

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ОЦІНЮВАННЯ ЗНАЧУЩОСТІ ОПИСОВИХ ХАРАКТЕРИСТИК ОБ'ЄКТІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
EVALUATING THE SIGNIFICANCE OF
DESCRIPTIVE CHARACTERISTICS OF OBJECTS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-123М
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ІВАНОВ Д.О.

(ПРІЗВИЩЕ та ініціали)

Керівник СТЕПАНЕНКО О.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ІВАНОВА Данила Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів оцінювання значущості описових характеристик об'єктів. Research and Software Implementation of Methods for Evaluating the Significance of Descriptive Characteristics of Objects

керівник проєкту (роботи) к.т.н., доцент, СТЕПАНЕНКО Олександр Олексійович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “29” листопада 2024 року № 498

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	СТЕПАНЕНКО О.О., доцент		
Нормоконтроль	КАЛІНІНА М.В., асистент		

7. Дата видачі завдання “09” вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

_____ Данило ІВАНОВ
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Олександр СТЕПАНЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
84 с., 3 табл., 25 рис., 3 дод., 25 джерел.

PYCHARM, PYTHON, ВІДБІР ОЗНАК, МОВА ПРОГРАМУВАННЯ,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СИНТЕЗ МОДЕЛЕЙ, ШТУЧНИЙ
ІНТЕЛЕКТ.

Об'єкт дослідження – процес синтезу розпізнавальних моделей.

Предмет дослідження – методи відбору інформативних ознак.

Мета роботи – дослідження та програмна реалізація методів
оцінювання значущості ознак.

Матеріали, методи та технічні засоби: мова програмування Python,
середовище розробки PyCharm.

Результати. Розроблено програмне забезпечення для оцінювання
інформативності ознак за допомогою мови програмування Python та
середовища розробки PyCharm.

Практична цінність роботи полягає у розробці програмного
забезпечення, що забезпечує оцінювання інформативності ознак.

Висновки. Виконано проєктування програмного забезпечення для
оцінювання інформативності ознак. Розроблено програмне забезпечення для
оцінювання інформативності ознак. Здійснено тестування розробленого
програмного забезпечення для оцінювання інформативності ознак.

Галузь використання – розпізнавання образів, моделювання.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 84 pages, 3 tables, 25 figures, 3 appendixes, 25 sources.

PYCHARM, PYTHON, FEATURE SELECTION, PROGRAMMING LANGUAGE, SOFTWARE, SYNTHESIS OF MODELS, ARTIFICIAL INTELLIGENCE.

The object of research is the process of synthesis of recognition models.

The subject of the research is methods of selecting informative features.

The purpose of this work is to research and to develop software for implementation of methods for evaluating the informativeness of features.

Materials, methods and technical tools: Python programming language, PyCharm development environment.

Results. The software tools for feature informativeness assessment using the Python programming language and the PyCharm development environment have been developed.

The practical value of the work lies in the development of software that provides evaluation of the informativeness of features.

Conclusions. The software for feature informativeness assessment has been designed. The software for feature informativeness assessment has been developed. The developed software for feature informativeness assessment has been tested.

Scope of use – project management.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Оцінювання інформативності ознак як етап процесу синтезу розпізнавальних моделей	11
1.2 Аналіз програмного забезпечення для синтезу розпізнавальних моделей	24
1.3 Висновки за розділом 1	33
2 Матеріали і методи	35
2.1 Вибір мови програмування	35
2.2 Вибір середовища розробки для створення програмного забезпечення для оцінювання інформативності ознак	37
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак	40
2.4 Висновки за розділом 2	42
3 Опис програми	43
3.1 Структура програмного забезпечення для підтримки процесу оцінювання інформативності ознак	43
3.2 Функціонування програмного забезпечення для оцінювання інформативності ознак	44
3.3 Використання методів оцінювання інформативності ознак у програмному забезпеченні для відбору ознак	45
3.4 Проєктування інтерфейсу програмного забезпечення для оцінювання інформативності ознак	49
3.5 Висновки за розділом 3	52
4 Експлуатація, тестування та експериментальне дослідження програми	53
4.1 Призначення й умови застосування програми	53
4.2 Характеристики програми для оцінювання інформативності ознак	54

4.3 Інструкція по експлуатації програми.....	55
4.3.1 Звернення до програми	55
4.3.2 Вхідні й вихідні дані.....	55
4.3.3 Повідомлення	55
4.4 Виконання програмного забезпечення для підтримки процесу оцінювання інформативності ознак	55
4.5 Тестування програмного забезпечення для підтримки процесу оцінювання інформативності ознак	58
4.6 Висновки за розділом 4	58
Висновки	59
Перелік джерел посилань	62
Додаток А Технічне завдання	65
Додаток Б Фрагмент тексту програми	69
Додаток В Слайди презентації	77

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

FS – Feature Selection;

БД – база даних;

ВО – відбір ознак;

ПЗ – програмне забезпечення;

РВО – рекурсивний відбір ознак;

ШІ – штучний інтелект.

ВСТУП

Оцінювання значущості (інформативності) ознак є важливим етапом у побудові моделей машинного навчання та аналізі даних. Цей процес допомагає визначити, які ознаки є найбільш значущими для прогнозування цільової змінної або результату [1]-[6].

Вибір найбільш інформативних ознак дозволяє зосередитися на релевантних даних, що може значно підвищити точність і ефективність моделі. Зменшення кількості ознак скорочує час навчання моделі та ресурси, необхідні для її обробки, що особливо важливо при роботі з великими наборами даних [1]-[6].

Інформативні ознаки допомагають уникнути ситуацій, коли модель надто добре запам'ятовує навчальні дані, що може призвести до погіршення її здатності до узагальнення на нові дані. Менша кількість важливих ознак полегшує інтерпретацію результатів моделі та допомагає краще зрозуміти, які фактори впливають на прийняття рішень [1]-[6].

Оцінювання інформативності дозволяє проводити оптимізацію моделі шляхом вибору найбільш релевантних ознак, що сприяє підвищенню її продуктивності та надійності [1]-[6].

Проте деяке програмне забезпечення для синтезу моделей може не підтримувати широкий спектр методів оцінки інформативності ознак, що може обмежувати можливості аналізу і вибору найбільш релевантних ознак. Іншим недоліком є те, що деякі програмні рішення можуть бути дорогими або вимагати значних обчислювальних ресурсів, що може бути недоступно для невеликих компаній або індивідуальних дослідників. Деяке програмне забезпечення може не завжди ефективно справлятися з незбалансованими наборами даних, де певні ознаки домінують над іншими, що може вплинути на точність оцінки інформативності [1]-[6].

Тому актуальною є розробка програмного забезпечення для оцінювання інформативності ознак. У дипломній кваліфікаційній роботі

магістра розв'язується актуальне завдання дослідження та програмної реалізації методів оцінювання інформативності ознак.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області стосовно оцінювання інформативності ознак;
- розробити модель взаємодії користувача з програмним забезпеченням оцінювання інформативності ознак;
- здійснити проєктування програмного забезпечення оцінювання інформативності ознак;
- створити програмне забезпечення для оцінювання інформативності ознак;
- виконати тестування розробленого програмного забезпечення для оцінювання інформативності ознак.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Оцінювання інформативності ознак як етап процесу синтезу розпізнавальних моделей

Модель є способом опису системи з метою розпізнавання, прогнозування і аналізу стану реальних об'єктів, процесів або явищ [1]-[7]. Моделі застосовуються для створення, моделювання і оцінки об'єктів у різних ситуаціях [1]-[7].

Часто проведення реальних експериментів для вивчення реальних проблем є дорогим або навіть неможливим. Тому моделі дозволяють передбачати поведінку і результати за допомогою математичних рівнянь та комп'ютерного моделювання [1]-[7].

Складність моделі не завжди гарантує покращення якості розпізнавання. Іноді простіші моделі можуть дати кращі результати. Зазвичай процес моделювання починається з простої моделі, яку потім ускладнюють за допомогою ітерацій, поки модель не почне адекватно описувати явище, яке потрібно вивчити [1]-[7].

Існують різні типи математичних моделей [1]-[7].

Детерміновані моделі прогнозують майбутнє на основі поточних даних і не враховують випадковості. Такі моделі часто використовуються в системах класифікації, що описують часовий розвиток [1]-[7].

