливу на кінцевий результат роботи моделі [5].

У тому випадку, якщо обрана у якості інформативної, ознака не несе або майже не несе інформації для питання, поставленого до даних, або у тому випадку, коли вплив на вихідний результат роботи моделі є мінімальним, така ознака вважається неінформативною та може бути виключена із процесу підбору ознак без втрати якості вихідного результату роботи моделі [5].

Сам по собі, процес оцінки інформативності ознак даних спрямований на те, щоб виявити найбільш значущі ознаки на обраному наборі даних, які відповідають розв’язанню поставленого питання та безпосередньо позитивно впливають на процес класифікації або кластеризації. Вимір ступеня інформативності ознак для поставленої задачі дозволяє сформулювати перелік лише тих ознак, які спрямовані на отримання найкращих результатів роботи моделей класифікації та кластеризації завдяки точним прогнозам та найбільшій швидкодії алгоритмів [5].

Другим фактором, який суттєво впливає на отримані результати роботи методів класифікації та кластеризації є поняття вимірності виділених ознак. Під поняттям вимірності ознак мається на увазі кількість ознак, обрана для виконання задач класифікації або кластеризації [5].

Ключовим завданням даного фактору є максимальне зменшення кількості ознак, обраних для аналізу вхідних даних, з огляду на поставлену задачу [5]. Варто також зауважити, що під час зменшення вимірності виділених ознак здатність до створення точного прогнозу на основі вхідних даних повинна залишатись високою.

Надмірна кількість ознак для виконання задач класифікації та кластеризації значно ускладнює роботу моделі, що, у свою чергу, впливає на різке підвищення ресурсоємності роботи моделі. Якщо розглядати ситуацію із надмірністю даних, тоді також варто враховувати той факт, що у такому випадку також можливе перенавчання моделі, що впливає на точність та релевантність отриманих результатів роботи моделі [5].

Постановку завдання виділення ознак, враховуючи поняття розмірності ознак, можна сформулювати як виділення найменшої задовільної множини ознак, яка буде давати найвищі результати точності роботи моделі. За умови створення такого балансу ознак даних, час навчання моделі для класифікації або кластеризації на вхідних даних може бути значно скорочено, а результати моделі будуть більш чіткими та зрозумілими для їх інтерпретування.

Третім фактором, який суттєво впливає на отримані результати роботи методів класифікації та кластеризації є поняття мультиколінеарності ознак вхідних даних. Поняття мультиколініарності даних визначає ступінь взаємного зв’язку між ознаками даних, які є лінійно незалежними, але деякою мірою повторюють одна одну [5]. Прикладами мультиколінераних ознак можуть бути ознаки, які мають різні одиниці вимірювання одного й того самого явища, наприклад, довжина пелюстки у міліметрах та дюймах, записані у різні ознаки, або колонки набору даних.

Наявність таких ознак вносить надмірність у даних і викривляє ваги у моделі, що призводить до менш точних прогнозів. Це відбувається через те, що модель може давати надмірну оцінку таких пов’язаних ознак через те, що вплив таких ознак є повторюваним [5]. Таким чином, вплив інших ознак на результати роботи моделі може бути розціненим моделлю як незначний.

Для підвищення показників точності та стабільності роботи моделі для класифікації або кластеризації даних необхідно уснунути або зменшити мультиколінеарність та взаємний зв’язок між виділеними ознаками даних.

Через те, що більшість задач побудована на тому, щоб обробити максимальну кількість даних за мінімальний проміжок часу. Але на шляху до цього постає велика кількість проблем через зростання кількості проблем через зростання обсягів даних для обробки при сталих ресурсах обробки.

При зростанні кількості даних відповідно зростає кількість ознак даних. Це спричиняє у свою чергу ряд проблем.

По-перше, ефективність роботи методів класифікації або кластеризації значно знижується в умовах надзвичайно великої кількості ознак. Результати аналізу даних за допомогою класифікації або кластеризації перестають бути точними та легко інтерпретованими для людини [6].

По-друге, коли на обробку даних за допомогою методів класифікації та кластеризації поступає великий набір даних із величезною кількістю ознак різної змістовності, модель може давати хибні результати, оскільки починає сприймати локальні екстремуми незначних за важливістю такими ж важливими, як і основні тенденції даних [6]. Це призводить до результатів із низькою точністю та нестабільною в умовах навіть незначної зміни вхідних даних, оскільки виділені локальні особливості одних даних не є релевантними для інших. Суміжною із наведеною проблемою є проблема перенавчання моделі. Це характеризується тим, що модель точно запам’ятовує тренувальні дані і не здатна зовсім до виділення нових основних особливостей інших даних, навіть навчальних [7]. Така модель не може бути адаптована до використання на інших даних та не буде ефективною у даних умовах.

По-третє, велика кількість даних та ознак даних, які необхідно обробити алгоритмам машинного навчання спричиняють значне використання ресурсів та потребу в додаткових обчислювальних ресурсах системи [6]. Оскільки кількість даних невпинно збільшується, це стає реальною проблемою.

Четвертою проблемою є нерівномірність розподілу ознак у наборі даних. Дана ситуація створюється тоді, коли ознаки, які необхідні для розв’язання

поставленого питання дослідження недостатньою мірою представлені у наборі даних, або навпаки, є представлені, але у меншому співвідношенні, ніж ознаки, які не мають цінності у рамках розглянутого питання дослідження даних [6]. Таким чином, через дану проблему із вхідними даними для задачі класифікації або кластеризації, отримані результати роботи методів можуть мати зміщення з огляду на нерівномірність представленості даних, а отже, бути неточними.

П’ятою проблемою є збільшення кількості взаємопов’язаних ознак зі збільшенням кількості ознак даних в цілому. Ознаки можуть бути взаємопов’язані різним чином. Вони можуть бути взаємнозалежними, тобто корелювати між собою, або бути мультиколінеарними, тобто повторюючими відносно інших ознак [6]. Чим більшою є кількість ознак у наборі даних, тим важче виявити такі ознаки та усунути їх для досягнення найкращого результату.

1.2 Дослідження існуючих методів відбору ознак для роботи алгоритмів кластеризації та кластеризації даних

Для розв’язання задачі підбору релевантних ознак у наборі даних для подальшої класифікації або кластеризації наразі існує безліч методів. Деякі із даних методів вже вважається класичними, інші, натомість, не отримали значного розповсюдження.

Одним із наразі існуючих сучасних трендів галузі машинного навчання є впровадження глибинного навчання та нейронних мереж для виділення релевантних ознак для розв’язання поставленої задачі. Такі методи, наприклад, використовують засоби для автоматичного кодування ознак. Згідно із принципом роботи таких методів, перелік ознак, який міститься у даних автоматично кодується, що слугує для зменшення розмірності даних шляхом навчання алгоритму за допомогою переліку ознак, наявних у даних, але у компактному

представленні [8]. Для кодування ознак даних такі методи виконують нелінійне перетворення вхідних ознак даних, зберігаючи важливу інформацію для подальших завдань класифікації або кластеризації. Даний метод не є засобом для виділення лише значущих ознак, але може бути використаним на фінальному етапі для прискорення навчання алгоритму шляхом передачі на автокодування лише тих ознак, які є релевантними та важливими для поставленої задачі дослідження даних.

Наразі також є популярним використання генеративно-змагальних мереж. Даний тип нейронних мереж у переважній більшості випадків створюється для генерації даних, зображень, тощо. Але також даний тип мереж можна застосувати для виділення значущих ознак для класифікації та кластеризації даних.

Генеративні змагальні мережі складаються із генератора і дискримінатора, які працюють наступним чином: за допомогою генератора відбувається створення даних, а за допомогою дискримінатора відбувається оцінка створених даних із перспективи того, наскільки створені дані близькі до реальних [9]. Таке порівняння згенерованих результатів із реальними даними відбувається до тих пір, поки генератор не стане генерувати дані найбільш близькі до реальних вхідних даних, на яких було навчено мережу [9].

З перспективи виділення важливих та значущих ознак у рамках розглянутого питання дослідження даних, даний тип нейронних мереж використовується таким чином: за допомогою генератора створюються маски або підмножини ознак, які дають найкращий результат у рамках поставленого завдання дослідження даних, а дискримінатор використано для того, щоб провести оцінку ефективності обраних ознак та передати генератору інформацію про те, які саме ознаки впливають на точність отриманих прогнозів у ході ітерацій навчання мережі [9].

Використання даного типу нейронних мереж дає ряд переваг, як-от вбудований захист перенавчання за рахунок того, що за своїм принципом дії

алгоритм поступово скорочує використану кількість ознак для навчання, зниження складності моделі та створення узагальненої моделі на основі вхідних даних.

Але ключовим недоліком використання генеративних змагальних мереж для виконання задачі виділення ознак для класифікації та кластеризації є те, що даний тип нейронних мереж відповідно до правил свого функціонування, складно навчається, потребує багато ресурсів апаратної частини для проведення навчання, а навчена модель є доволі вузько спрямованою, тобто працює лише на певному вузькому наборі даних та складно вдосконалюється, оскільки генератор і дискримінатор не вдосконалюються синхронно під час роботи алгоритму [9]. Через названі недоліки результати роботи генеративних змагальних мереж неможливо використовувати на великих наборах даних та ознак, а перелік обраних ознак, ідентифікованих алгоритмом, як важливих, необхідно перевіряти за допомогою додаткових метрик або інших моделей [10].

Також певного розповсюдження наразі набувають еволюційні алгоритми. Ключовою відмінністю еволюційних алгоритмів є використання природніх принципів відбору та еволюції у процесі навчання нейронної мережі [11]. У цьому й полягає ключова ідея еволюційних алгоритмів, а саме в створенні імітації природніх процесів мутації, відбору та кросоверу для поліпшення якості обраної множини ознак, або у термінології даного принципу, популяції [11].

Якщо розглянути поставлену задачу підбору значущих ознак із множини усіх вхідних ознак із огляду на розглянуту задачу, тож можна стверджувати, що у контексті еволюційних алгоритмів кожна окремо взята підмножина випадковим чином підібраних ознак – це особина популяції [11]. Таким чином, алгоритм створює популяцію із множин ознак, які було підібрано випадковим чином. Наступним кроком роботи алгоритму є оцінка ефективності розглянутої популяції підмножин за допомогою цільової функції, яка дозволяє оцінити якість отриманого результату [11]. Таким чином, порівнюючи результати, які

демонструє кожна із підмножин, підмножини із найвищими результатами мають найбільшу кількість шансів того, що вони залишаться у популяції та їхні характеристики буде передано наступним поколінням підмножин [11].

Використання еволюційних алгоритмів для виділення найбільш релевантних та значущих ознак для розв’язання поставленої задачі дослідження має ряд переваг, наприклад ефективна обробка великих масивів даних та ознак, урахування складних залежностей між даних, та можливість виділення глобальних тенденцій даних та прибрання із опрацювання локальних екстремумів даних. Алгоритми даного типу дозволяють уникнути перенавчання та видалити нерелевантні для розглянутого питання та корельовані ознаки для підвищення якості отриманого результату.

Однак, використання еволюційних алгоритмів для виділення найбільш релевантних та значущих ознак для розв’язання поставленої задачі дослідження має також ряд недоліків. По-перше, за своїм принципом еволюційні алгоритми є незвичними та складними у налаштуванні через те, що параметрів алгоритму, таких як ймовірності мутації та кросоверу вимагає багато часу на уваги для отримання найкращих результатів. По-друге, через свою складну структуру такі моделі довго навчаються та для проведення навчання необхідні значні ресурси. По-третє, через принцип сам функціонування еволюційних алгоритмів, результат їх роботи дуже сильно залежить від початкових умов та обраної першої популяції, через це необхідно зробити деяку кількість запусків для отримання стабільного результату роботи моделі. Еволюційні алгоритми найкращим чином демонструють результати саме для великих наборів даних, але зважаючи на зазначені недоліки використання їх не виглядає добрим вибором.

Перелічені вище методи можна умовно віднести до некласичних підходів до розв’язання задачі виділення ознак для задач кластеризації або класифікації оскільки вони мають досить вузьку ситуацію застосування та ряд потенційних вузьких місць, якщо один або декілька факторів не будуть відповідати ситуації.

До класичних методів до відбору ознак для розв’язання задач класифікації або кластеризації відносять основні три типи підходів: обгорткові, фільтрові та вбудовані методи [12]. Дані підходи та найбільш розповсюджені їх методи буде розглянуто у даній роботі. Кожен із зазначених класичних підходів до виділення значущих ознак даних має свої переваги та недоліки, а вибір підходу залежить від специфіки даних і поставленої задачі дослідження.

Фільтрові методи відбору значущих ознак для розв’язання поставленої задачі вважаються найбільш простими серед методів відбору ознак. Принцип дії даних методів відбору ознак базується на тому, що значущі ознаки обираються відповідно значень статистичних показників та математичних формул [13]. Перевагою використання даних методів є їх незалежність від обраного алгоритму машинного навчання для подальшої обробки.

Принцип дії фільтрових методів відбору значущих ознак для виконання задачі класифікації або кластеризації з огляду на поставлене питання дослідження є наступним. Для функціонування фільтраційних алгоритмів необхідно обрати цільові змінні або ознаки, далі, на основі кореляції інших ознак набору даних із обраними цільовими ознаками. Також релевантні ознаки можуть обиратися на основі значень властивостей ознак даних відносно даних набору. Робота даного фільтраційних методів заснована на статистичних тестах, таких як t-тести або коефіцієнт кореляції Пірсона [12].

Перевагами використання фільтрових методів є те, що вони мають високу швидкодію під час навчання та тренування, їх налаштування перед використанням не є складним. Загалом, фільтрові методи відбору ознак використовуються у тих випадках, коли необхідно зробити швидку попередню обробку даних, тобто виділити значущі ознаки для питання дослідження.

Однак, використання фільтрових методів для виділення ознак для розв’язання задач класифікації та кластеризації також має свої недоліки. Фільтрові методи не враховують кореляцію між даними, а також специфіку

обраного алгоритму [13]. Через це ознаки, насправді важливі для виконання поставленої задачі дослідження можуть бути випадково усунуті.

Обгорткові методи відбору ознак для задач класифікації та кластеризації, мають відмінний від фільтрових методів принцип функціонування. Першою відмінністю є те, що оцінка якості результату залежить від обраного алгоритму машинного навчання, а тому можуть визначити із більшою точністю, які саме ознаки є корисними для використання для конкретного поставленого питання.

Функціонування обгорткових методів підбору ознак проходить наступним чином: модель проходить навчання на різних підмножинах ознак вхідних даних, після цього проходить етап оцінки продуктивності роботи моделі на основі обраної метрики, наприклад, точності [14]. Таким чином, найкращим є набір ознак, який дає найкращі результати за обраною метрикою.

Незважаючи на перелічені переваги використання обгорткових методів виділення ознак для задач класифікації та кластеризації, порівняно, наприклад, із фільтровими методами, варто зазначити й потенційні недоліки їх використання. З огляду на той факт, що модель навчається багато разів на різних наборах ознак, необхідні системні потужності для використання алгоритму зростають відповідно до кількості ознак, що може складати проблему при обробці великих наборів даних.

Вбудовані методи відбору ознак для задач класифікації та кластеризації є своєрідним поєднанням характеристик та методів фільтрових та обгорткових методів відбору ознак [15]. Принцип роботи вбудованих алгоритмів виділення значущих ознак у тому, щоб провести відбір ознак у ході навчання самої моделі. Завдяки цьому, одночасно із навчанням, набір ознак постійно оптимізується і відповідно до цього проводиться налаштування моделі. Це забезпечує приріст продуктивності роботи алгоритмів порівняно із обгортковими методами, оскільки зниження розмірності даних проводиться одночасно із роботою алгоритму і не виноситься в окремий етап для відбору значущих ознак [16].

Основна ідея полягає в тому, що відбір ознак відбувається під час навчання моделі, що дозволяє одночасно оптимізувати набір ознак і налаштовувати модель. Це забезпечує зниження розмірності даних і покращення продуктивності без необхідності проводити окремі етапи відбору ознак і навчання.

Попри свої значні переваги, вбудовані методи відбору значущих ознак для класифікації та кластеризації також мають і ряд недоліків. По-перше, результати роботи даних методів досить сильною мірою залежать від обраного алгоритму. Так, отриманий результат роботи моделі може бути обраним найкращим лише для поточної моделі, натомість для іншої даний результат не буде повною мірою релевантним. По-друге, через структурний принцип роботи алгоритму, алгоритми даного типу можуть перенавчатись, тобто потрібно ретельно обирати такий набір даних для того, щоб мінімізувати такий ризик. Такий набір даних повинен бути достатньо великим. По-третє, за рахунок виконання декількох операцій в рамках одного процесу обчислювальна вартість вбудованих алгоритмів виділення ознак є достатньо високою, тож це впливає на швидкість їх роботи та потребу в системних ресурсах для виконання обчислень.

Новими підходами до поставленої задачі відбору ознак часто називають такі методи, які відходять від традиційних методик та пропонують рішення, засновані на методах машинного навчання, а саме методах випадкового лісу, для автоматичного відбору ознак за принципами їх важливості на основі ступеня впливу кожної окремої ознаки на вихідний результат. Використання таких підходів дозволяє отримати інтерпретовані результати роботи, а також можуть бути реалізовані для більшої наочності для того, щоб зрозуміти, які ознаки мають найбільшу вагу у процесі вибору значущих ознак.

Для методів відбору ознак важливим фактором є здатність методів, використаних для відбору ознак, обробляти різноманітні проблеми даних, наприклад взаємну кореляцію ознак. Через наявність сильно корельованих ознак можуть виникати проблеми, наприклад перенавчання моделей, через що моделі

перестають ефективно працювати на різних даних. Для того, щоб усунути проблеми із перенавчанням необхідно використовувати методи, які тим чи іншим чином проводять регуляризацію даних, тим самим зменшуючи вплив корельованих ознак на процес відбору.

Першим кроком формування вимог до результатів відбору ознак для класифікації або кластеризації є формування переліку вхідних даних задачі та виділення ознак даних, які підлягають дослідженню. Наступним кроком з огляду на вхідні дані, необхідно зробити висновки про те, які методи виділення ознак буде використано.

Другим кроком виконання задачі відбору ознак необхідно провести навчання моделей для виконання класифікації або кластеризації залежно від поставленої задачі. Після того, як навчання моделей буде проведено, потрібно використати метрики для оцінки ефективності отриманого результату та провести повторне навчання за потреби.

З метою проведення порівняння результатів випробуваних методів виділення значущих ознак необхідно проаналізувати продуктивність роботи обраних та розроблених моделей за допомогою відповідних метрик та статистичних тестів, які також дозволяють визначити статистичну значущість різниці точності отриманих результатів.

1.3 Формування актуальності та мети проведення дослідження методів відбору ознак для роботи алгоритмів кластеризації та кластеризації даних

Актуальність вивчення методів вибору ознак для класифікації та кластеризації даних зумовлена стрімким зростанням обсягів інформації у різних галузях, наприклад, у сфері бізнесу, медицини, тощо. Це явище є наслідком швидкого розвитку цифрових технологій, поширення Інтернету речей і

формування великих даних унаслідок функціонування різних сфер сучасного життя. У сучасному світі інформація стає дедалі масштабнішою та складнішою, що створює значні виклики для її аналізу та обробки. Особливим чином це стосується сфери машинного навчання, де велика кількість ознак може ускладнювати процес побудови моделей, подовжувати час їх навчання і негативно впливати на загальну ефективність та результати навчання. Через дані фактори виникає гостра потреба у створенні дієвих методів відбору значущих ознак для поставленої задачі кластеризації або класифікації, які допоможуть зменшити розмірність даних, водночас зберігаючи їхню інформативну цінність для поставленої задачі.

Розглянуті вище методи відбору значущих ознак, мають важливе значення для покращення ефективності алгоритмів класифікації та кластеризації. Правильний вибір методу відбору ознак здатний значною мірою впливати на точність обраних моделей для класифікації та кластеризації, сприяючи у свою чергу на зменшення ризику перенавчання моделі, та збільшення ефективності моделей для задач узагальнення отриманих знань та підвищенню інтерпретованості отриманих результатів класифікації та кластеризації. Для формування розуміння особливостей використання різноманітних підходів до виконання задачі підбору ознак, наприклад, фільтрових, обгорткових та вбудованих методів, необхідно сформувати уявлення про принципи їх функціонування та особливості ситуацій використання кожного із зазначених підходів.

Незважаючи на ряд переваг, які пропонує використання різноманітних методів відбору ознак для подальшого використання методів класифікації та кластеризації, усі вони також мають й спільні недоліки. Найголовнішим недоліком є те, що значущі ознаки для контексту поставленої задачі може бути відкинуто, а неважливі ознаки, натомість, можуть бути взяті до фокусу уваги за різних причин. Це може бути спричинено наявністю сильно корельованих та

повторюючих ознак. Тому важливо підібрати такі методи підбору ознак, які й на великих наборах даних зможуть оброблювати взаємопов’язані ознаки.