Стохастичні моделі включають елементи випадковості та базуються на ймовірностях і випадкових процесах [1]-[7].

Моделі на основі даних шукають закономірності в спостережуваних даних для передбачення результатів системи. Такі моделі часто є функціями, значення яких розраховуються на основі спостережень [1]-[7].

Математичне моделювання включає спостереження реальних явищ і створення їх математичного опису. Для моделювання використовують системний підхід [1]-[7].

Важливим етапом в процесі моделювання є визначення змінних і параметрів. Необхідно визначити незалежні змінні, що входять до системи, і ті, що виходять з неї. Змінні можуть бути не завжди явно зазначені у формулюванні проблеми. Параметри описують зв'язки між змінними і повинні бути чітко ідентифіковані і позначені. На цьому етапі важливо оцінити інформативність кожної ознак з метою недопущення включення в модель неінформативних ознак або надлишкових ознак [1]-[7].

На наступному етапі моделювання відбувається встановлення меж і припущень. Припущення спрощують модель і допомагають пояснити взаємозв'язки між змінними. Наприклад, можна припустити, що сила тяжіння є сталою для об'єктів поблизу Землі. Обмеження визначають допустимі значення змінних та параметрів [1]-[7].

Коли проблема чітко сформульована, а змінні, параметри, припущення і обмеження визначені, слід вирішити, які математичні інструменти застосовувати для створення моделі залежності. Вибір може включати детерміновані, стохастичні моделі або моделі, що базуються на даних, або їх комбінацію [1]-[7].

Знайдене рішення (синтезовану модель) потрібно перевірити, щоб упевнитися, що воно дійсно корисне і відповідає поставленій проблемі. Потрібно оцінити, чи рішення відповідає запитам і чи є результат справедливим і обґрунтованим [1]-[7].

Детермінована модель завжди генерує той самий результат при однакових початкових умовах. Іншими словами, детерміновані моделі не включають елемент випадковості. Часто зосереджуються на детермінованих моделях, що описуються через системи диференціальних рівнянь. Для створення таких систем використовуються загальні фізичні закони або закони хімічних реакцій, або інші, зокрема, кліматичні, закони [1]-[7].

Системи диференціальних рівнянь, що виникають у цьому контексті, часто містять багато параметрів, як маса рухомого об'єкта чи теплоємність нагрітої речовини. Для спрощення моделей використовують процедуру

зменшення розмірності, що дозволяє масштабувати змінні і краще зрозуміти поведінку системи з огляду на ці параметри [1]-[7].

Оскільки більшість систем диференціальних рівнянь неможливо вирішити аналітично через явні елементарні функції, використовуються чисельні методи для наближеного розв'язання. Для обчислень та візуалізації чисельних рішень застосовуються різні програмні бібліотеки для моделювання та аналізу даних [1]-[7].

Стохастичні моделі використовуються для опису та аналізу систем, у яких є елементи випадковості і невизначеності. На відміну від детермінованих моделей, де результат чітко визначається на основі початкових умов і не змінюється, стохастичні моделі враховують ймовірнісні коливання і випадкові процеси [1]-[7].

Ключовими аспектами стохастичних моделей є такі. Стохастичні моделі інтегрують елементи випадковості у своїх розрахунках. Це означає, що результати можуть варіюватися навіть при однакових початкових умовах. Стохастичні моделі використовують ймовірнісні процеси для опису і передбачення поведінки систем. Це може включати випадкові варіації, такі як випадкові коливання або шум у даних [1]-[7]. Моделі Маркова використовують піхід, де ймовірність переходу між станами залежить лише від поточного стану, а не від попередніх. Моделювання реальних ситуацій через симуляцію великої кількості випадкових сценаріїв, щоб отримати статистичні оцінки результатів. Стохастичні моделі часто використовуються для прогнозування майбутніх подій, оцінки ризиків і вивчення варіацій у системах, де вплив випадкових факторів є значним [1]-[7].

Застосування моделювання може бути різним. Фінансові ринки: аналіз цін акцій, ризиків інвестицій та оптимізація портфелів. Медицина: оцінка ефективності лікування, прогнозування розвитку захворювань. Екологія: моделювання популяцій тварин, впливу кліматичних змін. Оскільки стохастичні моделі часто не мають аналітичних рішень, використовуються чисельні методи та симуляції для наближеного розрахунку результатів [1]-[7].

Загалом, стохастичні моделі дозволяють краще зрозуміти та передбачити поведінку систем, у яких випадковість і невизначеність грають важливу роль [1]-[7].

Моделі на основі даних являють собою підходи до побудови математичних моделей, які використовують спостереження і дані для виявлення закономірностей, тенденцій і взаємозв'язків. Такі моделі дозволяють робити прогнози та приймати рішення на основі історичних даних [1]-[7].

Моделювання починається зі збору даних з різних джерел, таких як бази даних, опитування, сенсори або інші джерела інформації. Дані обробляються, очищуються і підготовляються для аналізу. Для аналізу даних використовуються різноманітні методи статистичного аналізу для виявлення шаблонів, тенденцій та взаємозв'язків між змінними. Це може включати кореляційний аналіз, регресійний аналіз та інші статистичні методи. Після цього на основі проаналізованих даних створюється модель, яка може бути регресійною моделлю, класифікаційною моделлю (моделі, які призначають дані до певних категорій або класів), моделлю кластеризації даних (групує дані на класи або сегменти на основі їхніх характеристик [1]-[7].

Після побудови моделі її точність і надійність перевіряються за допомогою тестових даних. Цей процес допомагає упевнитися, що модель працює добре і є здатною до узагальнення на нові дані [1]-[7].

Далі на основі моделі здійснюються прогнози і приймаються рішення. Моделі на основі даних використовуються для різних цілей, таких як прогнозування продажів, оцінка ризиків, виявлення аномалій або оптимізація процесів [1]-[7].

Дані можуть змінюватися з часом, тому моделі на основі даних повинні бути адаптивними і періодично оновлюватися для відображення нових тенденцій та інформації [1]-[7].

Математична модель є способом описати систему за допомогою математичних термінів. Вона застосовується не лише в природничих і

технічних науках, але й у таких областях, як біологія, економіка і соціологія. Моделі можуть бути як простими, так і досить складними [1]-[7].

Моделювання дозволяє прогнозувати, контролювати та оптимізувати системи. Основна мета моделювання полягає у розбитті складних систем на їх ключові динамічні елементи та взаємодії. Ці елементи описуються математично, при цьому кожен компонент рівняння моделі відповідає певній функції (хоча це не завжди так). Визначення кожної функції в контексті загальної моделі дозволяє експериментувати з системою через обчислення, з'ясовувати, які динамічні процеси є найважливішими, і вивчати вплив змін на систему без необхідності проводити реальні експерименти [1]-[7].

Такі системи часто використовуються в біології, але методи побудови ефективних математичних моделей можна застосовувати й у інших сферах, таких як епідеміологія, фінанси, прогнозування погоди і системи управління, включаючи сільське господарство і охорону здоров'я. Математичне моделювання забезпечує глибоке розуміння того, як прогнозувати, контролювати та оптимізувати системи, що є ключовим для нашого взаємодії з природним світом [1]-[7].

Важливим етапом процесу моделювання є концептуальне моделювання, де відбувається розподіл складної системи на основні функції та взаємозв'язки. Інформація для цього отримується з наукових джерел і спеціально проведених спостережень чи експериментів, які допомагають заповнити прогалини у знаннях. На цьому етапі формується концептуальна модель, яка дозволяє створити діаграму для опису системи, але ще не забезпечує точних прогнозів [1]-[7].

Щоб здійснити обґрунтовані прогнози щодо системи, потрібно перетворити концептуальну модель на чітко визначену математичну модель. Це вимагає збору достатньої кількості даних для визначення математичних рівнянь, які потім можна перевіряти на основі нових даних [1]-[7]. При цьому важливо включити в модель інформативні ознаки, тобто ті, які суттєво впливають на вихідний параметр [1]-[7].

Дані, отримані за результатами моделювання, використовуються для двох основних цілей: для підвищення точності шляхом встановлення параметрів і для перевірки моделі через порівняння її результатів з реальними результатами [1]-[7]. Перевірка моделі включає порівняння результатів моделі з відповідними вимірюваннями реальної системи. Для оцінки точності моделі зазвичай потрібні кілька незалежних точок даних, які не використовувалися при налаштуванні моделі [1]-[7].

На цьому етапі також формується цикл вдосконалення і перевірки моделі, оскільки результати можуть спричинити коригування параметрів або включення нових динамічних елементів. Можливо, знадобляться додаткові експерименти для вивчення нових функцій або перевірки альтернативних моделей [1]-[7].

Важливо, щоб модель була настільки складною, наскільки це необхідно для точності, але водночас залишалася досить простою для забезпечення зрозумілого опису системи. Тому важливо включати в модель тільки ті ознаки, які безпосередньо пов'язані з фізичними властивостями системи [1]-[7].

Математичне моделювання полягає в перетворенні реальних проблем у математичні описи, зазвичай у вигляді рівнянь. Цей процес дозволяє не лише краще зрозуміти початкову проблему, але й виявити нові аспекти цієї проблеми. Моделювання є основою знання про світ і надає дослідникам можливість розробляти новітні технології [1]-[7].

Як зазначалося вище, важливим етапом синтезу моделей є оцінювання та відбір інформативних ознак [8]-[11].

Відбір ознак є процесом зменшення кількості вхідних даних при створенні прогностної або розпізнавальної моделі. Зменшення кількості вхідних змінних може допомогти знизити обчислювальні витрати на моделювання та іноді поліпшити ефективність моделі [8]-[11].

Статистичні методи відбору ознак передбачають оцінку зв'язків між вхідними змінними та цільовою змінною і вибір тих вхідних змінних, які мають найвищу кореляцію з цільовою змінною. Такі методи можуть бути

швидкими і ефективними, але вибір конкретного статистичного показника залежить від типу вхідних і вихідних змінних [8]-[11].

Зазвичай розрізняють два типи відбору ознак: контрольований і неконтрольований. Контрольовані методи включають техніки, такі як конвертація, фільтрація та внутрішній відбір. Методи фільтрації ознак використовують статистичні методи для оцінки кореляції або різниць між вхідними змінними, що дозволяє вибрати найбільш значущі ознаки [8]-[11].

Статистичні критерії для відбору ознак підбираються в залежності від природи вхідних змінних і чутливості виходу або змінних відповіді [8]-[11].

Методи відбору ознак допомагають зменшити кількість вхідних даних до найбільш важливих для моделі, які сприяють точному прогнозуванню цільової функції. Проблеми в прогнозних моделях можуть виникати через велику кількість вхідних ознак, що може сповільнити процес розробки та аналізу моделей і вимагати більше системної пам'яті. Крім того, деякі моделі можуть бути обмежені через конфлікти між вхідними та цільовими змінними. Наявність зайвих ознак може додати невизначеність у прогнози та знизити загальну продуктивність моделі [8]-[11].

Стратегії відбору ознак можна розділити на керовані і неконтрольовані. Керовані стратегії враховують, чи вибрані ознаки мають значення для цільових змінних. Наприклад, методи, які генерують нові змінні на основі кореляції. Неконтрольовані стратегії ігнорують цільові змінні і зосереджуються на загальних змінах [8]-[11].

Інші методи відбору ознак включають порогові методи та методи фільтрації. Ці методи слід постійно перевіряти і оцінювати за продуктивністю результуючої моделі на збереженому наборі даних [8]-[11].

Методи відбору ознак, що застосовують кілька моделей з різними наборами вхідних параметрів і функцій, дозволяє вибрати модель з найкращими показниками ефективності. Ці методи можуть бути затратними з обчислювальної точки зору, але ефективними у досягненні високої точності. Прикладом такого методу є рекурсивний відбір ознак (РВО) [8]-[11].

Методи відбору ознак на основі фільтрів використовують статистичні підходи для оцінки взаємозв'язку між кожною вхідною ознакою та цільовою змінною. На основі цих оцінок визначають, які вхідні змінні (фільтри) слід залишити для моделі [8]-[11].

Існують також алгоритми машинного навчання, які інтегрують відбір ознак у процес навчання моделі. Такі стратегії називають ендогенними методами відбору [8]-[11]. До таких алгоритмів відносяться моделі з регуляризацією, а також дерева розв'язків, включаючи методи на основі випадкових лісів рішень [8]-[11].

Відбір ознак має певні схожості з методами кількісного зменшення, оскільки обидва підходи допомагають зменшити кількість вхідних змінних у прогнозній моделі. Однак, відбір ознак спрямований на те, щоб вирішити, які змінні залишити або видалити з набору даних, тоді як методи кількісного зменшення створюють нову репрезентацію даних, що може привести до повністю нового введення. Отже, кількісне зменшення є альтернативою відбору ознак, а не його підвидом [8]-[11].

Проаналізуємо методи відбору ознак на основі фільтрів. Методи фільтрації для вибору ознак зазвичай базуються на статистичних мірах взаємозв'язку між вхідними та вихідними змінними [8]-[11].

Таким чином, вибір статистичних методів залежить від типу даних, що використовуються. Основні категорії змінних включають числові типи (наприклад, довжина) та категорії (наприклад, мітки), причому кожна з цих категорій може мати підкатегорії, такі як цілі числа для арифметичних змінних, арифметичні або логічні змінні, порядкові чи номінальні змінні або вхідні ознаки [8]-[11].

Типи даних для вхідних змінних можуть бути такими: числові змінні (цілі числа, змінні з плаваючою комою, категоріальні змінні (логічні змінні, порядкові змінні, номінальні змінні) [8]-[11].

Знання типу даних змінної допомагає вибрати відповідний статистичний критерій для методу фільтрації ознак [8]-[11].

Числові змінні – це змінні, що представляють числові значення. Категоріальні змінні – це змінні, що приймають певні категорії або класи. Вхідні змінні – це ті, які вводяться до моделі для подальшого аналізу. Під час вибору ознак ми прагнемо зменшити кількість вхідних змінних. Вихідні змінні – це змінні, які модель намагається прогнозувати, часто їх називають змінними відповіді [8]-[11].

Тип виходу моделі визначає вид задачі прогнозування. Числовий вихід відповідає задачі регресії. Категоріальний вихід відповідає задачі класифікації [8]-[11].

Статистичні вимірювання для фільтрації ознак зазвичай оцінюють зв'язок між кожною вхідною змінною і вихідною змінною окремо. Це називається однофакторним статистичним аналізом і може не враховувати взаємодію між вхідними змінними [8]-[11].

У випадку числового входу та числового виходу розв'язується задача регресії, можна застосовувати метод кореляції Пірсона для лінійної кореляції, метод рангового кореляції Спірмена для нелінійної кореляції [8]-[11].

У випадку числового входу та категоріального виходу розв'язується задача класифікації, можна застосовувати методи для оцінки зв'язків між числовими вхідними змінними та категоріальними виходами, зокрема, дисперсійний аналіз, ранговий коефіцієнт Кендалла [8]-[11]. Це один з найбільш поширених прикладів класифікаційних завдань. Зазвичай для таких завдань використовуються методи кореляції, але при цьому необхідно враховувати категоріальну природу виходу [8]-[11].

У випадку категоріального входу та числового виходу розв'язується задача регресійного моделювання. Це рідкісний тип регресійного завдання. Однак, в такому випадку можна застосувати ті ж методи, що й для випадку «числовий вхід, категоріальний вихід», але в зворотному порядку [8]-[11].

У випадку категоріального входу та категоріального виходу розв'язується завдання класифікації, де і вхідні, і вихідні змінні є категоріальними. Найчастіше використовуються такі методи для оцінки

зв'язку між категоріальними змінними: тест хі-квадрат для аналізу таблиць частот, інформаційний критерій з теорії інформації. Інформаційний критерій є потужним інструментом, який може бути корисним як для категоріальних, так і для числових даних, оскільки не залежить від типу даних [8]-[11].

Перетворення змінних може допомогти застосувати різні статистичні методи. Наприклад, можна перетворити категоріальну змінну на порядкову для спроби отримання нових результатів. Також можна дискретизувати числові змінні (наприклад, розділити їх на інтервали) і перевірити категоричні показники [8]-[11].