Для оцінки результатів проведених досліджень необхідно обрати перелік метрик, які будуть інформативними для методів класифікації та кластеризації та дозволять найкращим чином оцінити результати роботи методів машинного навчання. Остаточним результатом такої оцінки є сформований перелік методів, які дають найкращі результати та є найбільш ефективними для розв’язання задач кластеризації та класифікації. Вивчення ситуацій застосування кожного окремо взятого методу дозволить приймати обґрунтовані рішення вибору алгоритмів для виконання виділення значущих ознак у контексті різних поставлених задач для різних даних.

Метою дослідження методів відбору ознак для задач кластеризації та класифікації є детальний аналіз випадків та особливостей застування різних підходів до виділення ознак та формування переліку закономірностей їх використання. У рамках даного дослідження планується проведення системного аналізу актуальних практик та підходів до підбору значущих ознак, які будуть мати найбільший вплив на результати роботи методів класифікації та кластеризації з огляду на поставлену задачу та контекст її використання.

У ході проведення даного дослідження планується планується розробка алгоритму проведення експериментів, які дозволять зробити найбільш інформативні висновки стосовно роботи методів виділення ознак, а також результатів кластеризаціі та класифікації на виділеному наборі ознак. Окрім цього, на основі проведених експериментів планується формування детальних висновків визначення ситуацій застосування кожного із методів виділення значущих ознак для розв’язання задачі з огляду на дані та особливості предметної області.

Результати розробленого дослідження буде корисним для формування розуміння принципів роботи виділення цільових та невідповідних ознак для

розв’язання задачі дослідження, а також формування підходів до автоматизації даного процесу.

Розробка даного дослідження також буде корисною для вивчення особливостей роботи методів виділення ознак для науковців та фахівців даної галузі, оскільки окрім детального опису ситуацій застосування методів, таке дослідження може відкрити потенційні недоліки методів, які можуть бути усунуті у майбутньому. Зокрема, це може сприяти розвитку розробки нових, більш універсальних підходів до виділення значущих ознак для розв’язання задачі виділення ознак для роботи методів машинного навчання, зокрема алгоритмів класифікації та кластеризації.

1.4 Постановка задачі дослідження методів відбору ознак для роботи алгоритмів кластеризації та кластеризації даних.

Формування постановки задачі є важливим для успішного проведення й завершення будь-якого дослідження. Не винятком є й дослідження актуальних методів для виділення значущих ознак для задач класифікації та кластеризації. Для задач кластеризації та класифікації підбір релевантних ознак є одним із найважливіших етапів, оскільки саме правильно підібраний перелік ознак визначає більшу частину ефективності роботи алгоритмів машинного навчання для конкретної поставленої задачі дослідження даних.

Об’єктом дослідження даної роботи з аналізу методів відбору ознак для задач класифікації та кластеризації є великі набори даних, які містять велику кількість ознак, однорідних або різнорідних, які необхідно відібрати для розв’язання задач кластеризації або класифікації.

Метою даного дослідження з аналізу методів відбору ознак для задач класифікації та кластеризації є проведення розробки та дослідження актуальних

методів підбору ознак з метою підвищення точності та ефективності роботи методів класифікації та кластеризації для розв’язання поставлених задач.

Для того, щоб досягти визначеної мети дослідження методів відбору ознак для задач класифікації та кластеризації, необхідно виконати ряд завдань, а саме:

– проаналізувати існуючі актуальні методи відбору ознак для задач класифікації та кластеризації, з’ясувати їх ефективність у різних ситуаціях застосування для задач класифікації та кластеризації даних;

– виділити ряд методів для визначення ознак для задач класифікації та кластеризації, які підлягають порівняльному аналізу у рамках фільтрового, обгорткового та вбудованого підходів до відбору ознак;

– з’ясувати вплив обраного методу для відбору ознак для задач класифікації та кластеризації на точність та продуктивність алгоритмів класифікації та кластеризації, створивши програмну реалізацію виділених методів фільтрових, обгорткових та вбудованих підходів до відбору ознак;

– створити порівняльний аналіз результатів класифікації та кластеризації відповідно до метрик, отриманих у наслідок застосування кожного із методів відбору ознак;

– сформувати перелік висновків та рекомендацій щодо вибору оптимальних методів відбору ознак для різних типів задач класифікації та кластеризації даних.

2 МАТЕМАТИЧНІ МОДЕЛІ ВІДБОРУ ОЗНАК ДЛЯ КЛАСИФІКАЦІЇ ТА КЛАСТЕРИЗАЦІЇ

У ході проведення аналізу будь-яких даних, необхідно працювати із групами. Наприклад, під час аналізу ринкових ніш або груп клієнтів, компанії об’єднують своїх клієнтів у групи для розуміння поточної ситуації використання продукту; компанії об’єднують у групи співробітників, які демонструють різні рівні продуктивності під час роботи для виділення премій найактивнішим; або ж потенційний пацієнт під час вибору медичного закладу умовно розподіляє їх на групи за їх рейтингом, для того, щоб обрати найкращий.

Для того, щоб сформувати групи за наявними даними використовується метод кластеризації даних [12]. Для виконання задачі кластеризації та класифікації, формування даних у групи, використовуються характеристики, відповідно до яких дані групи може бути виділено, незалежно від суті об'єктів, які необхідно кластеризувати, або сфери застосування.

Однак, варто враховувати той факт, що ознаки, які містить набір даних певної області, не завжди є однаково важливими та інформативними для виділеннями за ними груп об'єктів. Наприклад, для вже розглянутого прикладу із аналізом клієнтів компанії, яка, наприклад, поставляє програмний застосунок, більш важливими будуть дані саме про те, який відсоток часу користувачі використовують застосунок, або який відсоток користувачів використовує його протягом першого місяця або тижня. У такому контексті дані про те, чи має користувач дітей, не є важливим. Або ж для прикладу аналізу продуктивності співробітників компанії важливими будуть дані про кількість задач співробітника, або відсоток завершених задач протягом виділеного проміжку часу, натомість дані про використані соціальні привілеї ніяк не можна назвати значущими у контексті розглянутого питання.

Під час роботи із даними загалом, не залежно від того, чи будуть дані використовуватись для розв’язання задач кластеризації або класифікації, чи ні, типи даних які можуть мати ознаки, наявні у розглянутому наборі даних, можуть кардинально відрізнятися за вмістом та стандартами збереження даних відповідно до фізичного змісту кожної окремо взятої ознаки. Ознаки, наявні у певному наборі даних, можуть мати різний ступінь інформативності для характеристики того чи іншого явища або об’єкта. Якщо брати до уваги дані бізнесу або дані служб телеметрії або аналітики, то кількість ознак може бути надзвичайно великою, а тому визначення переліку саме значущих для задачі ознак може викликати складнощі.

Негативний вплив незначних характеристик на результати точності роботи алгоритмів кластеризації та класифікації є вже науково вивченим та обґрунтованим [17]. Згідно проведеним експериментам, зокрема, використовуючи добре відомий набір даних Iris, де видно, що класифікація видів рослин легше здійснюється за допомогою певних характеристик, таких як довжина пелюстки та довжина чашолистка [17]. Якщо розглянути інший набір експериментальний набір даних, що містить чотири кластери і п'ять характеристик, лише дві з них є важливими [17]. Це підтверджує той факт, що присутність незначних характеристик може ускладнити класифікацію [17].

Для того, щоб усунути проблему із включенням незначних ознак даних до алгоритмів класифікації, необхідно використовувати такі засоби обчислення, які дозволять обчислювати значні ознаки та включати їх до набору даних. Із метою виконання даної задачі використовуються ряд алгоритмів, які дозволяють обирати лише важливі ознаки для виконання подальшої задачі класифікації даних.

Якщо розглянути питання різниці у відборі ознак для підготовки для кластеризації та класифікації, то є як спільні риси, так і відмінності.

По-перше, для задачі класифікації вхідні дані повинні містити мітки, натомість для кластеризації мітки не є потрібними.

По-друге, іншим є підхід для відбору ознак. У випадку задачі класифікації, важливим є вихідний результат моделі, тобто точність отриманих передбачень на основі вхідних даних. Натомість для задачі кластеризації найбільш важливо зберегти саме природні для розглянутого набору даних групування даних.

По-третє, методи класифікації та кластеризації можуть використовувати різні метрики для оцінки якості роботи моделей.

2.1 Фільтрові методи підбору ознак для задач кластеризації та класифікації

Першою групою методів виділення ознак для задач класифікації та кластеризації було прийнято рішення розглянути фільтрові методи. Фільтрові методи підбору ознак ґрунтуються на тому, що релевантність кожної окремо взятої ознаки ґрунтується на аналізі власних характеристик ознак та не залучають додатково методи машинного навчання для оцінки релевантності.

Фільтраційні методи відбору ознак вважаються швидкими методами, які потребують менших обчислювальних ресурсів для роботи та використовуються для попередньої обробки великих наборів даних. Такі методи не вміють обробляти корельовані ознаки, тому за наявності кореляції між ознаками, фільтрові методи можуть давати велику похибку [19].

Загалом фільтрові методи не мають схильності до перенавчання та працюють швидко, особливо порівняно із обгортковими та вбудованими методами відбору ознак.

Принцип дії фільтрових методів відбору ознак, які можуть бути використані для розв’язання задач кластеризації та класифікації є наступним. Першим кроком для фільтрових методів відбору ознак є вибір набору ознак.

Наступним кроком для ознак обчислюються показники, які відображають її важливість щодо цільової змінної або даних у цілому. Ці метрики можуть оцінювати статистичні залежності (наприклад, кореляцію, ентропію, взаємну інформацію) або властивості даних (наприклад, дисперсію) [20]. Наступним кроком отриманий перелік ознак та їх отриманих показників порівнюється. Ознаки, значення метрик яких є меншим за заданий поріг впливу на цільову змінну, виключаються із переліку. Навчання моделі проводиться лише на наборі ознак із високим значенням показника впливу на цільову змінну.

Схему роботи фільтрових методів відбору ознак для алгоритмів машинного навчання, зокрема класифікації та кластеризації, наведено на рисунку 2.1.

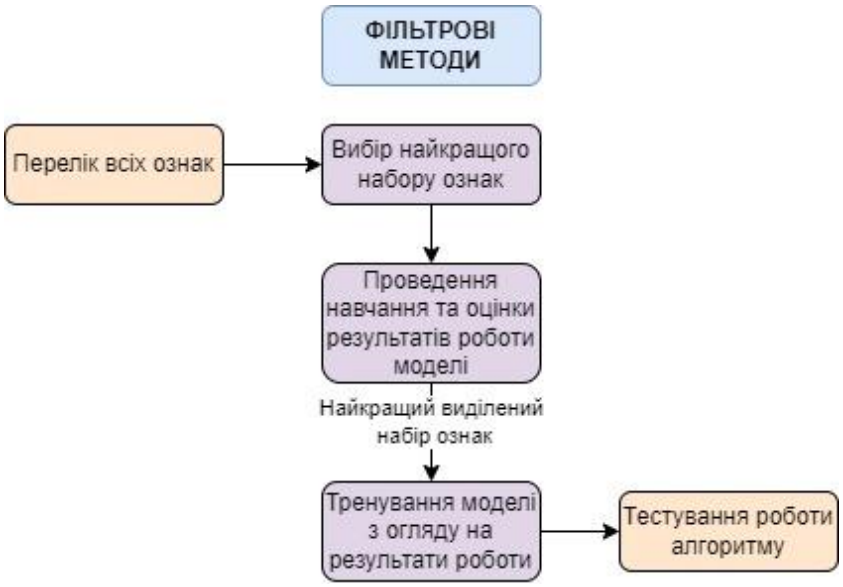


Рисунок 2.1 – Алгоритм роботи фільтрових методів відбору ознак

2.1.1 Метод відбору ознак за критерієм Фішера

Критерій Фішера, (англ. Fisher Score), перший розглянутий фільтровий метод відбору ознак, який ґрунтується на значенні статистичного критерію, який визначає можливість розрізняти класи за значенням ознаки. Згідно із методом,

показник обчислюється окремо для кожної ознаки на основі значення міжкласової до внутрішньокласової дисперсії [21]. На основі отриманого значення відношення показників, критерію Фішера, можна зробити висновки стосовно значущості ознаки для розглянутого питання класифікації або кластеризації. Таким чином, високі значення показника критерію Фішера говорять про те, що обрана ознака може бути використана для розділення класів.

Зважаючи на те, що за своїм принципом роботи критерій Фішера виконує оцінку кожної ознаки окремо від інших, даний метод не враховує ніякі взаємні зв'язки між ознаками, через що результати роботи методу можуть не бути повними.

Критерій Фішера часто використовується для попередньої обробки даних, зазвичай у завданнях із обробки тексту та для задач, де необхідно виконати розпізнавання зображень з метою зменшення розмірності вхідних даних для підвищення точності роботи моделі.

Загальну формулу критерію Фішера представлено у формулі (2.1).

$$F(i) = \frac{\sum_{c=1}^{C} N_c \, (\mu_{i,c} - \mu_i)^2}{\sum_{c=1}^{C} N_c \, \sigma_{i,c}^2}, \tag{2.1}$$

де $F(i)$ – значення критерію Фішера для ознаки i;

C – кількість класів;

N_c – кількість зразків у класі c;

$\mu_{i,c}$ – середнє значення ознаки i для класу c;

μ_i – середнє значення ознаки i для всіх зразків;

$\sigma^2_{i,c}$ – дисперсія ознаки i у класі c.

2.1.2 Метод відбору ознак за критерієм Хі-квадрат

Метод відбору ознак за критерієм Хі-квадрат, (англ. Chi-squared score), представляє собою статистичний метод, згідно із яким, кожна окрема ознака оцінюється відповідно до її зв’язку із цільовою змінною. Цільова змінна ж, у свою чергу може набувати категоріального значення.

Відповідно до принципу роботи, метод Хі-квадрат виконує обчислення статистики для кожної із ознак та визначає окремо значущість кожної ознаки на основі відхилення розподілу частот від очікуваних значень розподілу [22]. Високі значення даного критерію вказують на те, що зв’язок між розглянутими класами є сильним.

Даний метод застосовується зазвичай для задач класифікації, коли необхідно обробити дискретний набір ознак або бінарні ознаки. Для інших даних, дані необхідно попередньо обробити.

Як і усі фільтрові методи відбору ознак, метод відбору ознак за критерієм Хі-квадрат використовується у тих випадках, коли необхідно швидко обробити великі дані, наприклад текстові дані.

Загальну формулу критерію Хі-квадрат наведено у формулі (2.2).

$$x^2(i) = \sum_{k=1}^{K} \frac{(O_{ik} - E_{ik})^2}{E_{ik}}, \tag{2.2}$$

де $x^2(i)$ – значення критерію Хі-квадрат для ознаки i;

O_{ik} – спостережена кількість зразків у k-й категорії;

E_{ik} – очікувана кількість зразків у k-й категорії;

K – кількість категорій.

2.1.3 Метод відбору ознак на основі кореляції

Метод відбору ознак на основі кореляції, (англ. Correlation-based feature selection (CFS)), представляє собою метод відбору ознак, який використовує поняття кореляції для виконання вібору ознак.

Згідно зі своїм принципом роботи, метод відбору ознак на основі кореляції, виконує аналіз кореляції між обраним цільовим класом, та поточною ознакою. За даним методом, найбільш релевантними вважаються такі ознаки, які мають найсильнішу кореляцію із цільовим класом [23]. Натомість, найкращим випадком для методу є мінімальна або відсутня кореляція між самими ознаками з метою уникнення помилок у роботі.

Даний метод є універсальним методом відбору ознак із позиції вхідних даних. Так, даний метод може обробляти як дискретні, так і неперервні дані, та прагне усунення надлишковості в отриманих даних шляхом виділення лише найоптимальнішої підмножини релевантних ознак для розв'язання поставленої задачі.

З огляду на формулу функціонування даного методу відбору ознак, даний метод здатний виявляти лише лінійні залежності між ознаками даних.

Загальну формулу методу відбору ознак на основі кореляції наведено у формулі (2.3).

$$CFS = \frac{\sum_{i=1}^{m} Corr(f_i, y)}{\sqrt{m + m(m-1) \times Corr(f_i, f_j)}}, \tag{2.3}$$

де CFS – оцінка підмножини ознак за методом відбору на основі кореляції;

m – кількість ознак;

$Corr(f_i, y)$ – кореляція ознаки f_i із цільовою змінною y;

$Corr(f_i, f_j)$ – кореляція між ознаками f_i і f_j.

2.1.4 Метод відбору ознак на основі значення дисперсії

Метод відбору ознак на основі значення дисперсії, (англ. Variance Threshold), представляє собою метод визначення релевантних та важливих ознак на основі значення дисперсії їх значень.

Згідно принципів даного методу, чим більшою є значення дисперсії значень ознак даних, тим більшою є інформативність даних, які у ній містяться. Так, за умови низьких показників дисперсії значень ознаки вважається, що значення ознаки є постійними або майже постійними, а тому на основі них важко зробити висновки про відмінність між класами або кластерами [24].

Метод відбору ознак на основі значень дисперсії у більшості випадків використовується для того, що видалити нерелевантні ознаки на перших етапах обробки даних для подальшої класифікації або кластеризації. Особливо ефективним даний метод виділення релевантних ознак є для бінарних даних, наприклад для виміру частоти появи.

Значення дисперсії ознак є ефективним показником для виміру релевантності ознак. Недоліком є те, що даний метод не враховує кореляцію між ознаками, оскільки означення порогу дисперсії обчислюється окремо для кожної ознаки [24]. Даний недолік можна виправити використанням інших методів відбору ознак.

Загальну формулу методу відбору ознак на основі значення дисперсії наведено у формулі (2.4).

$$V(i) = Var(f_i), \tag{2.4}$$

де $V(i)$ – значення порогу дисперсії для ознаки i;

$Var(f_i)$ – дисперсія ознаки f_i.

2.1.5 Метод відбору ознак на основі значення середньої абсолютної різниці

Метод відбору релевантних ознак на основі значення середньої абсолютної різниці, (англ. Mean absolute difference (MAD)), представляє собою такий метод відбору ознак, який використовує підхід оцінки середнього абсолютного відхилення від середнього значення для кожної ознаки [25]. Використання даного методу виділення значущих ознак дозволяє зробити висновки про мінливість даних для виділення різниці між класами або кластерами даних, що є корисним для алгоритмів класифікації та кластеризації.

Метод виділення ознак на основі значення середнього абсолютного відхилення від середнього значення за ознакою використовується у тих випадках, коли постає ключова необхідність у збереженні внутрішньої структури даних. Так, даний метод також підходить для використання на таких даних, які не містять цільову мітку. Даний спектр задач є досить вузьким та має таку особливість використання, як відсутність представлення зв’язку між проаналізованими ознаками, лише інформацію всередині вмісту ознаки [25].

Загальну формулу методу відбору ознак на основі значення середнього абсолютного відхилення від середнього значення за ознакою наведено у формулі (2.5).

$$MAD(i) = \frac{1}{N}\sum_{j=1}^{N}\left|f_{ij} - \overline{f_i}\right|, \tag{2.5}$$

де $MAD(i)$ – середнє абсолютне відхилення для ознаки i;

f_{ij} – значення ознаки i у j-го зразка;

f_i – середнє значення ознаки i;

N – кількість зразків.

2.1.6 Метод відбору ознак за критерієм Лапласа

Метод відбору ознак за критерієм Лапласа, (англ. Laplacian score), представляє собою такий метод відбору ознак, який відбирає релевантні та значущі ознаки на основі їх властивостей до збереження унікальної локальної структури, властивої для розглянутого набору даних [26].

Принцип дії визначення критерію Лапласа для ознак даних полягає у тому, що алгоритм виконує побудову графу подібності між різними взятими об’єктами для обраної ознаки та будує висновок, наскільки даний граф має властивість до збереження близькості даних на локальному прикладі [26]. Чим вищим є значення критерію Лапласа, тим вищим є показник здатності ознаки до збереження локальних взаємних зв’язків даних ознаки.

Для виділення релевантних ознак критерій Лапласа зазвичай використовують перед виконанням задач кластеризації даних, але не рідкими є й випадки застосування для задач класифікації. Також, даний метод демонструє високу ефективність роботи із даними зі складною структурою.

Складність використання методу критерію Лапласа полягає у тому, що для отримання ефективного результату необхідно провести детальне налаштування локальних показників регіону, входження графу подібності.

Загальну формулу методу відбору ознак на за критерієм Лапласа наведено у формулі (2.6).

$$L(i) = \frac{\sum_{j,k} W_{jk}(f_{ij} - f_{ik})^2}{Var(f_i)}, \qquad (2.6)$$

де $L(i)$ – значення критерію Лапласа для ознаки i;

W_{jk} – ваги між j-м і k-м зразками (відстань або подібність);

f_{ij} – значення ознаки i у j-го зразка.

2.1.7 Мультикластерний відбір ознак

Метод мультикластерного відбору ознак, (англ. Multi-Cluster Feature selection (MCFS)), представляє такий метод відбору ознак, згідно із яким ознаки виділяються за принципом їх сприяння до поділу даних на декілька кластерів.

За своїм функціонуванням алгоритм є наступним: спочатку виконується побудова графа за ознаками за методом k-найближчих сусідів, потім, за допомогою спектрального аналізу отриманого графа, виконується виділення ознак, які найкращим чином зберігають межі між кластерами [27].

Особливостями використання даного методу є те, що він демонструє найбільшу ефективність при розв’язанні задач кластеризації та потребує великих обчислювальних ресурсів.

Загальну формулу методу мультикластерного виділення ознак наведено у формулі (2.7).

$$S(i) = \sum_{k=1}^{K} |w_{ik}|, \tag{2.7}$$

де $S(i)$ – значення мультикластерної оцінки для ознаки i;

w_{ik} – ваговий коефіцієнт ознаки i для k-го кластеру;

K – кількість кластерів.

2.1.8 Метод відбору ознак із випадковим вибором примірників

Метод відбору ознак із випадковим вибором примірників, (англ. Relief), представляє такий метод відбору ознак, який ґрунтується на оцінці релевантності ознак для поставленої задачі на основі здатності кожної окремо взятої ознаки до

розрізнення близьких за характеристиками об’єктів, які належать до різних класів [28].

Алгоритм функціонує таким чином, що випадковим чином обирається об’єкт, наступним кроком алгоритм виконує порівняння обраного об’єкту із сусідніми об’єктами класу об’єкта та інших класів [28].

На етапі порівняння вага ознаки оновлюється відповідно до того, наскільки кожна із них може бути використана для розрізнення класів. Чим більшою є різниця між взятим випадковим об’єктом та об’єктами, сусідніми із ним, тим вищою є показник ваги.

Даний алгоритм використовується як для бінарних задач, так і для задач із великою кількістю класів.

З огляду на принцип його роботи, даний алгоритм є вимогливим до обчислювальних ресурсів, особливо для великих наборів даних, але демонструє стабільну роботу та ефективність виділення релевантних ознак.

Загальну формулу алгоритму із випадковим вибором примірників наведено у формулі (2.8).

$$W_i = W_i - \frac{1}{m}\sum_{j=1}^{m}(Diff_{hit}(f_i) - Diff_{miss}(f_i)), \quad (2.8)$$

де W_i – значення ваги для ознаки i;

m – кількість зразків;

$Diff_{hit}(f_i)$ – відстань між значеннями f_i для найближчих сусідів того ж класу;

$Diff_{miss}(f_i)$ – відстань для найближчих сусідів іншого класу.

2.2 Обгорткові методи підбору ознак для кластеризації та класифікації

Другою групою методів виділення ознак для задач класифікації та кластеризації було прийнято рішення розглянути обгорткові методи. Обгорткові методи виділення ознак ґрунтуються на тому, що для оцінки якості підмножини ознак використовується конкретна модель. Так, обгорткові методи відбору ознак виконують оцінку для підмножини ознак, на відміну від фільтрових методів, які виконували оцінку кожної ознаки окремо. Оптимальною підмножиною ознак з позиції принципу роботи обгорткових методів відбору ознак вважається такий набір ознак, який забезпечує найвищу якість роботи моделі [15].

Завдяки своєму принципу функціонування, обгорткові методи відбору ознак дозволяють врахувати взаємні зв’язки між ознаками, що, у свою чергу, позитивним чином впливає на точність та релевантність отриманих результатів.

Недоліком, натомість, є те, що оскільки алгоритм переглядає усі можливі комбінації наборів ознак, алгоритми даного типу вважаються дуже повільними, особливо для великих наборів даних [29].

Розглянемо більш детально алгоритм роботи обгорткових методів відбору ознак.

Першим кроком виконується навчання моделі на певній обрані підмножині ознак даних, наступним кроком виконується оцінка роботи методу на даному наборі ознак. Наступним кроком алгоритм порівнює отриманий результат із уже досягнутими результатами роботи. Якщо найкращого результату не було досягнуто у ході даної ітерації, алгоритм знову переходить до навчання моделі на новій підмножині ознак та оцінки результату навчання. У випадку, якщо все ж таки найкращого результату було досягнуто, набір ознак, який дав такий результат, вважається найкращим і наступним кроком проводиться тренування алгоритму із використанням даного набору ознак [29].

Схему роботи обгорткових методів відбору ознак для алгоритмів машинного навчання, зокрема класифікації та кластеризації, відповідно до створеного опису, наведено на рисунку 2.2.

Рисунок 2.2 – Алгоритм роботи обгорткових методів відбору ознак

Зазвичай обгорткові методи відбору ознак використовують у тих випадках, коли найбільшим приорітетом результату є саме його точність, а кількість ознак не є надмірною.

2.2.1 Стратегії відбору ознак, використані в обгорткових методах

У ході своєї роботи обгорткові методи відбору ознак використовують різні стратегії. Найпопулярнішими з них, зокрема, є стратегії прямого та зворотнього відбору ознак. Розглянемо детальніше кожну із них.

Стратегія прямого відбору релевантних ознак характеризується поступовим доданням інформативних ознак до набору. Таким чином, якщо робота прямого методу розпочинається із пустого набору ознак, на кожному

наступному кроці роботи прямого методу відбору ознак додається нова ознака, яка позитивним чином впливає на ефективність роботи моделі [12]. Такий принцип відбору ознак є ефективним у випадках, коли необхідно зменшити розмірність даних для наборів із великою кількістю ознак.

Стратегія зворотнього відбору релевантних ознак характеризується зворотнім порядком перебору ознак. Починаючи із усієї сукупності ознак, наявних у розглянутих даних, алгоритм видаляє у ході кожного кроку такі ознаки, які мають найменший вплив на ефективність отриманих результатів [12]. Такий принцип відбору ознак також має свої переваги, оскільки розглядається уся інформація та відсівається лише найменш інформативна та значуща її частина.

Вибір стратегії відбору для обгорткових методів залежить від особливостей вхідних даних, поставленої задачі дослідження та приорітетів отриманого результату.

Таким чином, вибір стратегії прямого відбору ознак є аргументованим для тих випадків, коли набір даних має високу розмірність, але кількість ресурсів для виконання обчислень є дуже обмеженою. Натомість вибір стратегії зворотнього відбору буде доцільним для тих випадків, коли критично необхідно урахувати усі значущі ознаки для досягнення максимальної точності моделі [12].

2.2.2 Метод рекурсивного виключення ознак

Метод відбору ознак шляхом рекурсивного виключення, (англ. Recursive Feature Elimination (RFE)), це обгортковий метод, який використовує рекурсивний підхід до оцінки та виключення ознак із найменшою релевантністю.

Алгоритм рекурсивного виключення ознак працює за таким принципом. Модель тренується на повному наборі ознак, наявних у вхідному наборі даних. Після того, як тренування було завершене, оцінюється важливість та

інформативність ознаки у рамках поставленої задачі дослідження, ознаки, які було оцінено як найменш важливі, виключаються із набору [30]. Таким чином реалізовано зворотній відбір ознак. Кожної наступної ітерації найменш важливі ознаки будуть виключатись із набору до тих пір, поки не залишиться лише оптимальний набір ознак.

Недоліками використання даного методу є його висока обчислювальна вартість, оскільки він виконує повний перебір усіх ознак зворотнім методом. Зі збільшенням набору даних, вимогливість алгоритму до обчислювальних ресурсів відповідно зростає [30]. Але найголовнішою перевагою використання методу рекурсивного виключення ознак є висока точність отриманого результату як для класифікації, так і для кластеризації.

Загальну формулу методу відбору релевантних ознак шляхом рекурсивного виключення ознак наведено у формулі (2.9).

$$Rank(f_i) \propto Coef_i, \tag{2.9}$$

де $Rank(f_i)$ – ранг ознаки f_i;

$Coef_i$ – коефіцієнт ознаки f_i у моделі (наприклад, ваговий коефіцієнт).

2.3 Вбудовані методи підбору ознак для кластеризації та класифікації

Третьою групою методів виділення ознак для задач класифікації та кластеризації було прийнято рішення розглянути вбудовані методи. Дана група методів виділення ознак характеризується поєднанням властивостей фільтрових та обгорткових методів, а саме реалізацію процесу оцінки та відбору ознак у ході навчання моделі [31]. Завдяки такому підходу, вбудовані методи виділення ознак

одно виконують навчання і оптимізацію даних моделі, що, у свою чергу, значним чином економить ресурси обчислення та скорочує час роботи.

Процес суміщеного навчання та відбору ознак у вбудованих методах відбору ознак реалізовано у більшості випадків завдяки алгоритмам автоматичної регуляризації. Такий принцип дії робить вбудовані методи універсальними для обробки великих наборів даних, забезпечуючи не лише ефективність роботи, а й точність отриманого результату.

Схему роботи вбудованих методів відбору ознак для алгоритмів машинного навчання, зокрема класифікації та кластеризації, наведено на рисунку 2.3.

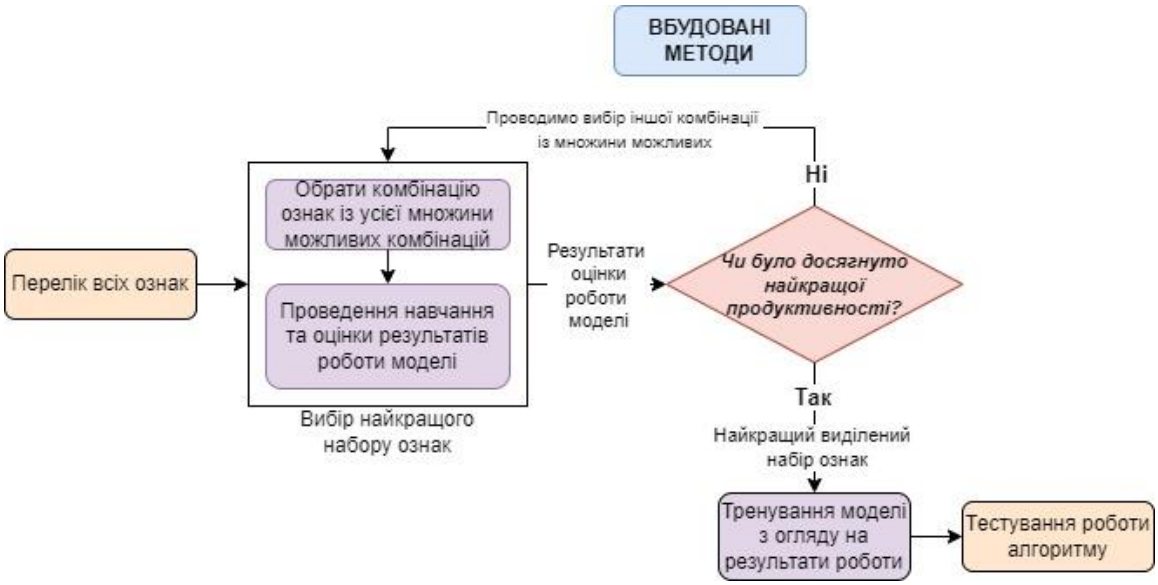


Рисунок 2.3 – Алгоритм роботи вбудованих методів відбору ознак

Вбудовані методи відбору ознак також мають й деякі обмеження. Так, результат роботи даних методів може значно залежати від алгоритмів, обраних для проведення навчання.

У окрему групу часто виділяють та порівнюють методи, які використовують L_1 та L_2 норми [32]. Суть даного порівняння полягає у тому, що дані методи різним чином обмежують коефіцієнти, які надаються ознакам під час

обробки за допомогою алгоритму. Так, як це наведено на рисунку 2.4, можна побачити, що метод оператора найменшого абсолютного стиснення та відбору, який використовує L_1 норму, обмежує коефіцієнти формою квадрата, метод еластичної сітки, який використовує поєднання норм L_1 та L_2, знаходиться посередині, поєднуючи вершини умовного квадрату норми L_1, та метод еластичної сітки, який використовує L_2 норму, чітко описує коло [32].

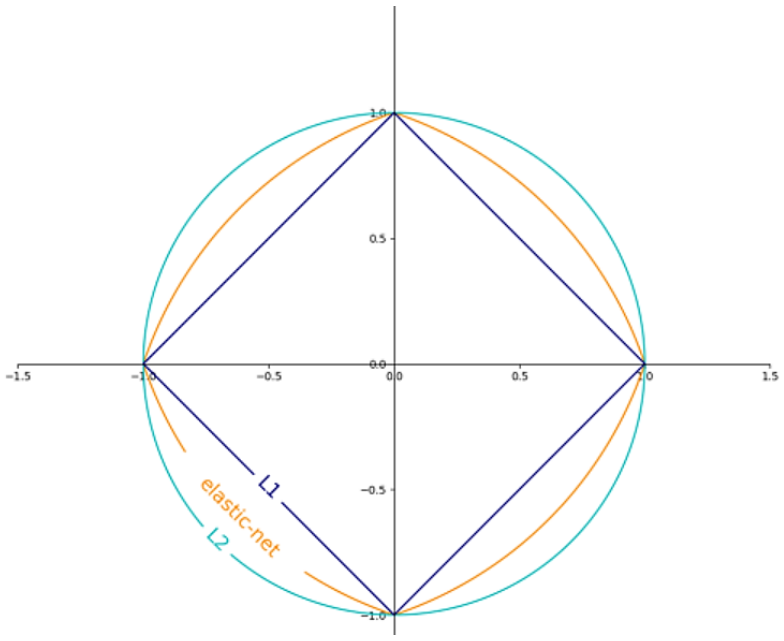


Рисунок 2.4 – Порівняння методів обмеження коефіцієнтів за допомогою методів L_1, L_2 та еластичної сітки

2.3.1 Метод розріженої мультиноміальної логістичної регресії

Метод відбору ознак за допомогою розріженої мультиноміальної логістичної регресії, (англ. Sparse Multinomial Logistic Regression (SMLR)), представляє собою такий метод відбору ознак, який поєднує засоби логісничної регресії та поняття розрідження даних для створення багатокласової класифікації класів даних.

Метод відбору ознак за допомогою розріженої мультиноміальної логістичної регресії завдяки своїй архітектурі, як і всі вбудовані методи, також використовує засоби регуляризації L_1 для виділення значущих ознак у процесі навчання моделі [33].

Даний метод найчастіше використовується для великих наборів даних, оскільки метод відбору ознак за допомогою розріженої мультиноміальної логістичної регресії відмінно справляється із багатокласовими даними та дозволяє дослідити вплив різних ознак на вихідні результати, а також взаємний зв'язок ознак даних [33].

Загальну формулу методу відбору ознак за допомогою розріженої мультиноміальної логістичної регресії наведено у формулі (2.10).

$$\min\|w\|_1 + \sum_{j=1}^{N} \log\left(1 + \exp\left(-y_j * (w \intercal x_i)\right)\right), \tag{2.10}$$

де w – ваги ознак;

y_j – цільове значення для j-го зразка;

x_j – ознаки j-го зразка.

2.3.2 Метод регресії автоматичного виділення релевантності

Метод виділення ознак шляхом регресії автоматичного виділення релевантності, (англ. Automatic Relevance Determination (ARD)), представляє собою метод виділення ознак, заснований на принципах регресії та L_2 регуляризації [34].

Із використанням даного підходу до виділення релевантних ознак, кожна ознака отримує свій ваговий коефіцієнт та регулятор важливості ознаки.

Відповідно до впливу ознак на побудову прогнозу, алгоритм зменшує вагу ознаки та значення регулятора важливості. Даний принцип реалізує принцип баєсівського навчання для встановлення ймовірнісних розподілів, що допомагає визначити, які ознаки є релевантними, а які можна вважати неінформативними для прогнозування даної окремо взятої задачі [34]. Згідно із алгоритмом, ознаки із меншими значеннями ваг поступово виключаються із переліку, натомість ознаки із найбільшими показниками ваг формують перелік релевантних ознак для розв'язання поставленої задачі дослідження.

Даний метод, як і інші вбудовані методи є ефективним як на невеликих наборах даних, так і на великих, які містять значну кількість ознак даних.

Загальну формулу методу виділення ознак за допомогою регресії автоматичного виділення релевантності наведено у формулі (2.11).

$$p(w|\alpha) \propto \exp(-\frac{\alpha}{2}\|w\|^2) , \qquad (2.11)$$

де w – ваги ознак;

α – параметр релевантності.

2.3.3 Метод відбору релевантних ознак за допомогою оператора найменшого абсолютного стиснення та відбору

Метод відбору релевантних ознак за допомогою оператора найменшого абсолютного стиснення та відбору, (англ. Lasso (Least Absolute Shrinkage and Selection Operator)), представляє собою метод відбору ознак, який ґрунтується на використанні засобів регресії та L_1 регуляризації для визначення значущості ознаки для поставленої задачі дослідження [35].

Ключовим принципом роботи методу відбору релевантних ознак за допомогою оператора найменшого абсолютного стиснення та відбору є використання у моделі штрафного механізму, значення якого є пропорційним до суми абсолютних значень коефіцієнтів ознак [35]. Таким чином, унаслідок використання такого механізму, коефіцієнти ознак розріджуються та стають нульовими у деяких випадках. Ознаки, що мають низькі значення коефіцієнтів, як і в інших методах відбору ознак, усуваються.

Метод відбору ознак за допомогою оператора найменшого абсолютного стиснення та відбору використовується для того спектру задач, коли критично необхідно залишити лише мінімальний набір максимально ефективних ознак, наприклад, для задач бізнесу із надзвичайно великими даними.

Загальну формулу методу відбору релевантних ознак за допомогою оператора найменшого абсолютного стиснення та відбору наведено у формулі (2.12).

$$min\|w\|_1 + \frac{1}{2}\|y - X_w\|_2^2 \quad , \tag{2.12}$$

де w – ваги ознак;

y – цільові значення;

X – матриця ознак.

2.3.4 Метод гребневої регресії

Метод відбору релевантних ознак за допомогою гребневої регресії, (англ. Ridge regression), представляє такий метод відбору ознак, який спрямований на

виявлення та видалення мультиколінеарних або дублюючих ознак. Даний метод використовує L_2 регуляризацію та є захищеним від перенавчання [36].

Принцип дії методу відбору ознак шляхом застосування гребневої регресії полягає в тому, що даний метод проводить згаджування та урівноваження ваги ознак та виміру їх впливу на результат прогнозування на наборі ознак. Це створюється шляхом накладання штрафів на ваги ознак, що рівні квадрату вагових ознак [36].

Так, відповідно до свого принципу роботи, метод не усуває ознаки, але зменшує їх вплив на вихідний результат. Даний принцип дії алгоритму гребневої регресії є важливим у тих випадках, коли важливо залишити усі ознаки даних, оскільки вони є важливими для розв'язання поставленої задачі дослідження. Також даний метод добре обробляє корельовані ознаки.

Загальну формулу методу відбору релевантних ознак за допомогою гребневої регресії наведено у формулі (2.13).

$$min\frac{\lambda}{2}\|w\|_2^2 + \frac{1}{2}\|y - X_w\|_2^2 \ , \qquad (2.13)$$

де w – ваги ознак;

y – цільові значення;

X – матриця ознак;

λ – параметр регуляризації.

2.3.5 Метод еластичної сітки

Метод відбору ознак шляхом еластичної сітки, (англ. Elastic Net), представляє собою такий метод відбору ознак, який поєднує властивості методів

гребневої регресії та оператора найменшого абсолютного стиснення та відбору та використовує L_1 та L_2 регуляризацію [37].

Принцип дії алгоритму полягає в тому, що він представляє баланс між повним виключенням із переліку релевантних ознак для розв'язання задачі та нівелюванням впливу ознак на результати прогнозування із використанням виділеного набору ознак.

Даний принцип реалізовано завдяки L_1 та L_2 регуляризації, оскільки вони реалізують різні типи штрафних функцій у ході навчання моделі: перший ґрунтується на абсолютних значеннях коефіцієнтів ознак, а інший – на квадратах коефіцієнтів [37].

Використання даного методу відбору ознак є аргументованим для тих наборів даних, які містять надзвичайно велику кількість ознак, а також містити перекриття даних, а також велику кількість ознак, які впливають на цільову та результати прогнозування на обраній підмножині релевантних ознак. Даний метод також ефективно обробляє висококорельовані ознаки.

Загальну формулу методу відбору ознак шляхом еластичної сітки наведено у формулі (2.14).

$$min\lambda_1\|w\|_1 + \frac{\lambda_2}{2}\|w\|_2^2 + \frac{1}{2}\|y - X_w\|_2^2, \qquad (2.14)$$

де w – ваги ознак;

y – цільові значення;

X – матриця ознак;

λ_1, λ_2 – параметри регуляризації.