Деякі статистичні методи, такі як метод Пірсона, припускають певний розподіл даних (наприклад, нормальний розподіл) і лінійну залежність. Трансформація даних може допомогти відповідати цим припущенням і дозволити провести тест, а потім порівняти результати [8]-[11].

Не існує універсального методу для відбору ознак. Потрібно знайти найбільш ефективний підхід для конкретної задачі розпізнавання або регресії шляхом систематичних експериментів. Необхідно експериментувати з різними моделями, які працюють з різними підмножинами вхідних ознак, відбираючи їх за допомогою різних методів, і визначаючи, який підхід найкраще підходить для конкретної проблеми. Відповідно, для вирішення таких завдань необхідно розробити програмне забезпечення, що дозволяє виконувати оцінювання інформативності ознак за допомогою різних методів [8]-[11].

Відбір ознак є процесом відбору підмножини характеристик з наявних, з метою оптимізації простору ознак відповідно до визначених критеріїв. Існує три основних типи алгоритмів для відбору ознак: методи фільтрації, методи згортання і методи вкладення [8]-[11].

Роль відбору ознак у машинному навчанні полягає в наступному: зменшення кількості компонентів у даних, прискорення процесу навчання, підвищення точності прогнозів класифікаційного алгоритму, правильний вибір ознак є критично важливим етапом у розробці моделей машинного

навчання, оскільки він допомагає зменшити складність, покращити ефективність навчання і досягти кращих результатів у прогнозуванні [8]-[11].

Методи фільтрації не використовують явний процес побудови моделі. Замість цього, з даних відбираються невеликі підмножини, а ознаки оцінюються за їхньою важливістю для розрізнення різних типів даних. Методи згортання для відбору ознак можуть використовуватися алгоритми, такі як генетичні алгоритми та алгоритми імітації відпалу. Методи вбудовування включають оцінку розгалужень і меж для точного визначення важливості ознак, і часто починають з повного набору ознак, поступово скорочуючи набори ознак [8]-[11].

Коли розробляється модель машинного навчання для реальних даних, у вибірці даних, як правило, є велика кількість ознак, не всі з яких є суттєвими. Включення важливих ознак у модель допомагає підвищити її точність, зменшити складність і підвищити узагальнювальну здатність моделі. Тому відбір ознак є критично важливим етапом у розробці моделей машинного навчання, мета якого знайти найбільш значущі ознаки для побудови ефективної моделі [8]-[11].

Методи відбору ознак на основі фільтрування зазвичай застосовуються до початку процесу моделювання. Вони обирають ознаки з бази даних без залучення алгоритмів машинного навчання. Вони швидкі та економічні в плані обчислень, що робить їх ідеальними для відсіювання надлишкових чи непотрібних ознак. Проте, ці методи не завжди враховують взаємозв'язки між ознаками. Вони ефективні, коли ознаки є незалежними одна від одної, але можуть бути менш корисними, якщо потрібно врахувати комбінації ознак для підвищення загальної продуктивності моделі [8]-[11].

Зазвичай використовують такі підходи у методах відбору ознак на основі фільтрації. Приріст інформації – вимірює кількість інформації, яку ознака надає для передбачення цільової змінної, і оцінює зменшення ентропії. Приріст інформації розраховується для кожної ознаки з урахуванням цільових змінних [8]-[11]. Тест хі-квадрат часто застосовується для перевірки зв'язків

між категоріальними змінними. Він порівнює фактичні спостережувані значення з очікуваними значеннями, щоб оцінити ступінь зв'язку між змінними [8]-[11].

Метод Фішера автоматично відбирає ознаки, оцінюючи їх за критерієм Фішера, що допомагає створити набір факторів з найкращими характеристиками. Чим вищий показник Фішера, тим більш ефективна ознака. Коефіцієнт кореляції Пірсона вимірює силу та напрямок зв'язку між двома змінними в діапазоні від -1 до 1. Порогова дискримінація видаляє ознаки, варіація яких не відповідає певному пороговому значенню. Зазвичай він відкидає ознаки з нульовою дисперсією. Припускається, що ознаки з більшою варіацією мають більше інформації. Метод середньої абсолютної різниці подібний до методу порогового відхилення, але відмінний тим, що використовує абсолютні відхилення від середнього значення, а не квадратичні відхилення [8]-[11].

Метод кореляції вимірює взаємозв'язок між змінними та дає уявлення про те, яку інформацію одна змінна може надати щодо іншої. Він оцінює, наскільки змінна сприяє прогнозуванню цільової змінної [8]-[11].

Методи-огортки, або жадібні алгоритми, часто використовують для навчання моделі на основі набору об'єктів. В процесі навчання з результатів попередніх ітерацій додаються або видаляються елементи. Критерії вибору найкращих ознак зазвичай визначаються розробником моделі, залежно від продуктивності моделі або доступної кількості ознак. Головна перевага методів згортки над методами фільтрації полягає в тому, що вони дозволяють досягти кращих результатів для моделі, забезпечуючи вищу точність, хоча вони є більш витратними з точки зору обчислень [8]-[11].

До таких методів відносяться наступні. Метод прямого відбору ознак – ітераційний процес, при якому спочатку використовується порожній набір ознак, і з часом додаються нові елементи для покращення моделі після кожної ітерації. Процес зупиняється, якщо додавання нової ознаки не покращує продуктивність моделі [8]-[11].

Метод зворотного видалення ознак є також ітераційним методом, але починається з усіх можливих ознак і поетапно видаляє найменш корисні з них. Процес продовжується до тих пір, поки видалення ознак не призведе до покращення продуктивності моделі. Метод подвійного виключення – комбінує методи вибору ознак і зворотного видалення для досягнення унікального рішення. Метод повного перебору генерує всі можливі набори ознак, створюючи модель для кожного підмножини і вибираючи той набір, який забезпечує найкращу продуктивність моделі. Метод рекурсивного видалення – жадібний метод, який вибирає ознаки, постійно відсіюючи менш значущі ознаки. Оцінювання виконується для кожної ознаки, і менш важливі елементи видаляються з набору, поки не залишиться необхідна кількість ознак [8]-[11].

Методи вбудовування поєднують відбір ознак безпосередньо в процес навчання алгоритму, що дозволяє їм мати власні вбудовані методи відбору. Ці методи уникають обмежень фільтраційних і оболонкових методів, при цьому об'єднують їхні переваги. Вони швидші, як методи фільтрації, і точніші, ніж методи оболонки, оскільки враховують взаємодію між функціями [8]-[11].

До таких методів відносяться методи на основі дерев розв'язків, зокрема, випадковий ліс, які оцінюють важливість функцій для відбору ознак. Важливість або інформативність ознак показує, які з них найбільше впливають на цільову змінну [8]-[11].

Відзначимо, що окрім проаналізованих методів, існує велика кількість інших підходів до відбору ознак. Гібридні підходи можуть комбінувати переваги різних методів, зменшуючи їх недоліки і покращуючи точність та ефективність моделей. Методи зменшення розмірності, такі як аналіз головних компонент та евристичні алгоритми пошуку, також дозволяють оцінювати ознаки та можуть допомогти зменшити їх кількість [8]-[11].

Визначено, що модель є способом опису системи з метою розпізнавання, прогнозування і аналізу стану реальних об'єктів, процесів або явищ; висока складність моделі не завжди гарантує покращення якості розпізнавання, іноді простіші моделі можуть дати кращі результати.

Проаналізовано процес моделювання. Відзначено, що важливим етапом в процесі моделювання є визначення змінних і параметрів, на цьому етапі важливо оцінити інформативність кожної ознак з метою недопущення включення в модель неінформативних ознак або надлишкових ознак. Відбір ознак є процесом зменшення кількості вхідних даних при створенні прогнозної або розпізнавальної моделі, зменшення кількості вхідних змінних може допомогти знизити обчислювальні витрати на моделювання та іноді поліпшити ефективність моделі. Проаналізовано методи відбору ознак. Відзначено, що не існує універсального методу для відбору ознак, тому потрібно знайти найбільш ефективний підхід для конкретної задачі розпізнавання або регресії шляхом систематичних експериментів, необхідно експериментувати з різними моделями, які працюють з різними підмножинами вхідних ознак, відбираючи їх за допомогою різних методів, і визначаючи, який підхід найкраще підходить для конкретної проблеми. Відповідно, для вирішення таких завдань необхідно розробити програмне забезпечення, що дозволяє виконувати оцінювання інформативності ознак за допомогою різних методів.