2.4 Гібридні методи

Гібридний підхід до відбору ознак не є традиційним методом відбору ознак для задач кластеризації та класифікації. Методи даного типу є поєднаннями засобів та підходів, використаних у фільтрових та обгорткових методах для досягнення загального компромісу та балансу у питаннях ефективності обчислень та точності отриманих прогнозів [38].

Загальний принцип функціонування гібридних методів відбору ознак характеризується тип, що алгоритм у першу чергу проводить базовий відбір ознак за допомогою статистичних критеріїв, як у фільтрових методах, а потім підсилює його глибшим аналізом, використовуючи модель, як у обгорткових методах [18].

Такий підхід до відбору ознак дає можливість у першу чергу відсіяти більшість нерелевантних ознак і зосередити обчислювальні ресурси на подальшій оптимізації тих, що залишилися і мають найбільшу ймовірність того, щоб бути релевантними [18].

Такий підхід до розв’язання задачі відбору ознак для задач класифікації та кластеризації знижує обчислювальні витрати, що є перевагою для наборів даних із великою розмірністю, та дозволяє досягти більш точного відбору ознак для задач, де потрібна детальна обробка даних і висока якість прогнозування моделі.

2.5 Порівняння розглянутих методів виділення ознак для задач класифікації та кластеризації

З огляду на описані вище типи методів виділення ознак для класифікації та кластеризації можна зробити узагальнені висновки за тим, у якому випадку було б доречно використовувати той чи інший тип виділення ознак.

Для цього було створено узагальнення отриманих даних, наведене у таблиці 2.1. Наведена таблиця містить відомості про кожний із принципів відбору ознак для класифікації та кластеризації даних та його показники, розподілені за різними критеріями порівняння методів.

Таблиця 2.1 – Узагальнення призначення та використання виділених типів виділення ознак для класифікації та кластеризації

№	Критерій	Фільтрові методи	Обгорткові методи	Вбудовані методи	Гібридні методи
1	**2**	**3**	**4**	**5**	**6**
1	Принцип роботи алгоритму	Виконують окрему оцінку кожної ознаки	Виконують побудову моделі для кожної комбінації ознак	Виконують відбір ознак, інтегрований із навчанням моделі	Виконують попередній відбір ознак за допомогою фільтрації та уточнення за допомогою обгорткового методу
2	Точність отриманих результатів	Низька для сильно корельованих ознак	Висока через урахування кореляції між ознаками	Середня або висока, залежить від алгоритму	Висока через комбінацію методів
3	Потреба в обчислювальних ресурсах	Низька	Висока	Середня	Середня, але залежить від методів
4	Виділені переваги	Швидкість роботи; простота реалізації алгоритму; підходить для початкового відбору ознак	Висока точність отриманих результатів за умови невеликої кількості ознак	Ефективність обчислень для даних із великою кількістю ознак	Висока точність отриманих результатів
5	Виділені недоліки	Не враховується кореляція ознак	Складність та технічна важкість обчислень для великої кількості ознак у наборі даних	Отриманий результат сильно залежить від типу обраної моделі та алгоритму	Висока складність реалізації та налаштування отриманий результат сильно залежить від обраних методів

Продовження таблиці 2.1

№	Критерій	Фільтрові методи	Обгорткові методи	Вбудовані методи	Гібридні методи
1	**2**	**3**	**4**	**5**	**6**
6	Випадки застосування	Для великих наборів даних, які необхідно швидко обробити	Для наборів даних із середньою кількістю ознак та потребою у максимальній точності результатів	Для великих наборів даних, де приорітетом є обчислювальна ефективність моделі	Для середніх та великих наборів даних для отримання точних результатів, але без перевантаження апаратної частини

2.6 Підбір критеріїв для порівняння якості результатів роботи задач класифікації та кластеризації на обраному наборі ознак

З метою виконання порівняння отриманих результатів за допомогою різноманітних виділених метрик підбору ознак було прийнято рішення про використання ряду метрик для оцінки якості класифікації та кластеризації на основі виділеного набору ознак. Самі ж методи відбору ознак також можна порівняти за значенням показника швидкості роботи кожного з методів.

Варто зазначити, що даний процес буде проходити у такій послідовності:

– першим кроком необхідно виділити такий набір даних на якому можна протестувати як ефективність методів підбору ознак, так і роботу методів класифікації, так і методів кластеризації;

– другим кроком необхідно створити реалізацію методу відбору ознак;

– третім кроком передаємо перелік ознак, отриманий на минулому етапі, у результаті роботи методу відбору ознак, алгоритму класифікації та кластеризації;

– четвертим кроком виконаємо порівняння отриманих результатів.

Для оцінки результатів виконання класифікації на виділеному наборі ознак на етапі виділення значущих ознак для розв’язання задачі було обрано наступний перелік метрик оцінки:

– accuracy – точність, частка правильно класифікованих об’єктів від їх загальної частки;

– precision – точність передбачення, частка правильно передбачених позитивних результатів від загальної кількості позитивних результатів;

– recall – повнота, частка правильно передбачених позитивних об’єктів серед усіх фактичних позитивних об’єктів;

– F1 score – метрика балансування відношення між точністю передбачення та повнотою передбаченого результату;

– ROC-AUC – площа під кривою помилок, дозволяє зробити висновки стосовно того, наскільки модель у випадку бінарної класифікації добре відрізняє класи. Так, ключовим показником метрики є те, наскільки ймовірним є надання вищого балу випадковому позитивному прикладу на противагу випадковому негативному прикладу;

– швидкість класифікації – швидкість роботи алгоритму класифікації на обраному наборі ознак.

Для оцінки результатів виконання кластеризації на виділеному наборі ознак на етапі виділення значущих ознак для розв’язання задачі було обрано наступний перелік метрик оцінки:

– silhouette score – показник силуету, міра того, наскільки добре об’єкт належить до свого кластеру порівняно із іншими кластерами;

– Davies-Bouldin index – індекс Девіса-Булдіна, міра відношення відстані між кластерами до їх ширини. Приймає до уваги дисперсію значень всередині кожного кластеру, а також відстань між центрами розглянутих кластерів;

– швидкість кластеризації – швидкість роботи алгоритму кластеризації на обраному наборі ознак.

3 ДОСЛІДЖЕННЯ ВИДІЛЕНИХ МЕТОДІВ ВІДБОРУ ОЗНАК ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ ТА КЛАСТЕРИЗАЦІЇ

3.1 Опис використаних програмних засобів для дослідження методів відбору ознак

Для створення програмної реалізації порівняльного аналізу обраного переліку методів відбору ознак для класифікації та кластеризації було обрано мову програмування Python та середовище розробки Jupyter Notebook.

Серед усіх мов програмування було обрано саме Python оскільки дана мова програмування має широке розповсюдження у науковому середовищі та має велику та розгалуджену систему бібліотек для спрощення створення програмної реалізації складних методів, зокрема й методів відбору ознак, а також зручний та лаконічний синтаксис для виконання маніпуляцій із даними різних типів. Для прикладу наведемо далі бібліотеки, які було використано у ході створення програмної реалізації виділених методів відбору ознак для задач класифікації та кластеризації.

Бібліотека Python NumPy – це бібліотека, яка використовується для виконання роботи із багатовимірними масивами даних, а також використовується для виконання для виконання математичних обчислень, що необхідно для усіх методів відбору ознак для задач класифікації та кластеризації.

Бібліотека Python Pandas – це бібліотека, яка використовується для виконання роботи із даними таблиць типу DataFrame. Такий тип представлення є не тільки швидким для обробки, а й зручним у представленні. Дана бібліотека також надає ряд функцій для легкої обробки ознак та міток ознак, а також безпосередньо результати дослідження методів відбору ознак для задач

класифікації та кластеризації було записано до DataFrame та експортовано до .csv файлу.

Бібліотека Python SciPy – це бібліотека, яка використовується для виконання роботи із статистикою. У рамках даної роботи визначену бібліотеку було використано для реалізації методів відбору ознак, пов’язаних із статистикою.

Бібліотека Python Scikit-learn – це бібліотека, яка використовується для виконання роботи із алгоритмами машинного навчання та методів відбору ознак. У рамках даної роботи дану бібліотеку було використано для використання створених програмних реалізації вбудованих, обгорткових та деяких фільтрових методів відбору ознак для задач класифікації та кластеризації.

Бібліотека Python Statsmodels – це бібліотека, яка використовується для виконання роботи із статистичними моделями та моделями регресії, для чого її й було використано у рамках дослідження методів відбору ознак для задач класифікації та кластеризації.

Бібліотека Python Matplotlib – це бібліотека, яка використовується для виконання роботи із базовими візуалізаціями отриманих результатів. У рамках даного дослідження дану бібліотеку було використано для створення площини та сітки графіків матриць теплових карт результатів прогнозів розглянутих методів відбору ознак для задач класифікації та кластеризації.

Бібліотека Python Seaborn – це бібліотека, яка використовується для виконання роботи зі створення інформативних графіків. У рамках даного дослідження дану бібліотеку було використано для побудови теплових карт результатів прогнозування методів класифікації на основі ознак, отриманих за допомогою розглянутих методів відбору ознак для задач класифікації та кластеризації.

Бібліотека Python Tensorflow – це бібліотека, яка використовується для виконання роботи із нейронними мережами та налаштуванням їх структури. У

рамках даного дослідження дану бібліотеку також було використано для реалізації мультикластерного методу відбору ознак для задач класифікації та кластеризації.

Бібліотека Python Feature-engine – це бібліотека, яка використовується для виконання роботи із ознаками, зокрема, для реалізації відбору релевантних ознак, а також має функціонал для обробки взаємопов'язаних, або корельованих, ознак, відібраних у результаті роботи методів відбору ознак для задач класифікації та кластеризації.

Середовище розробки Jupyter Notebook було обрано через те, що дане середовище дозволяє виконувати програмний код логічними блоками, через що дуже зручно тестувати програмний код за ходом його написання, а також отримувати проміжні результати кожного етапу, або логічного блоку коду. Також дане середовище має підтримку графічного виведення та використання інтерактивних візуалізацій.

Перелік створених підключень бібліотек та програмних пакетів, використаних у ході даного дослідження методів відбору ознак для класифікації та кластеризації наведено на рисунку 3.1.


Усі імпорти проекту було винесено в один блок

```
import pandas as pd
import numpy as np
import time
import matplotlib.pyplot as plt
import seaborn as sns
import ReliefF
from sklearn.metrics import confusion_matrix
from sklearn.datasets import load_breast_cancer
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.linear_model import LogisticRegression
from sklearn.cluster import KMeans
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix,
    silhouette_score, davies_bouldin_score
)
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import VarianceThreshold
from sklearn.neighbors import NearestNeighbors
from scipy.sparse import csgraph
from sklearn.feature_selection import chi2
from sklearn.neighbors import kneighbors_graph
from scipy.sparse.linalg import eigsh
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectKBest
from skfeature.function.similarity_based import reliefF
from sklearn.feature_selection import RFE
from sklearn.linear_model import ARDRegression
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.linear_model import Ridge
```

Рисунок 3.1 – Перелік створених підключень бібліотек та програмних пакетів, використаних у ході створення програмної реалізації методів відбору ознак

3.2 Опис вхідних даних для проведення дослідження методів відбору ознак

Для тестування методів відбору ознак було обрано бібліотечний набір даних, який описує параметри ракових пухлин. Обраний набір даних є вбудованим набором даних бібліотеки Scikit-learn, пакету datasets. У наведеному пакеті даний набір даних називається breast_cancer.

Перейдемо до опису вмісту датасету, який, як вже було зазначено містить параметри ракових пухлин раку грудей, зібраних у штаті Вісконсін, США, та оприлюдненого у листопаді 1995 року [39]. Даний набір даних містить 569 строк даних та 30 ознак даних. Ознаки даних представляють собою числові значення параметрів ракових пухлин, отриманих у ході їх ультразвукового дослідження (УЗД) для встановлення типу пухлини. Відповідно до цього, обраний набір даних містить два цільових класи, доброякісні та злоякісні пухлини, як це визначено у офіційній документації до набору даних, Benign та Malignant. Обраний набір даних не містить пропущених значень, а розподіл класів доброякісних та злоякісних випадків у наборі даних є 357 та 212 [39]. Таким чином, розподіл класів не є рівномірним, що важливо знати для аналізу результатів дослідження.

Як вже було зазначено, обраний набір даних аналізу параметрів ракових пухлин містить 30 числових ознак даних. Для кожного тестового зображення ракової пухлини було розраховано три показники за кожним із критеріїв: середнє значення, стандартну помилку та найгірше, або найбільше, значення. Характеристиками, за якими було обчислено по три показники є характеристики геометричних та текстурних ознак клітинних утворень пухлин, ознак симетрії клітинних утворень, також статистичні характеристики тенденцій ознак [39].

Переглянувши перелік ознак, можемо зробити висновок, що даний набір даних не містить дублюючих або мультиколінеарних ознак, тобто усі наведені ознаки описують різні сторони розглянутого предмету.

Опис даних ознак відповідно до офіційної документації набору даних є наступним [39]:

– radius – радіус, середнє значення відстаней від центру до точок на периметрі клітинних утворень;

– texture – текстура клітинних утворень, значення стандартного відхилення, розраховане завдяки градаціям сірого на знімку;

– perimeter – периметр клітинних утворень;

– area – площа клітинних утворень;

– smoothness – гладкість фактури клітинних утворень, вимірюється як локальна варіація довжин радіусів утворень розглянутої області пухлини;

– compactness – компактність клітинних утворень, вимірюється як відношення квадрату периметру до площі мінус один;

– concavity – вгнутість клітинних утворень, тобто яскравість вираження вгнутості окремих ділянок контуру клітинного утворення або контуру пухлини в цілому;

– concave points – вгнуті точки клітинного утворення, визначає кількість точок перегибу ділянок контуру клітинного утворення або контуру пухлини в цілому;

– symmetry – симетрія клітинного утворення або пухлини в цілому;

– fractal dimension – фрактальна ромірність, показник виміру складності та неоднорідності структури пухлини.

Для кожного із наведених вище показників було розраховано, як вже визначено вище, три значення, серед яких:

– mean – середнє значення, значення загальної тенденції значень для кожної ознаки для досліджуваного екземпляру;

– standard error – стандартна помилка, міра розсіювання показників кожної ознаки для досліджуваного екземпляру;

– worst – найгірше значення показника за ознакою, відображає значення екстремуму, що може бути важливим для визначення типу патології.

Стосовно кореляцій між ознаками, можна зробити припущення про те, що усі ознаки корелюють між собою у тій чи іншій мірі, оскільки вони описують різні сторони одного об'єкту.

Провівши аналіз обраного набору даних, можна сформулювати постановку завдання дослідження даних, а саме створити модель, яка за параметрами ракової пухлини грудей може визначити тип пухлини. Для того, щоб реалізувати задачу класифікації за даним набором даних необхідно додати мітки класів, де 0 – доброякісна пухлина, а 1 – злоякісна. Для задачі кластеризації виконується поділ також за визначеними класами.

Для демонстрації вмісту визначеного набору даних, завантажимо дані до зручного представлення DataFrame та продемонструємо перші 5 строк даних. На рисунку 3.2 також можна побачити, що для виконання роботи було створено новий стовбець міток цілей target.

```
import pandas as pd
from sklearn.datasets import load_breast_cancer
# Завантаження даних
data = load_breast_cancer()
# Створення DataFrame
df = pd.DataFrame(data.data,
                  columns=data.feature_names)
# Додаємо мітки класів
df['target'] = data.target
# Виводимо перші 5 строк
df.head()
```

mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension	target
0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890	0
0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902	0
0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758	0
0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300	0
0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678	0

Рисунок 3.2 – Демонстрація завантаження даних з метою представлення структури та вмісту даних

3.3 Створення програмної реалізації методів відбору ознак

Першим кроком виконуємо завантаження даних та розділення на X та y. Для завантажених даних отримуємо форму даних, яка за даними на рисунку 3.4, відповідає формі, заявленій в офіційній документації набору даних.

Також виконуємо крок зі створення пустої структури таблиці типу DataFrame, яка призначена для збереження даних досліджень методів відбору ознак. Створена структура даних буде заповнюватись за ходом дослідження даних та буде містити наступні критерії порівняння методів:

– Feature_selection_group - назва групи методів відбору ознак, можливі групи Filter_method, Wrapper_method, Embedded_method;

– Feature_selection_method - назва методу відбору ознак;

– Feature_selection_speed - швидкість роботи методу відбору значущих ознак;

– Classification_accuracy - точність, частка правильно класифікованих об’єктів від їх загальної частки;

– Classification_precision - точність передбачення, частка правильно передбачених позитивних результатів від загальної кількості позитивних результатів;

– Classification_recall - повнота, частка правильно передбачених позитивних об’єктів серед усіх фактичних позитивних об’єктів;

– Classification_F1_score - метрика балансування відношення між точністю передбачення та повнотою передбаченого результату;

– Classification_roc_auc - площа під кривою помилок

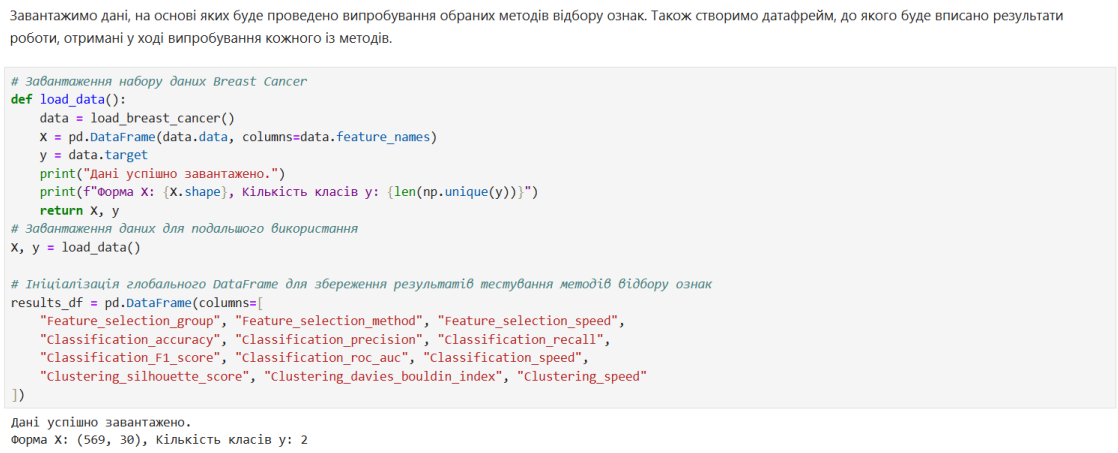
– Classification_speed - швидкість роботи методу класифікації на обраному наборі значущих ознак;

– Clustering_silhouette_score - показник силуету, міра того, наскільки добре об’єкт належить до свого кластеру порівняно із іншими кластерами;

– Clustering_davies_bouldin_index - індекс Девіса-Булдіна, міра відношення відстані між кластерами до їх ширини. Приймає до уваги дисперсію значень всередині кожного кластеру, а також відстань між центрами розглянутих кластерів;

– Clustering_speed - швидкість роботи методу кластеризації на обраному наборі значущих ознак.

Усі описані кроки було реалізовано. Код реалізації завантаження даних та створення набору даних для порівняння методів наведено на рисунку 3.3.



Завантажимо дані, на основі яких буде проведено випробування обраних методів відбору ознак. Також створимо датафрейм, до якого буде вписано результати роботи, отримані у ході випробування кожного із методів.

```
# Завантаження набору даних Breast Cancer
def load_data():
    data = load_breast_cancer()
    X = pd.DataFrame(data.data, columns=data.feature_names)
    y = data.target
    print("Дані успішно завантажено.")
    print(f"Форма X: {X.shape}, Кількість класів y: {len(np.unique(y))}")
    return X, y
# Завантаження даних для подальшого використання
X, y = load_data()

# Ініціалізація глобального DataFrame для збереження результатів тестування методів відбору ознак
results_df = pd.DataFrame(columns=[
    "Feature_selection_group", "Feature_selection_method", "Feature_selection_speed",
    "Classification_accuracy", "Classification_precision", "Classification_recall",
    "Classification_F1_score", "Classification_roc_auc", "Classification_speed",
    "Clustering_silhouette_score", "Clustering_davies_bouldin_index", "Clustering_speed"
])
```

```
Дані успішно завантажено.
Форма X: (569, 30), Кількість класів y: 2
```

Рисунок 3.3 – Код реалізації завантаження даних та створення набору даних для порівняння методів та результати його виконання

Наступним кроком переходимо до створення програмних реалізацій обраних методів відбору ознак. Першим кроком будемо створювати програмні реалізації фільтрових методів відбору ознак.

Переходимо до створення програмної реалізації методу відбору ознак за критерієм Фішера. Даний метод обчислює відношення міжкласової та внутрішньокласової варіації для кожної ознаки. У програмній реалізації було додано перевірку категоріальності ознак, оскільки метод працює лише із числовими даними. Для уникнення проблем через масштаб ознак застосовано нормалізацію, що забезпечило коректне порівняння різнорідних даних. Першим

кроком створення програмної реалізації методу відбору ознак за критерієм Фішера створимо код функції виконання методу, що наведено на рисунку 3.4.

```
# Реалізація методу Fisher Score
def fisher_score_analysis(X, y):
    method_name = "Fisher Score"
    feature_selection_group = "Filter_method"

    # === 1. Відбір ознак ===
    start_time = time.time()  # Початок вимірювання часу відбору
    selector = SelectKBest(score_func=f_classif, k=10)
    X_selected = selector.fit_transform(X, y)
    feature_selection_speed = time.time() - start_time  # Час виконання методу відбору
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.4 – Код функції методу відбору ознак за критерієм Фішера

Наступним кроком на основі відібраного набору ознак проведемо класифікацію та кластеризацію. У рамках даного дослідження для класифікації та кластеризації у всіх методах було використано традиційні та найбільш популярні методи, метод логістичної регресії (англ. Logistic Regression) для задач класифікації та метод к-середніх для задач кластеризації (англ. K-Means).

Метод логістичної регресії для задач класифікації було обрано через те, що даний метод має високу ефективність роботи та інтерпретованість результатів. Окрім виконання прогнозування класів даний метод також оцінює ступінь впевненості в отриманому рішенні, а також дозволяє використання метрик для оцінки результатів прогнозування [40].

Метод к-середніх для задач кластеризації було обрано через те, що це один із найбільш популярних і простих у використанні алгоритмів [41]. Даний алгоритм добре підходить для розбиття даних на групи. Даний метод також характеризується високою швидкістю роботи та простоті використання та підтримує використання метрик оцінки якості проведеної кластеризації.

На рисунку 3.5 представлено виконання класифікації даних ознак, отриманих за критерієм Фішера та оцінку результатів проведеної класифікації.

На рисунку 3.6 представлено побудову матриці помилок для результатів класифікації даних ознак, отриманих за критерієм Фішера.

```
# === 2. Класифікація ===
start_time = time.time()
clf = LogisticRegression(max_iter=1000, random_state=42)
clf.fit(X_selected, y)
y_pred = clf.predict(X_selected)
y_prob = clf.predict_proba(X_selected)[:, 1]
classification_speed = time.time() - start_time

# Розрахунок метрик класифікації
classification_accuracy = accuracy_score(y, y_pred)
classification_precision = precision_score(y, y_pred)
classification_recall = recall_score(y, y_pred)
classification_f1 = f1_score(y, y_pred)
classification_roc_auc = roc_auc_score(y, y_prob)

# Візуалізація нормалізованої матриці помилок
plot_confusion_matrix(y, y_pred, method_name=method_name)
```

Рисунок 3.5 – Виконання класифікації даних ознак, отриманих за критерієм Фішера та оцінка результатів класифікації

На рисунку 3.6 представлено побудову матриці помилок для результатів класифікації даних ознак, отриманих за критерієм Фішера.

```
# Функція для побудови матриці помилок
def plot_confusion_matrix(y_true, y_pred,
                          class_names=["Benign", "Malignant"], method_name=""):
    cm = confusion_matrix(y_true, y_pred)
    cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
    plt.figure(figsize=(6, 5))
    sns.heatmap(cm_normalized, annot=True, fmt='.2f', cmap='Blues',
                xticklabels=class_names, yticklabels=class_names)
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.title(f"Normalized Confusion Matrix: {method_name}")
    plt.show()
```

Рисунок 3.6 – Побудова матриці помилок для результатів класифікації даних ознак, отриманих за критерієм Фішера

На рисунку 3.7 представлено виконання кластеризації даних ознак, отриманих за критерієм Фішера та оцінку якості проведеної кластеризації.

```
# === 3. Кластеризація ===
start_time = time.time()
kmeans = KMeans(n_clusters=2, random_state=42)
labels = kmeans.fit_predict(X_selected)
clustering_speed = time.time() - start_time

# Розрахунок метрик кластеризації
clustering_silhouette_score = silhouette_score(X_selected, labels)
clustering_davies_bouldin_index = davies_bouldin_score(X_selected, labels)
```

Рисунок 3.7 – Виконання кластеризації даних ознак, отриманих за критерієм Фішера та оцінка якості проведеної кластеризації

На рисунку 3.8 представлено виконання запису результатів роботи методу виділення ознак за критерієм Фішера до таблиці результатів results_df.

```
    # === 4. Запис результатів у DataFrame ===
    global results_df
    new_row = pd.DataFrame({
        "Feature_selection_group": [feature_selection_group],
        "Feature_selection_method": [method_name],
        "Feature_selection_speed": [feature_selection_speed],
        "Classification_accuracy": [classification_accuracy],
        "Classification_precision": [classification_precision],
        "Classification_recall": [classification_recall],
        "Classification_F1_score": [classification_f1],
        "Classification_roc_auc": [classification_roc_auc],
        "Classification_speed": [classification_speed],
        "Clustering_silhouette_score": [clustering_silhouette_score],
        "Clustering_davies_bouldin_index": [clustering_davies_bouldin_index],
        "Clustering_speed": [clustering_speed]
    })

    results_df = pd.concat([results_df, new_row], ignore_index=True)

    # Вивід підсумків
    print(f"Метод {method_name}: результати записано до DataFrame.")

# === Виконання для методу Fisher Score ===
fisher_score_analysis(X, y)
```

Рисунок 3.8 – Запис результатів роботи методу виділення ознак за критерієм Фішера до таблиці результатів results_df

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом критерію Фішера. Отриману матрицю наведено на рисунку 3.9. За отриманою матрицею можемо зробити висновок про якість класифікації. 92% та 97% ознак класів було визначено вірно. 8% доброякісних та 3% злоякісних було визначено невірно. Даний результат є непоганим, але все ще потребує покращення.

Перейдемо до створення програмної реалізації функції методу відбору ознак за критерієм Хі-квадрат. Під час створення програмної реалізації методу було враховано, що даний метод вимагає додатних значень, оскільки хі-квадрат статистика розраховується на основі частотних значень. У реалізації було використано MinMaxScaler, щоб масштабувати всі ознаки в діапазон [0, 1]. Крім того, було враховано, що метод підходить лише для дискретних даних, тому категоріальні ознаки були перетворені на числові за допомогою one-hot encoding. Код реалізації функції методу відбору ознак за критерієм Хі-квадрат наведено на рисунку 3.10.

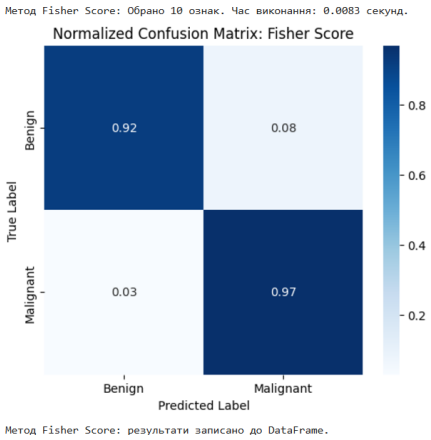


Рисунок 3.9 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом критерію Фішера

1.2 Chi-squared score (Метод відбору ознак за критерієм Хі-квадрат)

```python
def chi_squared_analysis_fixed(X, y):
    method_name = "Chi-squared score"
    feature_selection_group = "Filter_method"

    # === 1. Масштабування даних до невід'ємного діапазону ===
    scaler = MinMaxScaler()
    X_scaled = scaler.fit_transform(X)

    # === 2. Відбір ознак ===
    start_time = time.time()  # Початок вимірювання часу відбору
    selector = SelectKBest(score_func=chi2, k=10)
    X_selected = selector.fit_transform(X_scaled, y)
    feature_selection_speed = time.time() - start_time  # Час виконання методу відбору
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.10 – Код реалізації функції методу відбору ознак за критерієм Хі-квадрат

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом критерію Хі-квадрат. Отриману матрицю наведено на рисунку 3.11. За отриманою матрицею можемо зробити висновок про якість класифікації. 88% та 98% ознак класів було визначено вірно. 12% доброякісних та 2% злоякісних було визначено невірно. Даний результат потребує покращення.

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі кореляції між ознаками та цільовою змінною. Під час створення програмної реалізації методу було враховано, що даний метод включає етап обчислення кореляційної матриці, яка для великих наборів даних вимагає

значних обчислювальних ресурсів. Для оптимізації використовували методи лінійної алгебри з NumPy, а також було додано обробку пропущених значень. Код створення програмної реалізації функції методу відбору ознак на основі кореляції між ознаками та цільовою змінною наведено на рисунку 3.12.

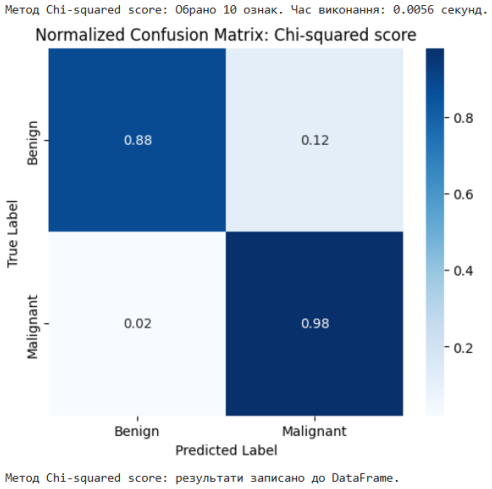


Рисунок 3.11 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом критерію Хі-квадрат

1.3 Correlation-based feature selection (CFS) (Метод відбору ознак на основі кореляції)

```python
def cfs_analysis(X, y):
    method_name = "CFS (Correlation-based Feature Selection)"
    feature_selection_group = "Filter_method"

    # === 1. Відбір ознак з використанням кореляції ===
    start_time = time.time()

    # Перетворюємо X у numpy-масив, якщо це DataFrame
    if isinstance(X, pd.DataFrame):
        X = X.values

    # 1.1. Перетворюємо цільовий вектор у числовий формат, якщо це потрібно
    if y.dtype == 'object' or y.dtype == 'str':
        y = LabelEncoder().fit_transform(y)

    # 1.2. Обчислення кореляції між ознаками та цільовою змінною
    cor_target = np.abs(np.corrcoef(X.T, y)[-1, :-1])  # Кореляція ознак з цільовою змінною

    # 1.3. Обчислення кореляції між ознаками
    cor_matrix = np.corrcoef(X, rowvar=False)
    mean_cor_feature = np.mean(np.abs(cor_matrix), axis=1)

    # 1.4. Оцінка CFS: співвідношення кореляції ознак до мети та між ознаками
    cfs_scores = cor_target / (mean_cor_feature + 1e-6)  # Додаємо невелике значення, щоб уникнути ділення на 0

    # 1.5. Вибір топ-10 ознак
    selected_indices = np.argsort(cfs_scores)[-10:]  # Обираємо топ-10 за CFS-оцінкою
    X_selected = X[:, selected_indices]  # Використовуємо numpy-індексацію

    feature_selection_speed = time.time() - start_time  # Час виконання методу відбору
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.12 – Код створення програмної реалізації функції методу відбору ознак на основі кореляції між ознаками та цільовою змінною

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом відбору ознак на основі кореляції між ознаками та цільовою змінною. Отриману матрицю наведено на рисунку 3.13. За отриманою матрицею можемо зробити висновок про якість класифікації. 85% та 96% ознак класів було визначено вірно. 15% доброякісних та 4% злоякісних було визначено невірно. Даний результат потребує покращення.

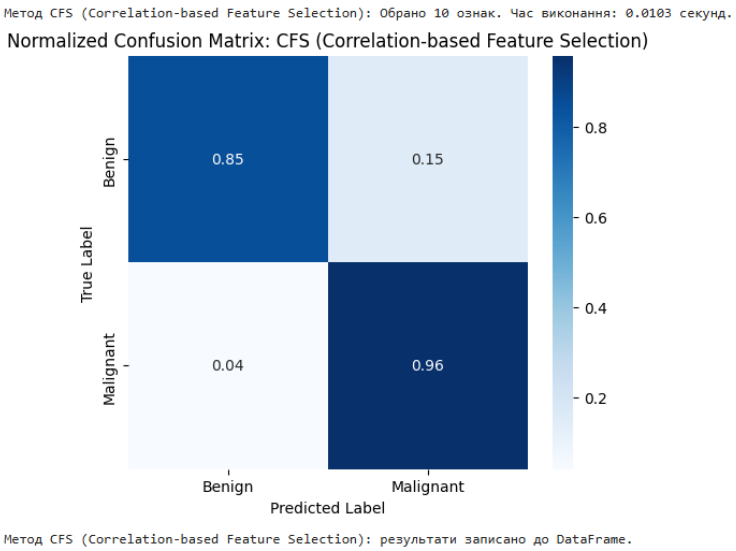


Рисунок 3.13 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом відбору ознак на основі кореляції

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі значення дисперсії. Під час створення програмної реалізації методу було враховано, що даний метод відбирає саме ті ознаки, значення дисперсії яких є найвищим. У реалізації було враховано, що метод відбору ознак на основі дисперсії працює лише із числовими даними, тому категоріальні ознаки були виключені або перетворені на числові. Додатково було налаштовано поріг дисперсії для визначення мінімальної інформативності ознаки. Код створення програмної реалізації функції методу відбору ознак на основі значення дисперсії на рисунку 3.14.

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом на основі значення дисперсії. Отриману матрицю наведено на рисунку 3.15. За отриманою матрицею можемо зробити висновок про якість класифікації. 85% та 96% ознак класів було визначено вірно. 15% доброякісних та 4% злоякісних було визначено невірно. Даний результат потребує покращення.

1.4 Variance Threshold (Метод відбору ознак на основі значення дисперсії)

```python
def variance_threshold_analysis(X, y):
    method_name = "Variance Threshold"
    feature_selection_group = "Filter_method"

    # === 1. Відбір ознак з використанням Variance Threshold ===
    start_time = time.time()

    # Ініціалізація та застосування VarianceThreshold
    selector = VarianceThreshold(threshold=0.01)  # Порогова дисперсія
    X_selected = selector.fit_transform(X)

    feature_selection_speed = time.time() - start_time  # Час виконання відбору
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.14 – Код створення програмної реалізації функції методу відбору ознак на основі значення дисперсії

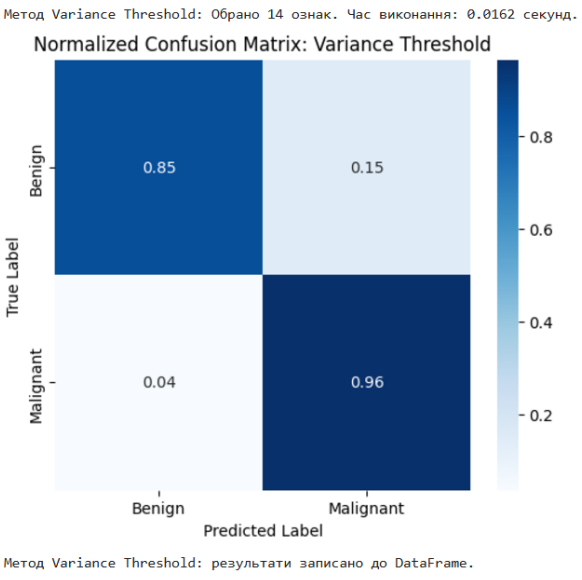


Рисунок 3.15 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом на основі дисперсії

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі значення середньої абсолютної різниці. Під час створення програмної реалізації методу було враховано, що даний метод виконує обчислення середньої абсолютної різниці значень кожної ознаки між різними класами даних. Створена програмна реалізація методу включає побудову окремих середніх значень для кожного класу набору даних. З метою уникнення впливу некоректних або пропущених значень, було додано етап попередньої перевірки та обробки даних. Код створення програмної реалізації функції методу відбору ознак на основі значення середньої абсолютної різниці наведено на рисунку 3.16.

1.5 Mean absolute difference (MAD) (Метод відбору ознак на основі значення середньої абсолютної різниці)

```
def mean_absolute_difference_analysis(X, y):
    method_name = "Mean Absolute Difference"
    feature_selection_group = "Filter_method"

    # === 1. Відбір ознак з використанням Mean Absolute Difference ===
    start_time = time.time()

    # Преобразуем X в NumPy-массив, если это не так
    if isinstance(X, pd.DataFrame):
        X = X.values

    # Розрахунок Mean Absolute Difference
    mad_scores = np.mean(np.abs(X - np.mean(X, axis=0)), axis=0)
    top_features_indices = np.argsort(mad_scores)[-10:]  # Обираємо 10 ознак з найбільшими значеннями MAD

    X_selected = X[:, top_features_indices]  # Вибір ознак за індексами

    feature_selection_speed = time.time() - start_time  # Час виконання відбору
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.16 – Код створення програмної реалізації функції методу відбору ознак на основі значення середньої абсолютної різниці

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом на основі значення середньої абсолютної різниці. Отриману матрицю наведено на рисунку 3.17. За отриманою матрицею можемо зробити висновок про якість класифікації. 85% та 96% ознак класів було визначено вірно. 15% доброякісних та 4% злоякісних було визначено невірно. Даний результат потребує покращення.

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі значення критерію Лапласа. Під час створення програмної

реалізації методу було враховано, що результати даного методу базуються на основі побудови графа подібності між об’єктами набору даних.

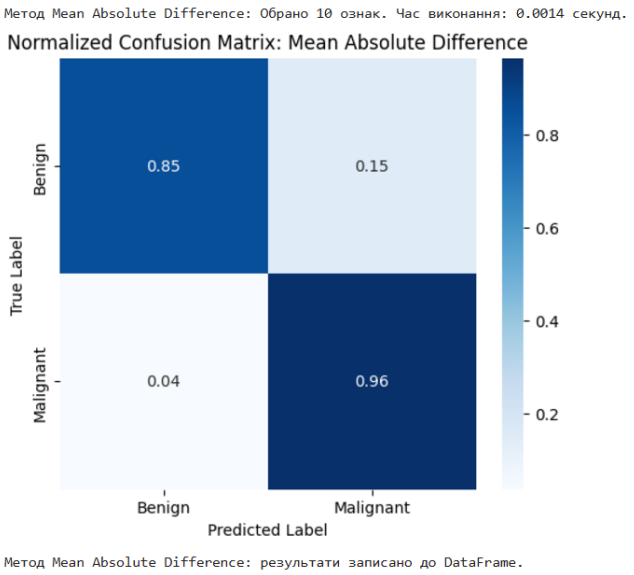


Рисунок 3.17 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом на основі значення середньої абсолютної різниці

У ході створення програмної реалізації методу відбору ознак на основі значення критерію Лапласа було використано функцію Гауса для ядра моделі, що дозволило мінімізувати вплив шуму даних на результати прогнозів моделі. Також було додано нормалізацію даних, для того, щоб мінімізувати вплив масштабу ознак на результати роботи. Код створення програмної реалізації функції методу відбору ознак на основі на основі значення критерію Лапласа наведено на рисунку 3.18.

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі значення критерію Лапласа. Отриману матрицю наведено на рисунку 3.19. За отриманою матрицею можемо зробити висновок про якість класифікації. 68% та 93% ознак класів було визначено вірно. 32% доброякісних та 7% злоякісних було визначено невірно.

Даний результат є найгіршим серед отриманих у ході дослідження, що говорить про ймовірну невідповідність методу поставленій задачі та даним.

1.6 Laplacian score (Метод відбору ознак за критерієм Лапласа)

```
def laplacian_score_analysis(X, y):
    method_name = "Laplacian Score"
    feature_selection_group = "Filter_method"

    # === 1. Відбір ознак з використанням Laplacian Score ===
    start_time = time.time()

    # Перетворюємо X на NumPy масив, якщо це DataFrame
    if isinstance(X, pd.DataFrame):
        X = X.values

    # Налаштування параметрів для Laplacian Score
    n_neighbors = 5  # Кількість сусідів
    nbrs = NearestNeighbors(n_neighbors=n_neighbors).fit(X)
    W = nbrs.kneighbors_graph(X, mode='connectivity')  # Матриця суміжності
    L = csgraph.laplacian(W, normed=True)  # Лапласіан графу
    D = np.diag(L.sum(axis=1).A.ravel())  # Ступені вершин графу

    # Обчислення Laplacian Score для кожної ознаки
    X_centered = X - X.mean(axis=0)
    scores = np.array([
        (X_centered[:, i].T @ L @ X_centered[:, i]) / (X_centered[:, i].T @ D @ X_centered[:, i])
        for i in range(X.shape[1])
    ])
    top_features_indices = np.argsort(scores)[:10]  # Обираємо 10 ознак із найнижчими значеннями Laplacian Score

    X_selected = X[:, top_features_indices]  # Вибрані ознаки
    feature_selection_speed = time.time() - start_time  # Час виконання відбору
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.18 – Код створення програмної реалізації функції методу відбору ознак на основі значення критерію Лапласа

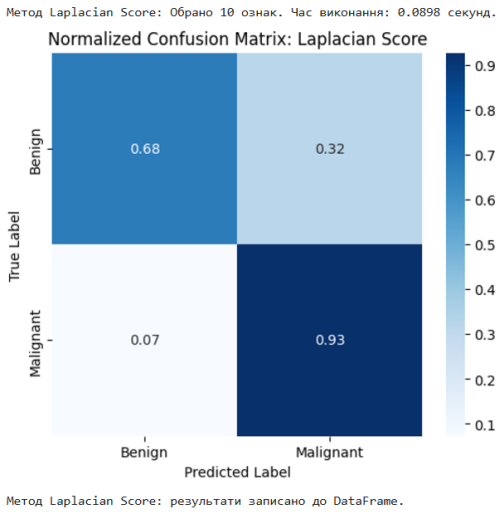


Рисунок 3.19 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі значення критерію Лапласа

Перейдемо до створення програмної реалізації функції мультикластерного методу відбору ознак. Під час створення програмної реалізації методу було

враховано, що для ефективної роботи даний метод потребує визначення кількості кластерів. Також, у ході створення мультикластерного методу відбору ознак було реалізовано етап оцінки оптимальної кількості кластерів. Код створення програмної реалізації функції мультикластерного методу відбору ознак наведено на рисунку 3.20.