1.2 Аналіз програмного забезпечення для синтезу розпізнавальних моделей

Проаналізуємо програмне забезпечення для синтезу розпізнавальних моделей [12]-[16].

Програмне забезпечення SuperAnnotate (рис. 1.1) є провідною платформою для створення, вдосконалення та управління моделями штучного інтелекту за допомогою високоякісних навчальних даних. Програмне забезпечення SuperAnnotate пропонує розширені інструменти для розпізнавання образів та контролю якості, управління даними, автоматизації процесів і інтеграції, що дозволяє швидко створювати набори даних і ефективні конвеєри машинного навчання [12]-[14].

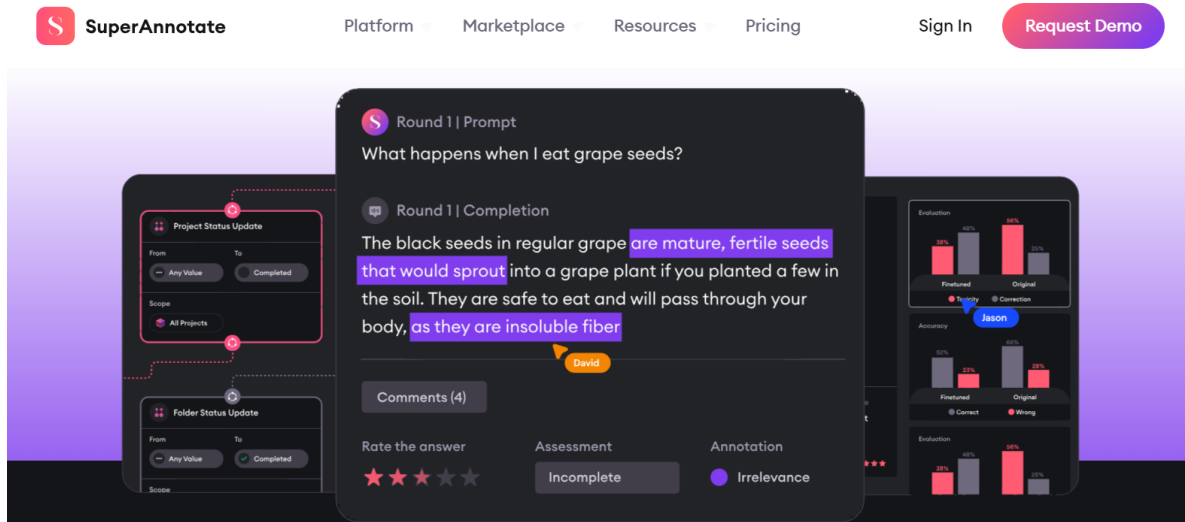


Рисунок 1.1 – Програмне забезпечення SuperAnnotate [12]

Програмне забезпечення SuperAnnotate підтримує роботи з професійно навченим персоналом, який забезпечує швидкий доступ до якісних даних для розробки найефективніших моделей [12]-[14]. Програмне забезпечення SuperAnnotate допомагає забезпечити точність і ефективність у створенні моделей штучного інтелекту, зменшуючи час і зусилля, необхідні для підготовки високоякісних даних [12]-[14].

Основними характеристиками програмного забезпечення SuperAnnotate є такі [12]-[14]:

- управління даними: програмне забезпечення SuperAnnotate надає функції для організації, зберігання та керування великими обсягами даних, що полегшують роботу з даними [12]-[14];

- висока якість попередньої обробки даних: програмне забезпечення SuperAnnotate надає точні та детальні інструменти для попередньої обробки даних, що забезпечує високу якість навчальних наборів даних [12]-[14];

- інструменти для анотації даних: програмне забезпечення SuperAnnotate включає в себе розширені інструменти для різних типів анотацій, таких як об'єктна детекція, сегментація, класифікація, розпізнавання тексту тощо [12]-[14];

– контроль якості даних: програмне забезпечення SuperAnnotate надає механізми для перевірки і покращення якості анотацій даних, що допомагає уникнути помилок і забезпечити точність даних [12]-[14];

– інтеграція: вбудовані засоби інтеграції з іншими інструментами та платформами для забезпечення безперервного робочого процесу і легкого імплементації в існуючі системи [12]-[14];

– співпраця з експертами: можливість працювати з професійно навченою командою анотацій, яка може допомогти у створенні та оптимізації наборів даних [12]-[14];

– масштабованість: програмне забезпечення SuperAnnotate дозволяє масштабувати процеси синтезу розпізнавальних моделей та обробки даних відповідно до потреб конкретного завдання [12]-[14];

– користувацькі інтерфейси: програмне забезпечення SuperAnnotate надає інтуїтивно зрозумілі інтерфейси для полегшення роботи користувачів і підвищення продуктивності [12]-[14];

– засоби аналізу та звітності: програмне забезпечення SuperAnnotate надає інструменти для моніторингу і аналізу якості даних і ефективності процесів, що дозволяє оптимізувати робочий процес [12]-[14].

Серед недоліків програмного забезпечення SuperAnnotate можна відзначити високу вартість підписки або ліцензії, що може бути перешкодою для малих компаній або індивідуальних користувачів. Крім того, новим користувачам може знадобитися час для освоєння всіх функцій та інструментів програмного забезпечення SuperAnnotate, що може зменшити швидкість впровадження. Іншим недоліком є те, що можуть виникнути проблеми з продуктивністю при обробці дуже великих обсягів даних або при високих навантаженнях на систему [12]-[14].

Програмне забезпечення Dataloop (рис. 1.2) представляє собою інноваційну платформу для розробки моделей на основі штучного інтелекту, яка спрощує процес створення ШІ-рішень у компаніях [12], [15].

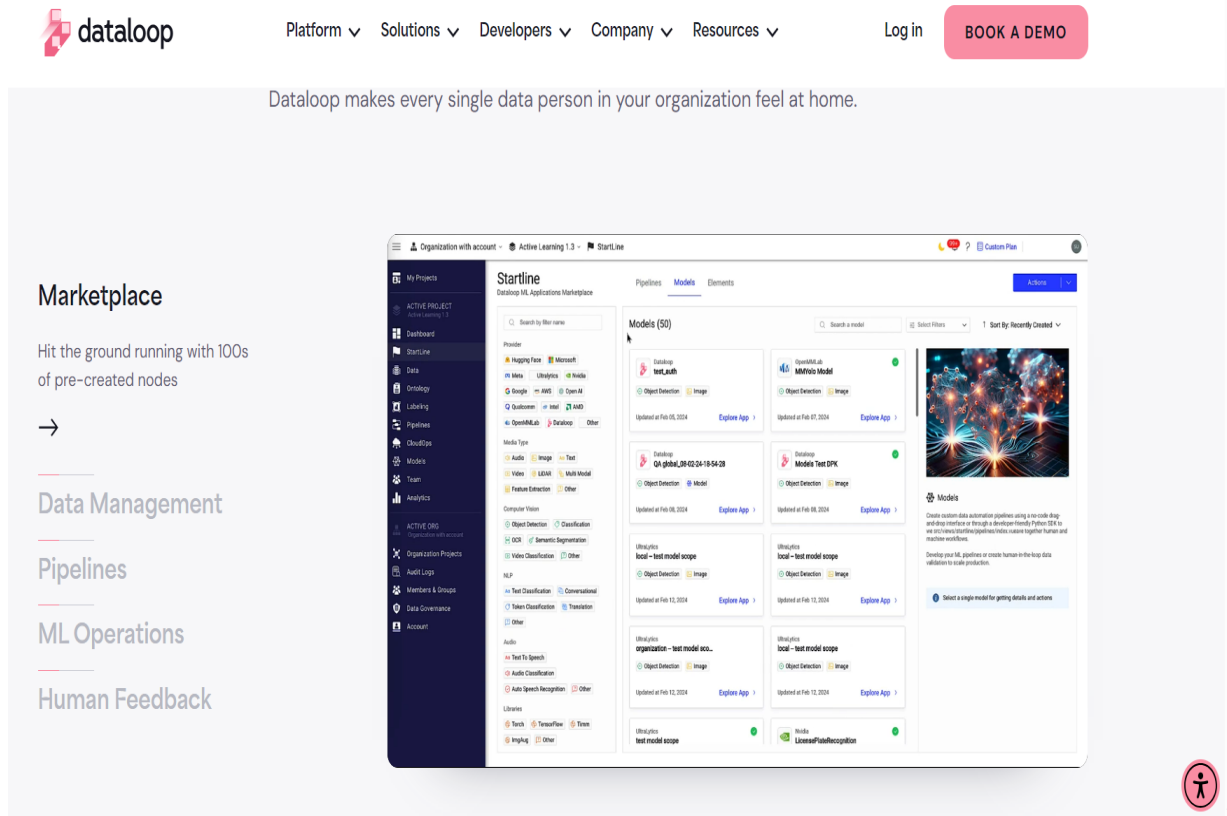


Рисунок 1.2 – Програмне забезпечення Dataloop [15]

Програмне забезпечення Dataloop ретельно розроблена, щоб забезпечити розробникам простоту і зручність при роботі з даними та моделями штучного інтелекту. Програмне забезпечення Dataloop є комплексним рішенням для всього циклу розробки моделей на основі ШІ, включаючи інструменти для управління даними, анотації, вибору моделі та розгортання. Dataloop акцентує увагу на співпраці, що дозволяє розробникам, дослідникам даних та інженерам ефективно працювати разом, подолати традиційні бар'єри і сприяти інноваціям [12], [15].

Ключові функції програмного забезпечення Dataloop включають інтуїтивно зрозумілий інтерфейс для створення конвеєрів даних, багатий вибір попередньо створених елементів і моделей ШІ, а також потужні можливості для контролю даних та анотацій. Ці інструменти допомагають швидко розгортати інтелектуальні програмні рішення, адаптуючи їх до швидко змінюваних вимог ринку [12], [15].

Програмне забезпечення Dataloop покращує процес розробки засобів штучного інтелекту, надаючи розробникам платформу, яка вирішує складнощі та виклики у сфері розробки інтелектуальних систем та управління складними даними [12], [15].

Програмне забезпечення Dataloop має такі особливості [12], [15]:

- засоби управління даними: програмне забезпечення Dataloop надає ефективні інструменти для збирання, обробки та організації даних, що включає в себе можливості для інтеграції з різними джерелами даних і управління великою кількістю інформації [12], [15];

- інтуїтивно зрозумілий інтерфейс: програмне забезпечення Dataloop має зручний графічний інтерфейс з можливістю перетягування, що полегшує створення конвеєрів даних і автоматизацію процесів [12], [15];

- анотація даних: програмне забезпечення Dataloop має розширені можливості для анотації даних, що дозволяє точно маркувати і позначати дані для навчання моделей штучного інтелекту [12], [15];

- можливість вибору моделі та методу навчання: програмне забезпечення Dataloop забезпечує інструменти для вибору і налаштування моделей, включаючи можливості для тренування моделей на основі різних алгоритмів машинного навчання [12], [15];

- розгортання моделей: програмне забезпечення Dataloop полегшує розгортання інтелектуальних рішень в продуктивному середовищі, забезпечуючи підтримку для впровадження та моніторингу моделей [12], [15];

- співпраця і комунікація: програмне забезпечення Dataloop сприяє командній роботі, дозволяючи розробникам, дослідникам та інженерам безперешкодно співпрацювати і обмінюватися результатами роботи [12], [15];

- бібліотека інструментів штучного інтелекту: програмне забезпечення Dataloop містить велику кількість попередньо створених елементів і моделей штучного інтелекту, що спрощує процес створення та модифікації інтелектуальних рішень [12], [15];

- масштабованість: програмне забезпечення Dataloop здатне обробляти великі обсяги даних і працювати з масштабованими моделями [12], [15];
- гнучкість: програмне забезпечення Dataloop підходить для різних типів інтелектуальних проєктів, включаючи комп'ютерне бачення, обробку природної мови та інші [12], [15];
- безпека: програмне забезпечення Dataloop підтримує високий рівень захисту даних і відповідність стандартам безпеки [12], [15];
- інтеграція: програмне забезпечення Dataloop має можливість інтеграції з іншими інструментами та платформами, що полегшує впровадження в існуючі робочі процеси [12], [15];
- аналіз і звітність: програмне забезпечення Dataloop пропонує інструменти для аналізу результатів, моніторингу продуктивності і створення звітів [12], [15].

Серед недоліків програмного забезпечення Dataloop можна відзначити високу вартість, що може бути перешкодою для малих компаній або індивідуальних користувачів. Крім того, новим користувачам може знадобитися час для освоєння всіх функцій та інструментів програмного забезпечення Dataloop, що може зменшити швидкість впровадження та зручність використання. Іншим недоліком є те, що можуть виникнути проблеми з продуктивністю при обробці дуже великих обсягів даних або при високих навантаженнях на систему [12], [15].

Програмне забезпечення Neural Designer (рис. 1.3) є потужною платформою для машинного навчання, яка дозволяє створювати моделі без написання коду, використовуючи інструменти перетягування та клацання. Платформа відрізняється високою швидкістю навчання, надаючи можливість обробки великих даних з використанням графічного процесору. Хоча Neural Designer написано на C++, що може ускладнити використання, це забезпечує значний приріст швидкості [13], [16].



Рисунок 1.3 – Програмне забезпечення Neural Designer [16]

Програмне забезпечення Neural Designer має такі особливості [13], [16]:

- висока продуктивність: програмне забезпечення Neural Designer забезпечує високу швидкість навчання завдяки оптимізації під графічні процесори [13], [16];

- зручний інтерфейс: програмне забезпечення Neural Designer дозволяє створювати та налаштовувати моделі машинного навчання без необхідності писати код. Використовується інтуїтивно зрозумілий інтерфейс перетягування та клацання для побудови діаграм блоків [13], [16];

- використання мови C++: платформа повністю реалізована на мові C++, що дозволяє досягти високої продуктивності, але може ускладнити деякі аспекти використання для кінцевих користувачів [13], [16];

- гнучкість у налаштуваннях моделей: програмне забезпечення Neural Designer підтримує широкий спектр алгоритмів машинного навчання, включаючи нейронні мережі, методи підтримки векторних машин, дерева рішень та інші [13], [16];

- засоби аналізу даних: програмне забезпечення Neural Designer пропонує потужні інструменти для обробки та аналізу даних, включаючи візуалізацію результатів навчання та оцінки моделей [13], [16];

– автоматизація процесів: програмне забезпечення Neural Designer включає функції для автоматизації рутинних завдань, таких як налаштування параметрів і вибір найбільш підходящих моделей [13], [16];

– підтримка обробки великих даних: програмне забезпечення Neural Designer ефективно обробляє великі обсяги даних, що дозволяє працювати з масштабованими проєктами машинного навчання [13], [16];

– інтеграція з іншими інструментами: програмне забезпечення Neural Designer може бути інтегрований з різними зовнішніми інструментами та платформами для полегшення роботи в багат шарових середовищах [13], [16].

Серед недоліків програмного забезпечення Neural Designer можна відзначити високу вартість, що може бути перешкодою для малих компаній або індивідуальних користувачів. Крім того, новим користувачам може знадобитися час для освоєння всіх функцій та інструментів програмного забезпечення Neural Designer, що може зменшити швидкість впровадження та зручність використання [13], [16].

Порівняльну характеристику програмного забезпечення для підтримки процесу синтезу розпізнавальних моделей, зокрема, для оцінювання інформативності ознак, наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для синтезу розпізнавальних моделей

Критерій порівняння	SuperAnnotate [14]	Dataloop [15]	Neural Designer [16]
Масштабованість	+	+-	+
Робота з великими даними	+-	+	+
Підтримка різних методів оцінки інформативності ознак	+-	+-	+-
Інтеграція з іншим ПЗ	+-	+	+
Низька вартість	-	+-	+-

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для підтримки процесу синтезу розпізнавальних моделей, зокрема, для оцінювання інформативності ознак.