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих у ході роботи функції мультикластерного методу відбору ознак. Отриману матрицю наведено на рисунку 3.21. За отриманою матрицею можемо зробити висновок про якість класифікації. 85% та 95% ознак класів було визначено вірно. 15% доброякісних та 4% злоякісних було визначено невірно. Даний результат потребує покращення.

1.7 Multi-Cluster Feature selection (MCFS) (Метод мультикластерного відбору ознак)

```
# pip install skfeature-chappers
```

```
def mcfs_analysis(X, y, n_clusters=2, n_selected_features=10):
    method_name = "MCFS (Multi-Cluster Feature Selection)"
    feature_selection_group = "Filter_method"

    # === 1. Відбір ознак за допомогою MCFS ===
    start_time = time.time()

    # Нормалізація даних
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Побудова графа суміжності
    affinity_matrix = kneighbors_graph(X_scaled, n_neighbors=5, mode='connectivity', include_self=True).toarray()

    # Спектральний аналіз графа
    _, eigenvectors = eigsh(affinity_matrix, k=n_clusters + 1, which='LM')
    spectral_features = eigenvectors[:, 1:n_clusters + 1]  # Исключаем первый вектор (нулевой собственный вектор)

    # Оцінка важливості графа
    feature_scores = np.sum(np.abs(np.dot(X_scaled.T, spectral_features)), axis=1)
    top_features_indices = np.argsort(feature_scores)[::-1][:min(n_selected_features, X.shape[1])]  # Учет размера X

    # Перевіряємо, що індекси коректні
    if isinstance(X, pd.DataFrame):
        X_selected = X.iloc[:, top_features_indices].values
    else:
        X_selected = X[:, top_features_indices]

    feature_selection_speed = time.time() - start_time  # Час виконання відбору
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.20 – Код створення програмної реалізації функції мультикластерного методу відбору ознак

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі методу із випадковим вибором примірників. Під час створення

програмної реалізації методу було враховано, що даний метод базується на оцінці відстаней між випадково взятими об’єктами для оцінки важливості ознак. У ході створення програмної реалізації даного методу було також реалізовано пошук найближчих сусідів за допомогою дерев, що суттєво пришвидшило роботу алгоритму. Також, для оцінки якості відбору було використано Евклідову відстань. Код створення програмної реалізації функції методу відбору ознак із випадковим вибором примірників на рисунку 3.22.

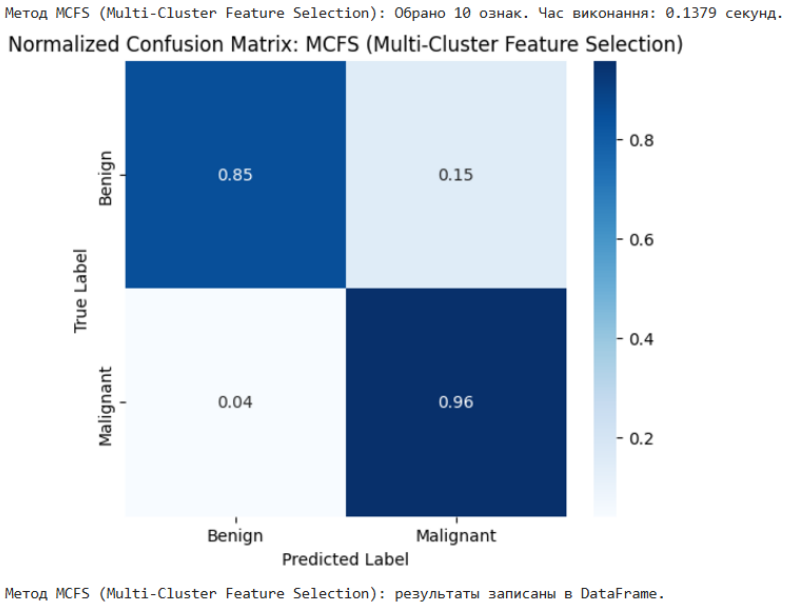


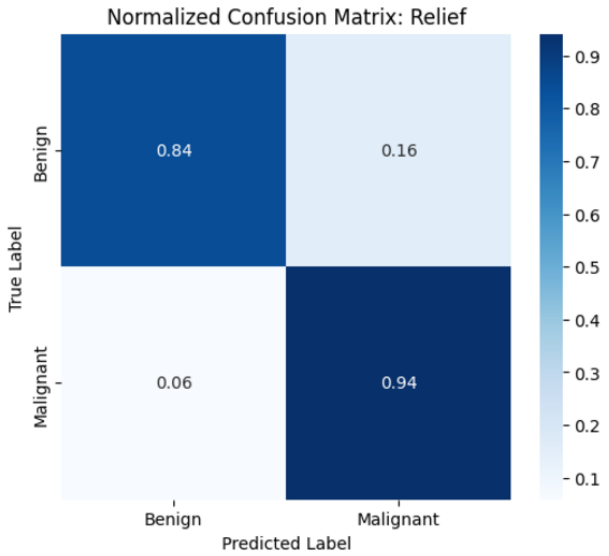
Рисунок 3.21 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих у ході роботи функції мультикластерного методу відбору ознак

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом із випадковим вибором примірників. Отриману матрицю наведено на рисунку 3.23. За отриманою матрицею можемо зробити висновок про якість класифікації. 84% та 94% ознак класів було визначено вірно. 16% доброякісних та 6% злоякісних було визначено невірно. Даний результат потребує покращення.

1.8 Relief (Метод відбору ознак із випадковим вибором примірників)

```python
def relief_analysis(X, y, n_selected_features=10):
    method_name = "Relief"
    feature_selection_group = "Filter_method"

    # === Попередня обробка вхідних даних ===
    # Перетворення в NumPy-масив, якщо це потрібно
    if not isinstance(X, np.ndarray):
        X = np.array(X)
    if not isinstance(y, np.ndarray):
        y = np.array(y)

    # Перевірка правильності розмірностей
    if len(X.shape) != 2:
        raise ValueError(f"Очікувано двовимірний масив для X, отримано розмірність {X.shape}")
    if len(y.shape) != 1:
        raise ValueError(f"Очікувано одномірний масив для y, отримано розмірність {y.shape}")

    # === 1. Відбір ознак за допомогою Relief ===
    start_time = time.time()

    # Обчислення ваг ознак за методом ReliefF
    scores = reliefF.reliefF(X, y)  # Отримання оцінок для кожної ознаки
    top_features_indices = np.argsort(scores)[::-1][:n_selected_features]  # Індекси топ-ознак

    # Перевірка на коректність отриманих індексів
    if np.max(top_features_indices) >= X.shape[1]:
        raise ValueError(f"Індекси ознак виходять за межі: {top_features_indices}")

    X_selected = X[:, top_features_indices]  # Обрані ознаки

    feature_selection_speed = time.time() - start_time  # Час виконання
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.22 – Код створення програмної реалізації функції методу відбору ознак із випадковим вибором примірників




```
Метод Relief: результати записано до DataFrame.
```

Рисунок 3.23 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом із випадковим вибором примірників

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі рекурсивного виключення ознак. Під час створення програмної

реалізації методу було враховано, що даний метод використовує принцип послідовного виключення ознак із найменшою важливістю із переліку. У ході створення програмної реалізації також було додано перевірку точності моделі на кожному кроці, щоб зупинити процес виключення ознак при досягненні оптимальної кількості ознак. Код створення програмної реалізації функції методу відбору ознак на основі рекурсивного виключення ознак наведено на рисунку 3.24.

2.1 Recursive Feature Elimination (RFE) (Метод рекурсивного виключення ознак)

```python
def rfe_analysis(X, y, n_selected_features=10):
    method_name = "Recursive Feature Elimination"
    feature_selection_group = "Wrapper_method"

    # === Попередня обробка вхідних даних ===
    # Перетворення X і y в NumPy, якщо це потрібно
    if not isinstance(X, np.ndarray):
        X = np.array(X)
    if not isinstance(y, np.ndarray):
        y = np.array(y)

    # Перевірка правильності розмірностей
    if len(X.shape) != 2:
        raise ValueError(f"Очікувано двовимірний масив для X, отримано розмірність {X.shape}")
    if len(y.shape) != 1:
        raise ValueError(f"Очікувано одномірний масив для y, отримано розмірність {y.shape}")

    # === 1. Відбір ознак за допомогою RFE ===
    start_time = time.time()

    # Використання Logistic Regression як базової моделі для RFE
    model = LogisticRegression(max_iter=5000, random_state=42, solver='saga')
    rfe = RFE(estimator=model, n_features_to_select=n_selected_features)
    rfe.fit(X, y)

    # Використання булевого масиву для відбору ознак
    X_selected = X[:, rfe.support_]

    feature_selection_speed = time.time() - start_time  # Час виконання
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.24 – Код створення програмної реалізації функції методу відбору ознак на основі рекурсивного виключення ознак

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом рекурсивного виключення ознак. Отриману матрицю наведено на рисунку 3.25. За отриманою матрицею можемо зробити висновок про якість класифікації. 85% та 96% ознак класів було визначено вірно. 15% доброякісних та 4% злоякісних було визначено невірно. Даний результат потребує покращення.

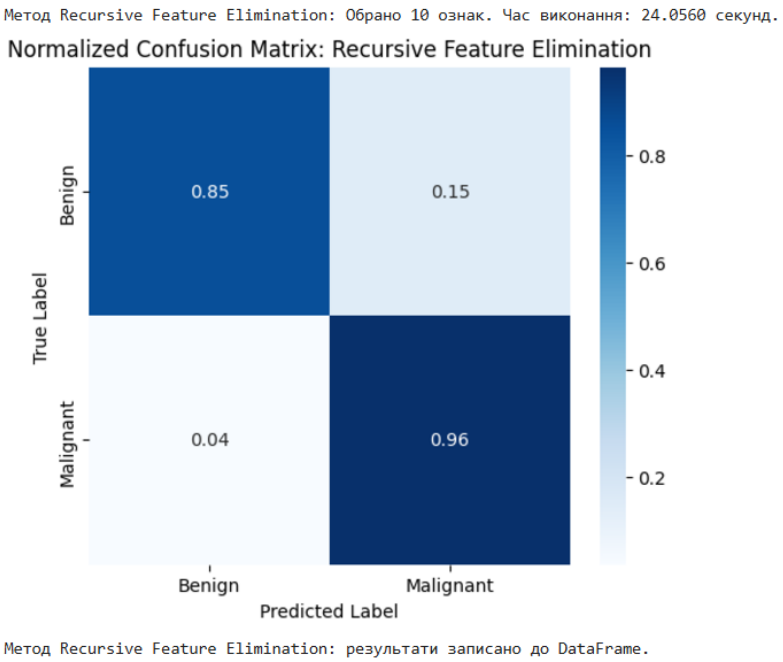


Рисунок 3.25 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих за методом на основі рекурсивного виключення ознак

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі розрідженої мультиноміальної логістичної регресії. Під час створення програмної реалізації методу було враховано, що даний метод базується на багатокласовій логістичній регресії з L_1-регуляризацією. У ході створення програмної реалізації також було додано алгоритм оптимізації, що враховує рідкість ваг, а також налаштування коефіцієнта регуляризації, щоб уникнути перенавчання і втрати важливих ознак. Код створення програмної реалізації функції методу відбору ознак на основі розрідженої мультиноміальної логістичної регресії наведено на рисунку 3.26.

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі розрідженої мультиноміальної логістичної регресії. Отриману матрицю наведено на рисунку 3.27. За отриманою матрицею можемо зробити висновок про якість класифікації.

93% та 97% ознак класів було визначено вірно. 7% доброякісних та 3% злоякісних було визначено невірно. Даний результат є задовільним.

3.1 Sparse Multinomial Logistic Regression (SMLR) (Метод розрідженої мультиноміальної логістичної регресії)

```python
def smlr_analysis(X, y):
    method_name = "Sparse Multinomial Logistic Regression"
    feature_selection_group = "Embedded_method"

    # === Попередня обробка вхідних даних ===
    # Перетворення X і y в NumPy, якщо це потрібно
    if not isinstance(X, np.ndarray):
        X = np.array(X)
    if not isinstance(y, np.ndarray):
        y = np.array(y)

    # Перевірка правильності розмірностей
    if len(X.shape) != 2:
        raise ValueError(f"Очікувано двовимірний масив для X, отримано розмірність {X.shape}")
    if len(y.shape) != 1:
        raise ValueError(f"Очікувано одномірний масив для y, отримано розмірність {y.shape}")

    # === 1. Відбір ознак за допомогою SMLR ===
    start_time = time.time()

    # Логістична регресія з L1-регуляризацією
    model = LogisticRegression(penalty='l1', solver='saga', max_iter=5000, random_state=42)
    model.fit(X, y)

    # Відбір ознак (ознаки з ненульовими вагами)
    selected_features = np.where(model.coef_[0] != 0)[0]
    X_selected = X[:, selected_features]

    feature_selection_speed = time.time() - start_time
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.26 – Код створення програмної реалізації функції методу відбору ознак на основі розрідженої мультиноміальної логістичної регресії

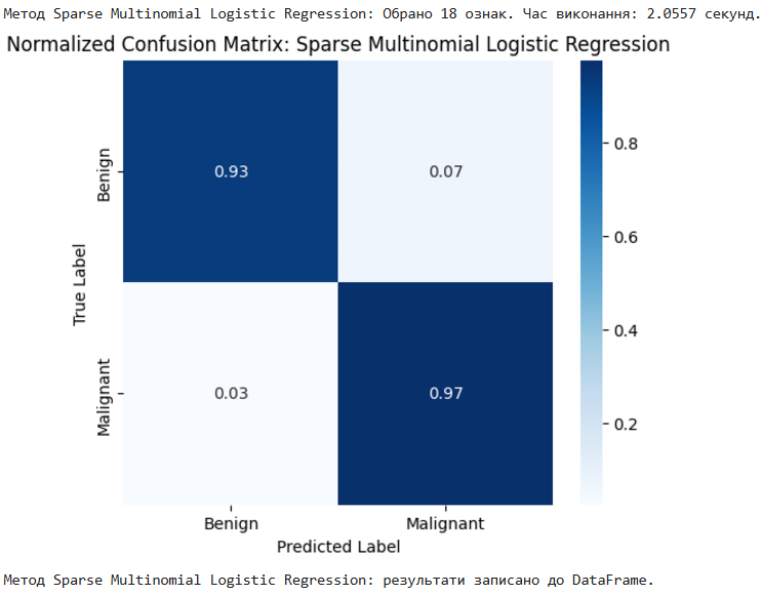


Рисунок 3.27 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі розрідженої мультиноміальної логістичної регресії

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі регресії автоматичного виділення релевантності. Під час створення програмної реалізації методу було враховано, що даний метод працює за допомогою обчислення ваг із апріорним байєсовим розподілом. Уході створення програмної реалізації була необхідна стандартизація ознак через високу чутливість до масштабу. Також було використано StandardScaler для забезпечення стабільності роботи моделі. Код створення програмної реалізації функції методу відбору ознак на основі регресії автоматичного виділення релевантності наведено на рисунку 3.28.

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі регресії автоматичного виділення релевантності. Отриману матрицю наведено на рисунку 3.29. За отриманою матрицею можемо зробити висновок про якість класифікації. 92% та 97% ознак класів було визначено вірно. 8% доброякісних та 3% злоякісних було визначено невірно. Даний результат є задовільним.

3.2 Automatic Relevance Determination (ARD) (Метод регресії автоматичного виділення релевантності)

```python
def ard_analysis(X, y):
    method_name = "Automatic Relevance Determination Regression"
    feature_selection_group = "Embedded_method"

    # === Попередня обробка вхідних даних ===
    # Перетворення X і y в NumPy, якщо це потрібно
    if not isinstance(X, np.ndarray):
        X = np.array(X)
    if not isinstance(y, np.ndarray):
        y = np.array(y)

    # Перевірка правильності розмірностей
    if len(X.shape) != 2:
        raise ValueError(f"Очікувано двовимірний масив для X, отримано розмірність {X.shape}")
    if len(y.shape) != 1:
        raise ValueError(f"Очікувано одномірний масив для y, отримано розмірність {y.shape}")

    # === 1. Відбір ознак за допомогою ARD ===
    start_time = time.time()

    # Модель ARD
    model = ARDRegression()
    model.fit(X, y)

    # Відбір ознак (ненульові коефіцієнти моделі)
    selected_features = np.where(np.abs(model.coef_) > 1e-5)[0]
    X_selected = X[:, selected_features]

    feature_selection_speed = time.time() - start_time
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.28 – Код створення програмної реалізації функції методу відбору ознак на основі регресії автоматичного виділення релевантності

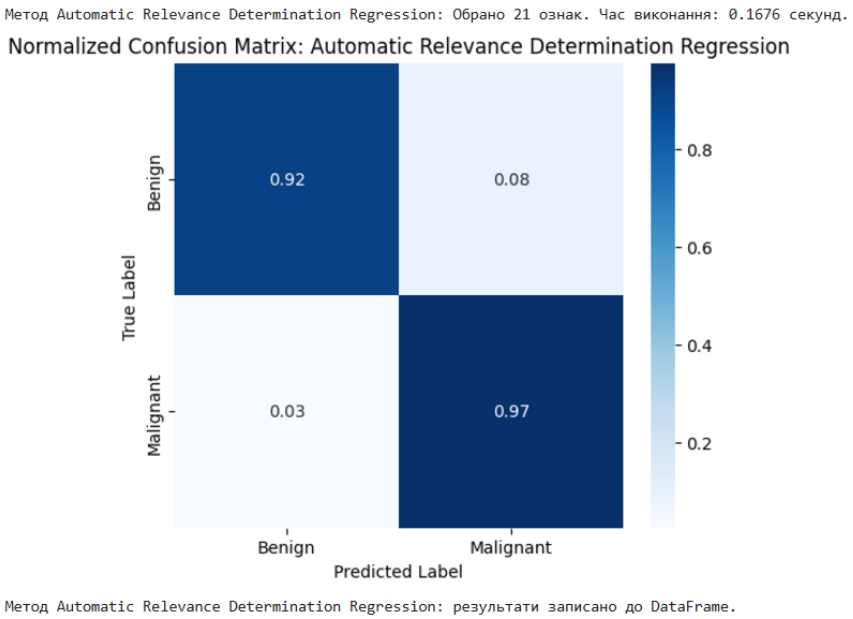


Рисунок 3.29 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі регресії автоматичного виділення релевантності

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі значення оператора найменшого абсолютного стиснення та відбору. Під час створення програмної реалізації методу було враховано, що даний метод автоматично обнуляє ваги малозначущих ознак завдяки L_1-регуляризації. У ході створення програмної реалізації також було встановлено параметр alpha, який визначає ступінь регуляризації. Також було додано перевірку точності, щоб уникнути втрати значущих ознак через надмірну регуляризацію. Код створення програмної реалізації функції методу відбору ознак на основі значення оператора найменшого абсолютного стиснення та відбору наведено на рисунку 3.30.

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі значення оператора найменшого абсолютного стиснення та відбору. Отриману матрицю наведено на рисунку 3.31. За отриманою матрицею можемо зробити висновок про якість

класифікації. 93% та 97% ознак класів було визначено вірно. 7% доброякісних та 3% злоякісних було визначено невірно. Даний результат є задовільним.

3.3 Least Absolute Shrinkage and Selection Operator (LASSO) (Метод відбору релевантних ознак за допомогою оператора найменшого абсолютного стиснення та відбору)

```python
def lasso_analysis(X, y):
    method_name = "Lasso Regression (L1 regularization)"
    feature_selection_group = "Embedded_method"

    # === Попередня обробка вхідних даних ===
    # Перетворення X і y в NumPy, якщо це потрібно
    if not isinstance(X, np.ndarray):
        X = np.array(X)
    if not isinstance(y, np.ndarray):
        y = np.array(y)

    # Перевірка правильності розмірностей
    if len(X.shape) != 2:
        raise ValueError(f"Очікувано двовимірний масив для X, отримано розмірність {X.shape}")
    if len(y.shape) != 1:
        raise ValueError(f"Очікувано одномірний масив для y, отримано розмірність {y.shape}")

    # === 1. Відбір ознак за допомогою Lasso ===
    start_time = time.time()

    # Модель Lasso
    lasso = Lasso(alpha=0.01, random_state=42, max_iter=10000)
    lasso.fit(X, y)

    # Відбір ознак (ненульові коефіцієнти моделі)
    selected_features = np.where(np.abs(lasso.coef_) > 1e-5)[0]
    X_selected = X[:, selected_features]

    feature_selection_speed = time.time() - start_time
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.30 – Код створення програмної реалізації функції методу відбору ознак на основі значення оператора найменшого абсолютного стиснення та відбору

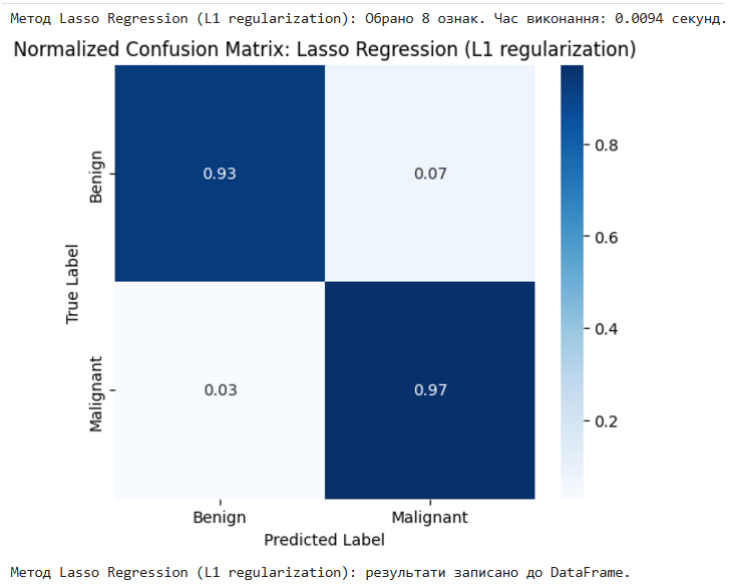


Рисунок 3.31 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі значення оператора найменшого абсолютного стиснення та відбору

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі гребневої регресії. Під час створення програмної реалізації методу було враховано, що даний метод усуває перенавчання через L_2-регуляризацію. У створеній програмній реалізації було використано стандартизацію ознак, оскільки дана модель є чутливою до масштабу. Крім того, було налаштовано значенення параметру alpha для знаходження балансу між регуляризацією і точністю. Код створення програмної реалізації функції методу відбору ознак на основі гребневої регресії наведено на рисунку 3.32.

Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі гребневої регресії. Отриману матрицю наведено на рисунку 3.33. За отриманою матрицею можемо зробити висновок про якість класифікації. 93% та 97% ознак класів було визначено вірно. 7% доброякісних та 3% злоякісних було визначено невірно. Даний результат є задовільним.

3.4 Ridge regression (Метод гребневої регресії)

```python
def ridge_analysis(X, y):
    method_name = "Ridge Regression (L2 regularization)"
    feature_selection_group = "Embedded_method"

    # === Попередня обробка вхідних даних ===
    # Перетворення X і y в NumPy, якщо це потрібно
    if not isinstance(X, np.ndarray):
        X = np.array(X)
    if not isinstance(y, np.ndarray):
        y = np.array(y)

    # Перевірка правильності розмірностей
    if len(X.shape) != 2:
        raise ValueError(f"Очікувано двовимірний масив для X, отримано розмірність {X.shape}")
    if len(y.shape) != 1:
        raise ValueError(f"Очікувано одномірний масив для y, отримано розмірність {y.shape}")

    # === 1. Відбір ознак за допомогою Ridge Regression ===
    start_time = time.time()

    # Модель Ridge
    ridge = Ridge(alpha=1.0, random_state=42, max_iter=10000)
    ridge.fit(X, y)

    # Відбір ознак (основі ваг коефіцієнтів)
    feature_importance = np.abs(ridge.coef_)
    selected_features = np.where(feature_importance > 1e-5)[0]
    X_selected = X[:, selected_features]

    feature_selection_speed = time.time() - start_time
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.32 – Код створення програмної реалізації функції методу відбору ознак на основі гребневої регресії

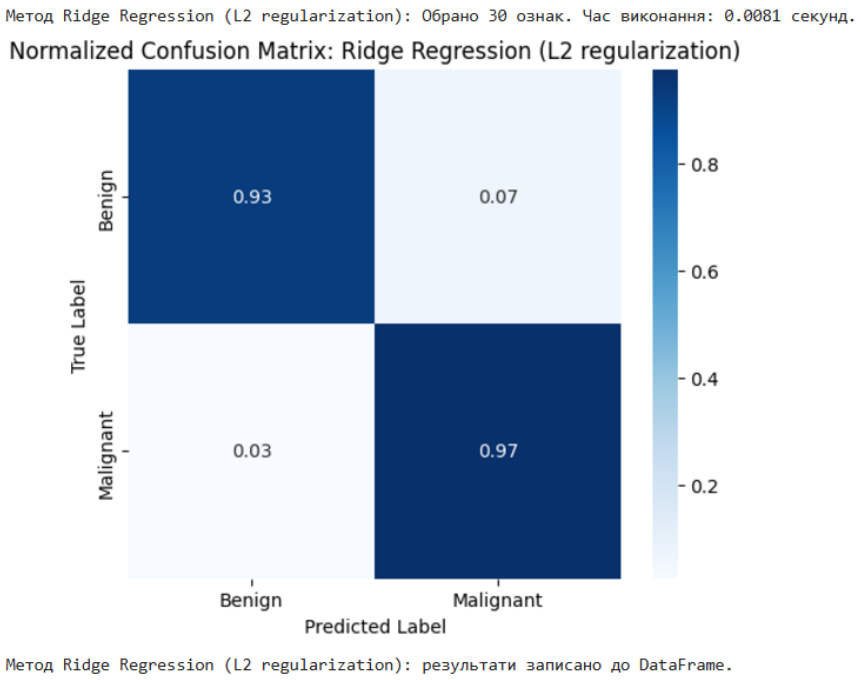


Рисунок 3.33 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі гребневої регресії

Перейдемо до створення програмної реалізації функції методу відбору ознак на основі еластичної сітки. Під час створення програмної реалізації методу було враховано, що даний метод реалізує відбір ознак із врахуванням корельованих змінних. У створеній програмній реалізації було налаштовано параметр l1_ratio, який контролює співвідношення між L_1 і L_2 -регуляризаціями, а також масштабувати дані для стабільної роботи алгоритму. Код створення програмної реалізації функції методу відбору ознак на основі еластичної сітки наведено на рисунку 3.34.

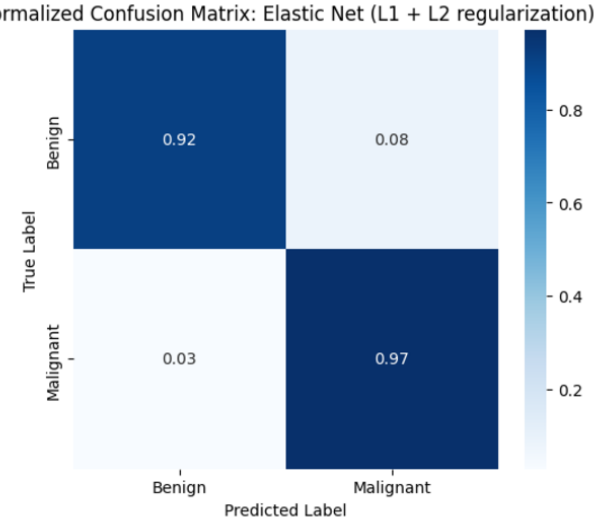
Наступним кроком будуємо матрицю помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі еластичної сітки. Отриману матрицю наведено на рисунку 3.35. За отриманою матрицею можемо зробити висновок про якість класифікації. 92% та 97% ознак класів було визначено вірно. 8% доброякісних та 3% злоякісних було визначено невірно. Даний результат є задовільним.

3.5 Elastic Net (Метод еластичної сітки)

```python
def elastic_net_analysis(X, y):
    method_name = "Elastic Net (L1 + L2 regularization)"
    feature_selection_group = "Embedded_method"

    # === Попередня обробка вхідних даних ===
    # Перетворення X і y в NumPy, якщо це потрібно
    if not isinstance(X, np.ndarray):
        X = np.array(X)
    if not isinstance(y, np.ndarray):
        y = np.array(y)

    # Перевірка правильності розмірностей
    if len(X.shape) != 2:
        raise ValueError(f"Очікувано двовимірний масив для X, отримано розмірність {X.shape}")
    if len(y.shape) != 1:
        raise ValueError(f"Очікувано одномірний масив для y, отримано розмірність {y.shape}")

    # === 1. Відбір ознак за допомогою Elastic Net ===
    start_time = time.time()

    # Модель Elastic Net
    elastic_net = ElasticNet(alpha=1.0, l1_ratio=0.5, random_state=42, max_iter=10000)
    elastic_net.fit(X, y)

    # Відбір ознак (основі ваг коефіцієнтів)
    feature_importance = np.abs(elastic_net.coef_)
    selected_features = np.where(feature_importance > 1e-5)[0]
    X_selected = X[:, selected_features]

    feature_selection_speed = time.time() - start_time
    print(f"Метод {method_name}: Обрано {X_selected.shape[1]} ознак. Час виконання: {feature_selection_speed:.4f} секунд.")
```

Рисунок 3.34 – Код створення програмної реалізації функції методу відбору ознак на основі еластичної сітки

```
Метод Elastic Net (L1 + L2 regularization): Обрано 3 ознак. Час виконання: 0.0167 секунд.
```




```
Метод Elastic Net (L1 + L2 regularization): результати записано до DataFrame.
```

Рисунок 3.35 – Матриця помилок для результатів класифікації даних за даними переліку ознак, отриманих на основі еластичної сітки

У ході створення програмних реалізацій виділених методів виділення ознак для класифікації та кластеризації, дані роботи визначених методів було записано

до структури Dataframe, повний вміст якого на рисунку 3.36. Також отримані дані було імпортовано до файлу .csv для подальшого аналізу.

Імпорт результатів для аналізу

```
results_df.head(15)
```

Feature_selection_group	Feature_selection_method	Feature_selection_speed	Classification_accuracy	Classification_precision	Classification_recall	Classification_F1_score	Cl
Filter_method	Fisher Score	0.015156	0.950791	0.953168	0.969188	0.961111	
Filter_method	Chi-squared score	0.005553	0.943761	0.933333	0.980392	0.956284	
Filter_method	CFS (Correlation-based Feature Selection)	0.010304	0.917399	0.914439	0.957983	0.935705	
Filter_method	Variance Threshold	0.016208	0.922671	0.917333	0.963585	0.939891	
Filter_method	Mean Absolute Difference	0.001410	0.922671	0.917333	0.963585	0.939891	
Filter_method	Laplacian Score	0.089839	0.834798	0.829574	0.927171	0.875661	
Filter_method	MCFS (Multi-Cluster Feature Selection)	0.137853	0.917399	0.914439	0.957983	0.935705	
Filter_method	Relief	0.890841	0.905097	0.910569	0.941176	0.925620	
Wrapper_method	Recursive Feature Elimination	24.055960	0.922671	0.917333	0.963585	0.939891	
Embedded_method	Sparse Multinomial Logistic Regression	2.055692	0.957821	0.958678	0.974790	0.966667	
Embedded_method	Automatic Relevance Determination Regression	0.167570	0.952548	0.950820	0.974790	0.962656	
Embedded_method	Lasso Regression (L1 regularization)	0.009407	0.956063	0.958564	0.971989	0.965229	
Embedded_method	Ridge Regression (L2 regularization)	0.008118	0.957821	0.958678	0.974790	0.966667	
Embedded_method	Elastic Net (L1 + L2 regularization)	0.016696	0.950791	0.950685	0.971989	0.961219	