Проте деякі програмні засоби для синтезу розпізнавальних моделей можуть не підтримувати широкий спектр методів оцінки інформативності ознак, що може обмежувати можливості аналізу і вибору найбільш релевантних ознак. Іншим недоліком є те, що деякі програмні рішення можуть бути дорогими або вимагати значних обчислювальних ресурсів, що може бути недоступно для невеликих компаній або індивідуальних дослідників. Деяке програмне забезпечення може не завжди ефективно справлятися з незбалансованими наборами даних, де певні ознаки домінують над іншими, що може вплинути на точність оцінки інформативності.

Тому актуальною є розробка програмного забезпечення для оцінювання інформативності ознак.

При розробці програмного забезпечення для підтримки процесу оцінювання інформативності ознак необхідно забезпечити такі функціональні вимоги:

- підтримка можливості завантаження вибірки даних, що містить історичну інформацію про досліджувані об'єкти або процеси;
- підтримка можливості вибору методу оцінювання інформативності ознак;
- можливість оцінювання інформативності ознак обраним користувачем методом;
- можливість виведення результатів відбору ознак;
- підтримка можливості оцінювання ефективності методів відбору ознак.

1.3 Висновки за розділом 1

Визначено, що модель є способом опису системи з метою розпізнавання, прогнозування і аналізу стану реальних об'єктів, процесів або явищ; висока складність моделі не завжди гарантує покращення якості розпізнавання, іноді простіші моделі можуть дати кращі результати. Проаналізовано процес моделювання. Відзначено, що важливим етапом в процесі моделювання є визначення змінних і параметрів, на цьому етапі важливо оцінити інформативність кожної ознак з метою недопущення включення в модель неінформативних ознак або надлишкових ознак. Відбір ознак є процесом зменшення кількості вхідних даних при створенні прогнозовної або розпізнавальної моделі, зменшення кількості вхідних змінних може допомогти знизити обчислювальні витрати на моделювання та іноді поліпшити ефективність моделі. Проаналізовано методи відбору ознак. Відзначено, що не існує універсального методу для відбору ознак, тому потрібно знайти найбільш ефективний підхід для конкретної задачі розпізнавання або регресії шляхом систематичних експериментів, необхідно експериментувати з різними моделями, які працюють з різними підмножинами вхідних ознак, відбираючи їх за допомогою різних методів, і визначаючи, який підхід найкраще підходить для конкретної проблеми. Відповідно, для вирішення таких завдань необхідно розробити програмне забезпечення, що дозволяє виконувати оцінювання інформативності ознак за допомогою різних методів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу синтезу розпізнавальних моделей, зокрема, для оцінювання інформативності ознак. Проте деякі програмні засоби для синтезу розпізнавальних моделей можуть не підтримувати широкий спектр методів оцінки інформативності ознак, що може обмежувати можливості аналізу і вибору найбільш релевантних ознак. Іншим недоліком є те, що деякі програмні рішення можуть бути дорогими або

вимагати значних обчислювальних ресурсів, що може бути недоступно для невеликих компаній або індивідуальних дослідників. Деяке програмне забезпечення може не завжди ефективно справлятися з незбалансованими наборами даних, де певні ознаки домінують над іншими, що може вплинути на точність оцінки інформативності. Тому актуальною є розробка програмного забезпечення для оцінювання інформативності ознак.

Сформульовано функціональні вимоги до програмного забезпечення для оцінювання інформативності ознак.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для підтримки процесу оцінювання інформативності ознак було використано мову програмування Python [17], [18].

Python – це мова програмування, яка поєднує точність і функціональність, та є широко відомою за свою простоту в навчанні програмуванню. Як і багато інших мов, вона використовує знаки пунктуації, але має менше граматичних структур, ніж деякі інші мови. Python є інтерпретованою мовою, тобто програмний код виконується без попередньої компіляції [17], [18].

Python також є інтерактивною мовою, що дозволяє працювати безпосередньо з інтерпретатором для написання і тестування коду. Мова підтримує об'єктно-орієнтоване програмування, що дозволяє організовувати код у вигляді об'єктів для зручності та ефективності [17], [18].

Python є відмінним вибором для розробки різних типів програмного забезпечення, від простих текстових процесорів до складних вебзастосунків і ігор [17], [18].

Основні переваги Python включають [17], [18]:

- легкість у використанні, завдяки чому мова програмування Python підходить для створення прототипів та управління функціями без зайвої складності [17], [18];

- інтерактивний режим, який дозволяє швидко тестувати короткі фрагменти коду [17], [18];

- розширена стандартна бібліотека, яка підтримує різні функції програмування, включаючи роботу з вебзастосунками, управління клавіатурою та файлами [17], [18];

- можливість поєднання мови програмування Python з іншими мовами програмування [17], [18];

- простий синтаксис, що полегшує розуміння коду [17], [18];
- легкість інтеграції з іншими мовами програмування, такими як С або С++ [17], [18];
- кросплатформність: Python підтримує різні інструменти і має версії для платформ Android та iOS [17], [18].

Python є відкритим програмним забезпеченням, що дозволяє змінювати та поширювати його без обмежень, оскільки мова є авторським правом, але використовує ліцензію, що дозволяє вільне використання [17], [18].

Основні характеристики Python [17], [18]:

- динамічна типізація даних, що полегшує виявлення помилок при несумісності типів [17], [18];
- велика спільнота розробників, яка підтримує численні бібліотеки та модулі [17], [18];
- широкий спектр базових типів даних, таких як числа (цілі, комплексні, довгі), рядки (ASCII та Unicode), файли і словники [17], [18];
- підтримка об'єктно-орієнтованого програмування, включаючи генерацію і успадкування [17], [18];
- розширені функції програмування, такі як генератори та розбір списків [17], [18];
- можливість організації програмного коду у модулі та пакети для зручності [17], [18];
- обробка винятків для керування помилками [17], [18];
- простота граматики, схожої на англійську, що робить код легким для розуміння [17], [18];
- автоматичне управління пам'яттю, що усуває фрагментацію і зменшує потребу в ручному управлінні пам'яттю [17], [18].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для оцінювання інформативності ознак наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для оцінювання інформативності ознак

Критерій порівняння мов програмування	Мова програмування		
	Python	Matlab	C#
Велика спільнота розробників	+	–	+
Простота синтаксису	+	+-	–
Зручність для створення ПЗ для оцінювання інформативності ознак	+	–	–
Масштабованість	+	+-	+
Бібліотеки для обробки даних	+	+	+-

Отже, для реалізації програмного забезпечення для підтримки процесу оцінювання інформативності ознак обрано мову програмування Python, яка має зрозумілий та простий синтаксис, що полегшує написання і читання програмного коду. Python пропонує велику кількість бібліотек для обробки даних і машинного навчання, таких як NumPy, pandas, Scikit-learn, TensorFlow, Keras. Ці бібліотеки надають інструменти для оцінювання важливості ознак, що значно спрощує реалізацію алгоритмів і моделей. Крім того, Python дозволяє швидко тестувати та перевіряти фрагменти коду в інтерактивному режимі, що зручно для експериментів з різними методами оцінювання ознак.

2.2 Вибір середовища розробки для створення програмного забезпечення для оцінювання інформативності ознак

В якості середовища розробки для створення програмного забезпечення для оцінювання інформативності ознак було обрано середовище PyCharm [19], [20].

Середовище розробки PyCharm є найкращим інтегрованим середовищем розробки для програмування на Python. Багато функцій роблять PyCharm одним із найбільш повних і комплексних середовищ розробки для використання мови програмування Python [19], [20].

PyCharm пропонує безліч модулів, пакетів та інструментів, що значно спрощують процес розробки на Python і знижують необхідні зусилля. Крім того, середовище можна налаштувати відповідно до особистих вподобань та потреб [19], [20].

Середовище розробки PyCharm відзначається своєю комплексністю та широким спектром функцій, що робить його одним з найбільш повних інструментів для Python. PyCharm забезпечує сумісність з різними операційними системами, включаючи Linux, macOS і Windows [19], [20].

Серед ключових функцій PyCharm є графічний інтерфейс для налагодження коду, вбудовані інструменти для тестування програмного забезпечення та інтеграція з розвиненими системами контролю версій. Також підтримується інтеграція з Anaconda, що полегшує роботу з науковими даними та інструментами для аналізу [19], [20].