```
results_df.to_csv('methods_results2.csv', index=False)
```

Рисунок 3.36 – Сформований набір даних про виконання відбору ознак та результати виконання класифікації та кластеризації на кожному наборі ознак

3.4 Аналіз та формування висновків за отриманими результатами створення програмної реалізації обраних методів відбору ознак

Аналіз отриманих результатів було вирішено провести у середовищі Salesforce Tableau Desktop. Дане середовище дозволяє створювати інформативні та лаконічні візуалізації. Завантажені дані до середовища, отримані у ході дослідження, наведено на рисунку 3.37.

methods_results2

Connection: Live / Extract

Filters: 0 | Add

methods_results2.csv — 12 fields 14 rows — 14 rows

Table Details

Feature selecti...	Feature selection method	Feature ...	Ac...	Precision	Recall	F1-sc...	ROC...	Classifi...
Filter method	Fisher Score	0.0152	0.9508	0.9532	0.9692	0.9611	0.9864	0.5028
Filter method	Chi-squared score	0.0056	0.9438	0.9333	0.9804	0.9563	0.9856	0.0163
Filter method	CFS (Correlation-based Feat...	0.0103	0.9174	0.9144	0.9580	0.9357	0.9682	0.7717
Filter method	Variance Threshold	0.0162	0.9227	0.9173	0.9636	0.9399	0.9707	0.9310
Filter method	Mean Absolute Difference	0.0014	0.9227	0.9173	0.9636	0.9399	0.9707	0.7231
Filter method	Laplacian Score	0.0898	0.8348	0.8296	0.9272	0.8757	0.9157	0.0210
Filter method	MCFS (Multi-Cluster Feature ...	0.1379	0.9174	0.9144	0.9580	0.9357	0.9674	0.4926
Filter method	Relief	0.8908	0.9051	0.9106	0.9412	0.9256	0.9581	1.0339
Wrapper method	Recursive Feature Elimination	24.0560	0.9227	0.9173	0.9636	0.9399	0.9707	0.6787
Embedded method	Sparse Multinomial Logistic ...	2.0557	0.9578	0.9587	0.9748	0.9667	0.9947	3.4534
Embedded method	Automatic Relevance Determ...	0.1676	0.9525	0.9508	0.9748	0.9627	0.9888	0.0758
Embedded method	Lasso Regression (L1 regulari...	0.0094	0.9561	0.9586	0.9720	0.9652	0.9934	0.1831

Рисунок 3.37 – Представлення вхідних даних результатів проведення дослідження у середовищі Salesforce Tableau Desktop

Наступним кроком виконаємо порівняння швидкості відбору ознак різними методами відбору ознак. Створену стовбчасту діаграму наведено на рисунку 3.38.

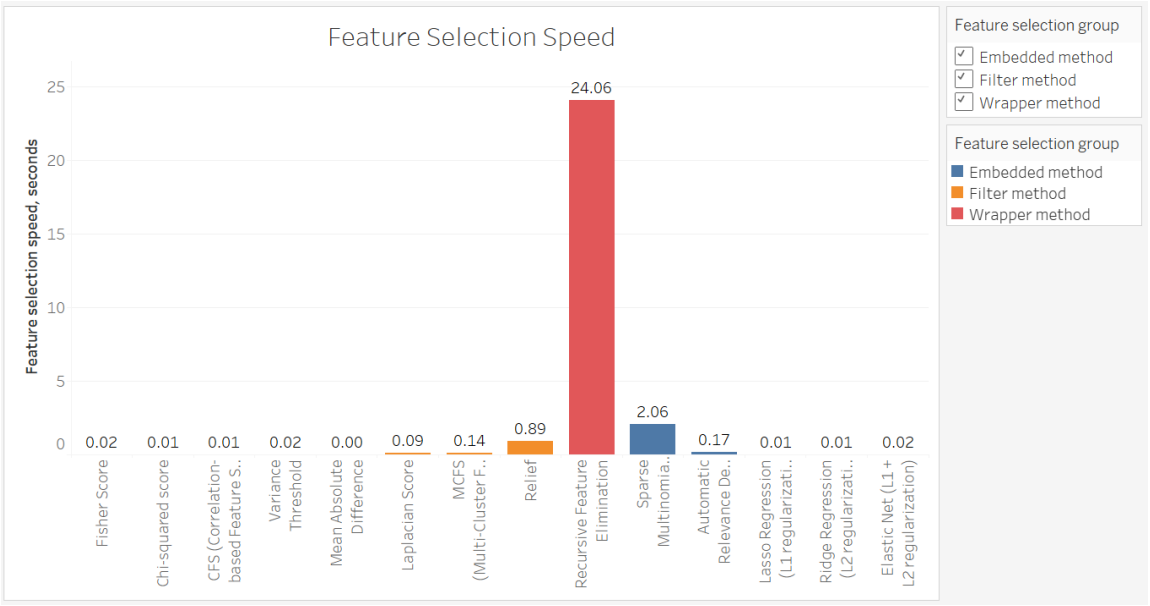


Рисунок 3.38 – Порівняння швидкості відбору ознак різними методами відбору ознак

Із отриманих результатів можна побачити, що найбільшу швидкодію мають фільтрові та вбудовані методи відбору ознак, натомість обгорткові методи демонструють найгірші показники швидкості роботи. Враховуючи, що дані для виконання аналізу є середніх розмірів, тож використання обгорткових методів не постає можливим для великих наборів даних.

Наступним кроком виконаємо порівняння швидкості класифікації на наборах ознак, відібраних різними методами відбору ознак. Створене порівняння за допомогою стовбчастої діаграми наведено на рисунку 3.39.

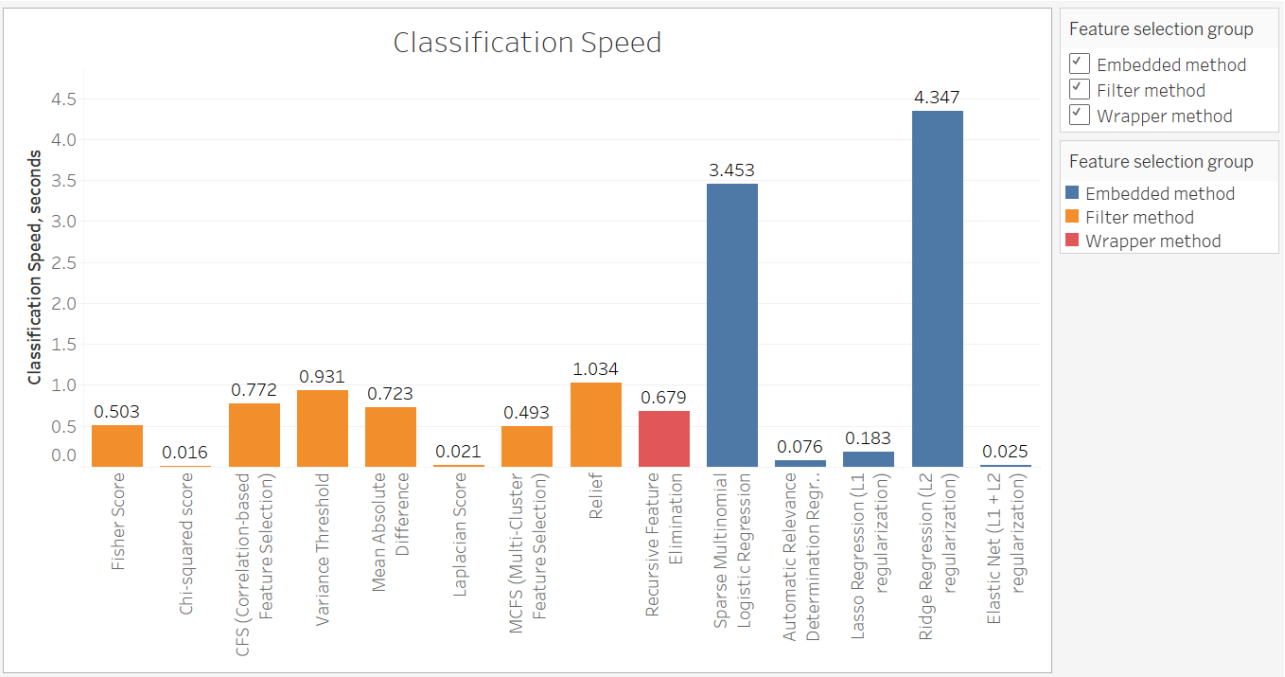


Рисунок 3.39 – Порівняння швидкості класифікації на обраному наборі ознак за допомогою різних методів відбору ознак

На рисунку 3.39 можна побачити, що найбільша швидкість класифікації спостерігається у двох фільтрових та трьох вбудованих методах. Також два вбудованих методи мають найбільші значення швидкості класифікації, 3 та 4 секунди, що не дозволяє використовувати дані методи для більших наборів даних.

Наступним кроком порівняємо швидкості кластеризації, представлені за допомогою стовбчастої діаграми на рис 3.40. Можна побачити, що усі значення швидкості є приблизно на одному рівні та не є високими. Але, все ж таки, кластеризація за допомогою вбудованих методів займає найменше часу.

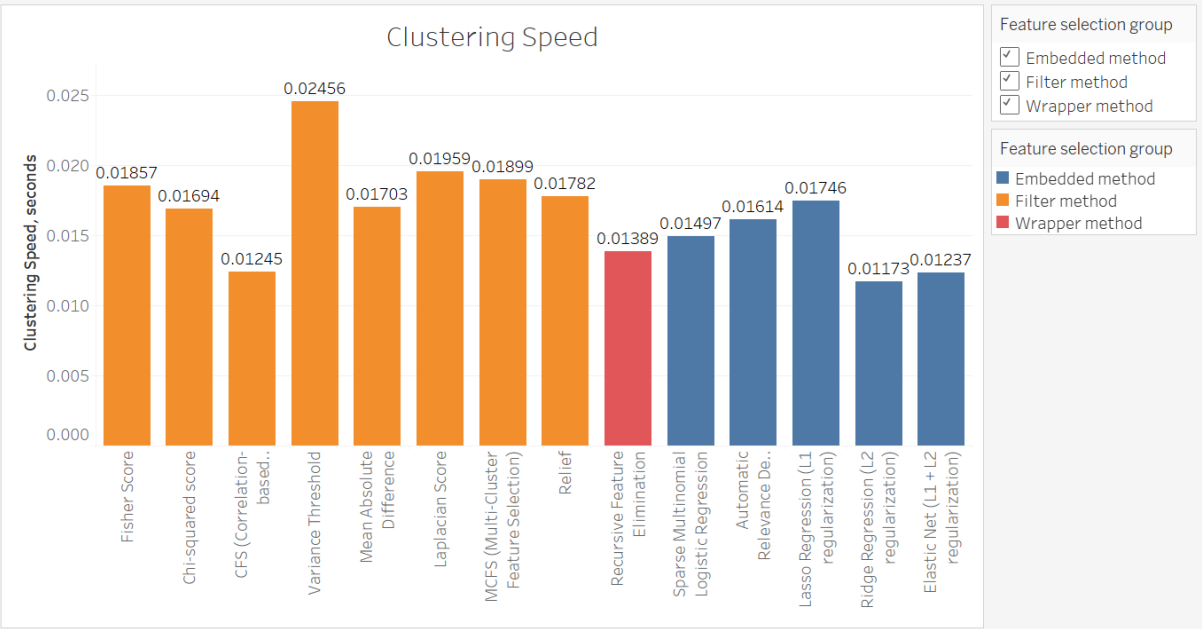


Рисунок 3.40 – Порівняння швидкості кластеризації на обраному наборі ознак за допомогою різних методів відбору ознак

Наступним кроком перейдемо до порівняння результатів отриманих метрик оцінки якості класифікації. Для цього було створено карту значень, як це представлено на рисунку 3.41.

Найкращі значення точності методів мають усі вбудовані методи та фільтровий метод відбору ознак за критерієм Фішера. Найгірші значення демонструє фільтровий метод критерію Лапласа та фільтровий метод із випадковим вибором примірників (Relief).

Найкращі значення точності отриманих прогнозів мають знову усі вбудовані методи та фільтровий метод відбору ознак за критерієм Фішера. Найгірші значення демонструє фільтровий метод критерію Лапласа.

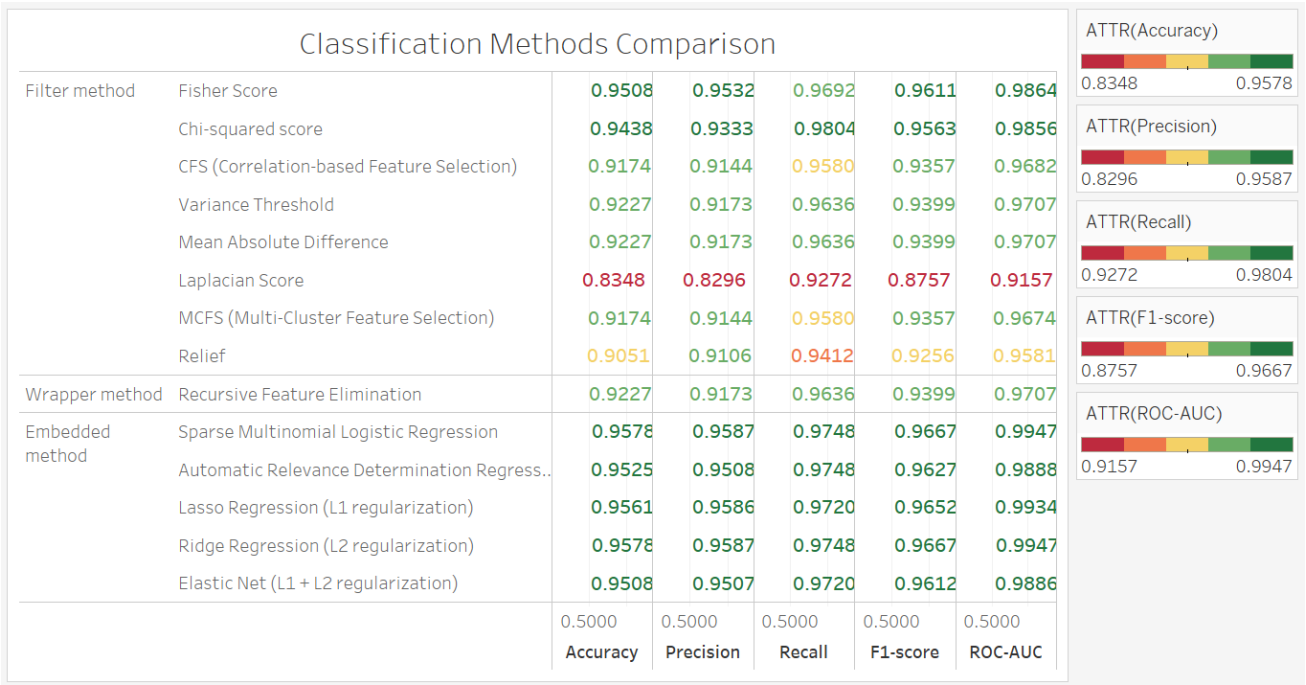


Рисунок 3.41 – Порівняння отриманих метрик класифікації за допомогою різних методів відбору ознак

Найкращі значення повноти мають усі вбудовані методи, фільтрові методи критеріїв Фішера та Хі-квадрат, а також обгортковий метод. Найгірші значення повноти мають фільтрові методи відбору ознак, а саме критерію Лапласа, на основі кореляції, мультикластерного відбору та із випадковим вибором примірників (Relief).

Найкращі значення метрики F1 мають усі вбудовані методи, а також фільтрові методи відбору ознак за критеріями Фішера та Хі-квадрат. Найгірші значення балансу між точністю прогнозу та повнотою мають фільтрові методи відбору ознак за критерієм Лапласа та із випадковим вибором примірників (Relief).

Найкращі значення метрики кривої помилок мають усі вбудовані методи та фільтрові методи відбору ознак за критеріями Фішера та Хі-квадрат. Найгірші значення мають фільтрові методи відбору ознак за критерієм Лапласа та із випадковим вибором примірників (Relief).

Наступним кроком перейдемо до порівняння результатів отриманих метрик оцінки якості кластеризації. Для цього було створено карту значень, як це представлено на рисунку 3.42.

За метрикою критерію силуету усі методи, окрім фільтрових методів Хі-квадрат, критерію Лапласа та методу із випадковим вибором примірників (Relief), демонструють приблизно однаковий результат.

Аналогічні результати можна побачити й для метрики Девіса-Булдіна.

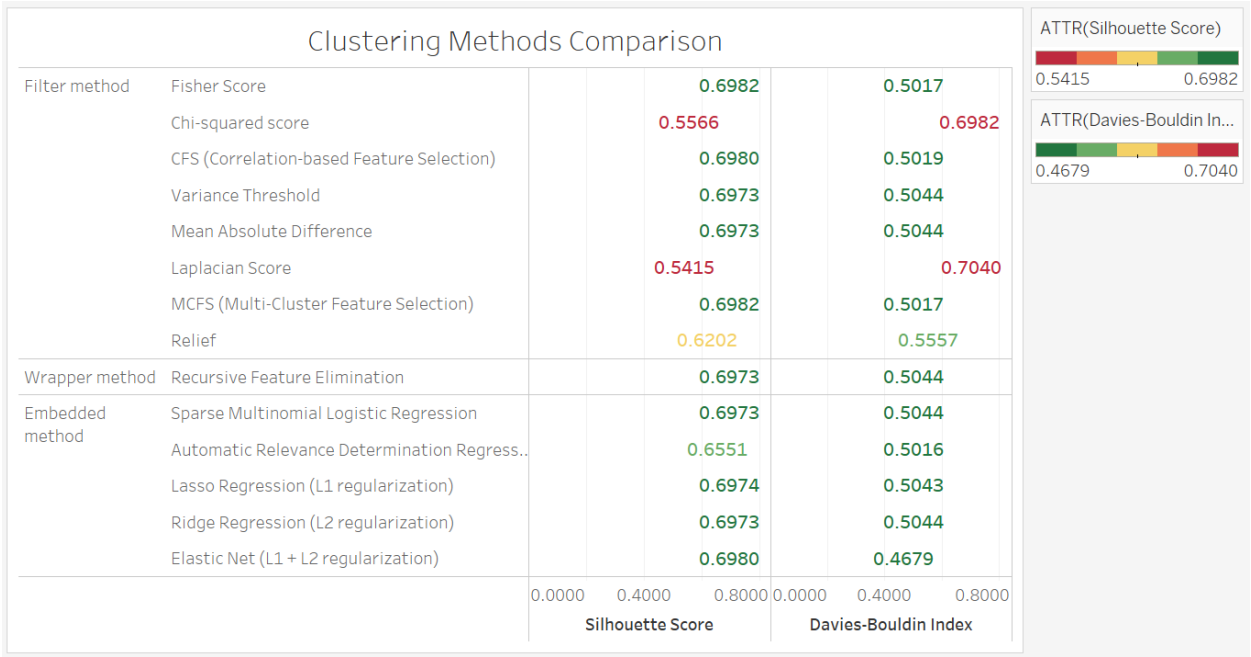


Рисунок 3.42і – Порівняння отриманих метрик кластеризації за допомогою різних методів відбору ознак

ВИСНОВКИ

У рамках кваліфікаційної роботи було створено програмну реалізацію алгоритму порівняння обраних методів відбору ознак для виконання задачі класифікації або кластеризації. Було визначено, що вибір найбільш суттєвих ознак є необхідним етапом при підготовці даних до аналізу. Порівняння засвідчило, що різні методи мають свої особливості та вплив на результати класифікації та кластеризації, а також було визначено, які методи демонструють себе як найбільш ефективні.

У ході виконання даної роботи першим кроком було проведено аналіз поточного становища та потреб аналізу даних, а також підходів до виділення релевантних ознак для задач класифікації та кластеризації.

Наступним кроком було проаналізовано та детально описано обрані традиційні методи відбору релевантних ознак для задач класифікації та кластеризації даних, а також було обрано перелік метрик для оцінки проведених класифікації та кластеризації та обраному алгоритмом переліку ознак. Для наочності отриманих результатів було створено порівняльну таблицю особливостей використання різних підходів до відбору ознак.

Останнім кроком було створено програмну реалізацію, яка дозволила виконати порівняння різних аспектів функціонування виділених методів відбору ознак та визначити ситуації застосування кожного із них та релевантність використання для поставленої задачі та вхідних даних.

Для вхідних даних медичного набору даних із середньою кількістю ознак було отримано наступні результати. Для задач класифікації у рамках поставленої задачі найбільш ефективними є вбудовані методи, а також фільтровий метод відбору ознак за критерієм Фішера та Хі-квадрат. Швидкість роботи даних методів відбору ознак є рівною, натомість швидкість виконання класифікації є

досить низькою для методів відбору ознак за допомогою розріженої мультиноміальної логістичної регресії та гребневої регресії, тому їх треба використовувати із обережністю для наборів даних із більшою кількістю ознак.

Для задач кластеризації найбільш релевантними методами відбору ознак було визначено, що усі методи, окрім фільтрових методів Хі-квадрат, критерію Лапласа та методу із випадковим вибором примірників (Relief), демонструють приблизно однаковий добрий результат. Швидкодія роботи кластеризації також є приблизно однаковою.

Таким чином, універсальними методами відбору ознак було визначено фільтровий метод відбору ознак за критерієм Фішера, а також вбудовані методи регресії автоматичного виділення релевантності, метод відбору релевантних ознак за допомогою оператора найменшого абсолютного стиснення та відбору, а також метод еластичної сітки.

На основі порівняння було створено рекомендації щодо вибору методів та застосування їх для аналізу даних залежно від специфіки задачі та об’єму вхідних даних, а також сформовано перспективи подальшої роботи.

Результати дослідження апробовано вигляді тез доповідей під час III Міжнародної науково-практичної конференції «Science and technology: challenges, prospects and innovations» [19].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. M. Jackson AI-Driven Data Modeling: Transforming Insights and Decision-Making: USA: Independently published, 2024, 78 p.

2. M. Sherry Machine Learning (ML) Guide: An Extensive Exploration of Key Concepts, Data Handling, Model Building, and Application Scenarios: USA: Independently published, 2024, 77 p.

3. Zheng, A. Casari Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists: USA: O'Reilly Media, 2020, 360 p.

4. M. Kuhn, K. Johnson Feature Engineering and Selection: A Practical Approach for Predictive Models: GB: Chapman and Hall/CRC, 2019, 314 p.

5. S. Appavu Feature Selection Methods Best Practices: Data mining Approach: Germany: LAP LAMBERT Academic Publishing, 2012, 64 p.

6. H. Liu, H. Motoda Computational Methods of Feature Selection: GB: Chapman and Hall/CRC, 2007, 440 p.

7. Why feature selection in clustering is important. URL: https://marufsazed.medium.com/why-feature-selection-in-clustering-is-important-de4c5c907c29 (дата звернення 08.10.2024).

8. Unsupervised feature selection via transformed auto-encoder. URL: https://www.sciencedirect.com/science/article/abs/pii/S0950705121000113 (дата звернення 08.10.2024).

9. Automatic feature engineering using Generative Adversarial Networks. URL: https://towardsdatascience.com/automatic-feature-engineering-using-generative-adversarial-networks-8e24b3c16bf3 (дата звернення 12.10.2024).

10. P. Bühlmann Statistics for High-Dimensional Data: Methods, Theory and Applications (Springer Series in Statistics): USA: Springer, 2011, 576 p.

11. Evolutionary Feature Selection for Machine Learning. URL: https://towardsdatascience.com/evolutionary-feature-selection-for-machine-learning-7f61af2a8c12 (дата звернення 12.10.2024).

12. Overview of feature selection methods. URL: https://towardsdatascience.com/overview-of-feature-selection-methods-a2d115c7a8f7 (дата звернення 31.09.2024).

13. Feature Selection: Filter Methods. URL: https://medium.com/analytics-vidhya/feature-selection-73bc12a9b39e (дата звернення 23.10.2024).

14. Understanding Wrapper Methods in Machine Learning: A Guide to Feature Selection. URL: https://arismuhandisin.medium.com/understanding-wrapper-methods-in-machine-learning-a-guide-to-feature-selection-23f71059abf8 (дата звернення 21.10.2024).

15. Filter vs Wrapper vs Embedded Methods For Feature Selection. URL: https://medium.com/@learnwithwhiteboard_digest/filter-vs-wrapper-vs-embedded-methods-for-feature-selection-8cc21e2174f7 (дата звернення 10.10.2024).

16. Feature Selection with Embedded Methods. URL: https://www.blog.trainindata.com/feature-selection-with-embedded-methods/ (дата звернення 10.10.2024).

17. Top 9 applications of classification in machine learning based on data type. URL: https://medium.com/@mohamadhasan.sarvandani/top-applications-of-classification-in-machine-learning-e7b4351f64eb (дата звернення 11.10.2024).

18. S. K. Subramanian Hybrid Methods in Feature Selection: A Data Classification Perspective: Hybrid Feature Selection Methods are the proven methods for Large Scale Feature Selection: Germany: Lap Lambert Academic Publishing, 2011, 64 p.

19. Машталір В.П., Сотникова А.В. Дослідження та порівняння методів відбору ознак для класифікації та кластеризації даних. *Наука та технології:*

завдання, перспективи та інновації: тези доповідей III міжнародної науково-практичної конференції (Осака, 1-3 листопада 2024р.). Осака, 2024. С. 192-196.

20. K. P. Bharani Chandra, Da-Wei Gu. Nonlinear Filtering: Methods and Applications: USA: Springer, 2020, 326 p.

21. Fisher Scores for Feature Selection. URL: https://genolearn.readthedocs.io/en/latest/background/fisher-score.html (дата звернення 01.12.2024).

22. Chi-Square Test for Feature Selection in Machine learning. URL: https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223 (дата звернення 01.12.2024).

23. Correlation-based Feature Selection in a Data Science Project. URL: https://medium.com/@sariq16/correlation-based-feature-selection-in-a-data-science-project-3ca08d2af5c6 (дата звернення 01.12.2024).

24. A Comprehensive Guide to Feature Selection using Variance Threshold in Scikit-Learn. URL: https://medium.com/aimonks/a-comprehensive-guide-to-feature-selection-using-variance-threshold-in-scikit-learn-0b10146aa71f (дата звернення 01.12.2024).

25. Mean Absolute Deviation: Definition, Finding & Formula. URL: https://statisticsbyjim.com/basics/mean-absolute-deviation/ (дата звернення 01.12.2024).

26. Laplacian Score: Gene Selection Methods for Microarray Data. URL: https://www.sciencedirect.com/topics/computer-science/laplacian-score (дата звернення 01.12.2024).

27. Efficient multi-cluster feature selection on text data. URL: https://www.tandfonline.com/doi/pdf/10.1080/02522667.2019.1703259 (дата звернення 01.12.2024).

28. Feature Selection with RReliefF (Regression). URL: https://www.kaggle.com/code/jorgesandoval/feature-selection-with-rrelieff-regression (дата звернення 01.12.2024).

29. Feature Selection with Wrapper Methods in Python. URL: https://www.blog.trainindata.com/feature-selection-with-wrapper-methods/ (дата звернення 01.12.2024).

30. Recursive Feature Elimination (RFE) for Feature Selection in Python. URL: https://machinelearningmastery.com/rfe-feature-selection-in-python/ (дата звернення 01.12.2024).

31. Feature Selection with Embedded Methods. URL: https://www.blog.trainindata.com/feature-selection-with-embedded-methods/ (дата звернення 01.12.2024).

32. L1 and L2 Regularization Methods, Explained. URL: https://builtin.com/data-science/l2-regularization (дата звернення 01.12.2024).

33. Multiclass sparse logistic regression on 20newgroups. URL: https://scikit-learn.org/1.5/auto_examples/linear_model/plot_sparse_logistic_regression_20newsgroups.html (дата звернення 01.12.2024).

34. Automatic Relevance Determination Regression: Unleashing the Power of Python for Enhanced Predictive Modeling. URL: https://medium.com/@danielwume/automatic-relevance-determination-regression-unleashing-the-power-of-python-for-enhanced-b63ebb9b37ca (дата звернення 01.12.2024).

35. Least Absolute Shrinkage and Selection Operator(LASSO Regression). URL: https://medium.com/@sidharths758/least-absolute-shrinkage-and-selection-operator-lasso-regression-9132dc80654b (дата звернення 01.12.2024).

36. What is ridge regression?. URL: https://www.ibm.com/topics/ridge-regression (дата звернення 01.12.2024).

37. Elastic Net Regression – Combined Features of L1 and L2 regularization. URL: https://medium.com/@abhishekjainindore24/elastic-net-regression-combined-features-of-l1-and-l2-regularization-6181a660c3a5 (дата звернення 01.12.2024).

38. Mastering Feature Selection: Key Applications and Differences – Part 1: Introduction. URL: https://medium.com/@aspnet_22/mastering-feature-selection-practical-applications-and-key-differences-explained-62da0bd063c0 (дата звернення 01.12.2024).

39. Scikit-learn toy datasets. URL: https://scikit-learn.org /1.5/datasets/toy_dataset.html#breast-cancer-dataset (дата звернення 20.11.2024).

40. What is logistic regression? URL: https://www.ibm.com/topics/logistic-regression (дата звернення 10.12.2024).

41. What is k-means clustering? URL: https://www.ibm.com/topics/k-means-clustering (дата звернення 10.12.2024).