Важливою перевагою PyCharm є редактор коду, який спрощує написання зрозумілого і читабельного коду Python. Він автоматично виділяє ключові слова, класи і функції, а також надає стандартні помилки для виявлення та виправлення помилок [19], [20].

PyCharm розроблено для спрощення програмування на Python, забезпечуючи зручну роботу з різними операційними системами. Це середовище включає інструменти для аналізу коду, налагодження, тестування та управління версіями. Розробники можуть також створювати власні модулі завдяки наявним інструментам [19], [20].

PyCharm також підтримує роботу з різними джерелами даних і взаємодію з ними без потреби в додаткових інструментах. Хоча основний фокус середовища на Python, його також можна використовувати для розробки HTML, CSS і JavaScript [19], [20].

Особливістю PyCharm є його підтримка наукових бібліотек, що дозволяє розробникам реалізовувати проєкти в галузі аналізу даних і машинного навчання [19], [20].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для оцінювання інформативності ознак наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для оцінювання інформативності ознак

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Atom	Sublime Text
Інтеграція з системами контролю версій	+	+	–
Вбудоване управління пакетами	+	+–	–
Інтеграція з бібліотеками для обробки даних	+	+–	+–
Професійна підтримка та документація	+	+–	–
Розширена підтримка Python	+	–	+

Таким чином, для створення програмного забезпечення для підтримки процесу оцінювання інформативності ознак обрано середовище розробки PyCharm, що надає потужні інструменти для розробки, налагодження та тестування Python-коду, зокрема, інструменти для автоматичного завершення коду, перевірки синтаксису та інші функції, які полегшують розробку програмного забезпечення. PyCharm підтримує інтеграцію з системами

контролю версій, що полегшує співпрацю в командах і керування версіями коду.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак

З метою можливості розробки програмного забезпечення для оцінювання інформативності ознак створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак враховуємо те, що користувачі програми мають можливості завантаження вибірки даних, що містить історичну інформацію про досліджувані об'єкти або процеси, вибору методу оцінювання інформативності ознак, проведення оцінювання інформативності ознак обраним методом, виведення результатів відбору ознак, оцінювання ефективності методів відбору ознак.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для оцінювання інформативності ознак можна описати таким чином:

- користувач програмного забезпечення для оцінювання інформативності ознак запускає програму;
- після запуску програмного забезпечення для оцінювання інформативності ознак користувач бачить головну форму;
- користувач завантажує вибірку даних, що містить історичну інформацію про досліджувані об'єкти або процеси;
- користувач обирає метод оцінювання інформативності ознак;
- користувач натискає відповідну кнопку для оцінювання інформативності ознак обраним методом;
- користувач ознайомлюється з результатом відбору ознак та порівнює ефективність методів відбору ознак;

– користувач завершує роботу з програмним забезпеченням для оцінювання інформативності ознак.

Враховуючи сценарії роботи користувачів при роботі з програмою, модель взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.

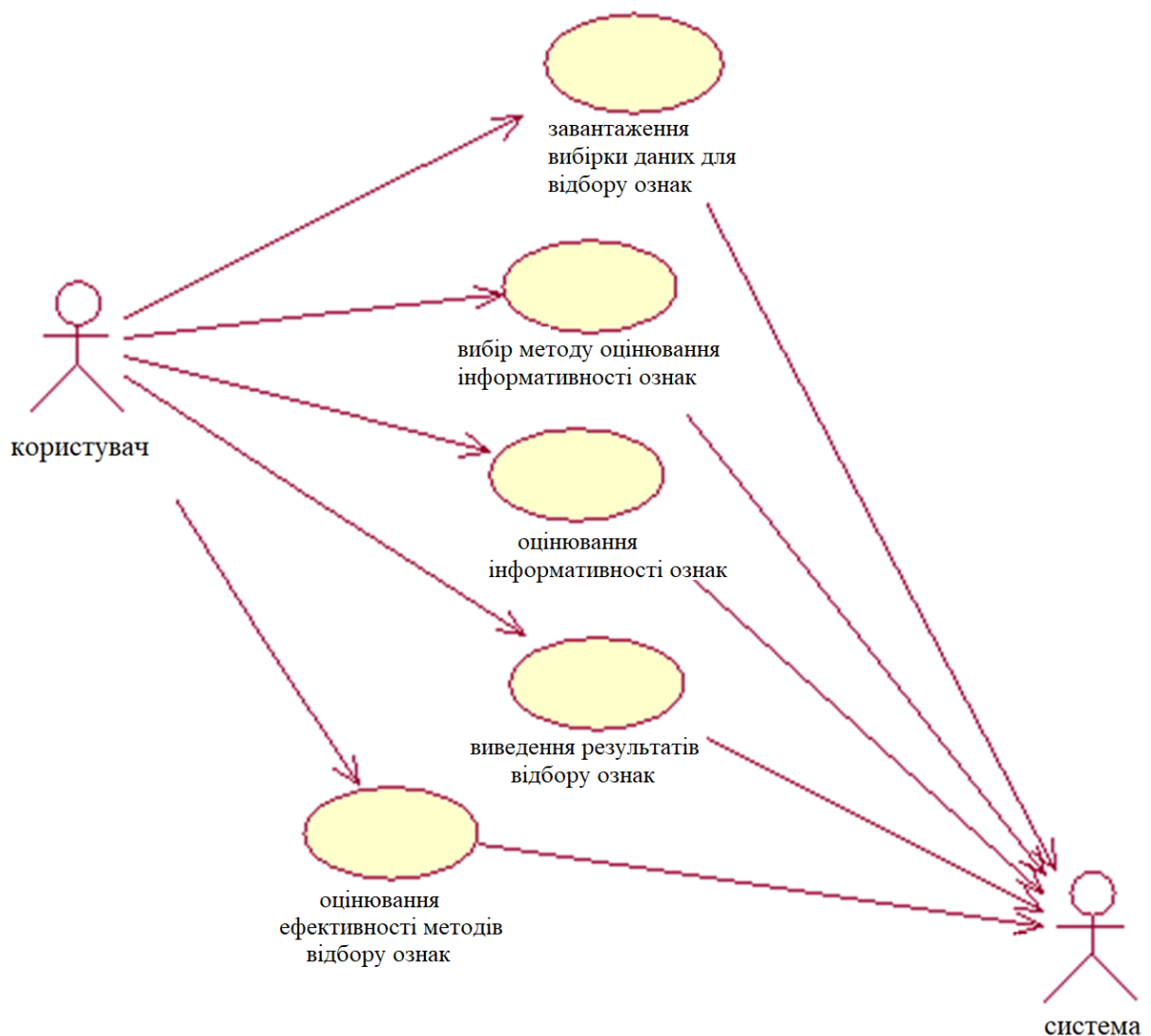


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак, яка подана у вигляді

діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для відбору інформативних ознак та синтезу розпізнавальних моделей.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для підтримки процесу оцінювання інформативності ознак обрано мову програмування Python, яка має зрозумілий та простий синтаксис, що полегшує написання і читання програмного коду. Python пропонує велику кількість бібліотек для обробки даних і машинного навчання, таких як NumPy, pandas, Scikit-learn, TensorFlow, Keras. Ці бібліотеки надають інструменти для оцінювання важливості ознак, що значно спрощує реалізацію алгоритмів і моделей. Крім того, Python дозволяє швидко тестувати та перевіряти фрагменти коду в інтерактивному режимі, що зручно для експериментів з різними методами оцінювання ознак.

Для створення програмного забезпечення для підтримки процесу оцінювання інформативності ознак обрано середовище розробки PyCharm, що надає потужні інструменти для розробки, налагодження та тестування Python-коду, зокрема, інструменти для автоматичного завершення коду, перевірки синтаксису та інші функції, які полегшують розробку програмного забезпечення. PyCharm підтримує інтеграцію з системами контролю версій, що полегшує співпрацю в командах і керування версіями коду.

Створено модель взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для відбору інформативних ознак та синтезу розпізнавальних моделей.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для підтримки процесу оцінювання інформативності ознак

Структуру програмного забезпечення для оцінювання інформативності ознак наведено на рис. 3.1.

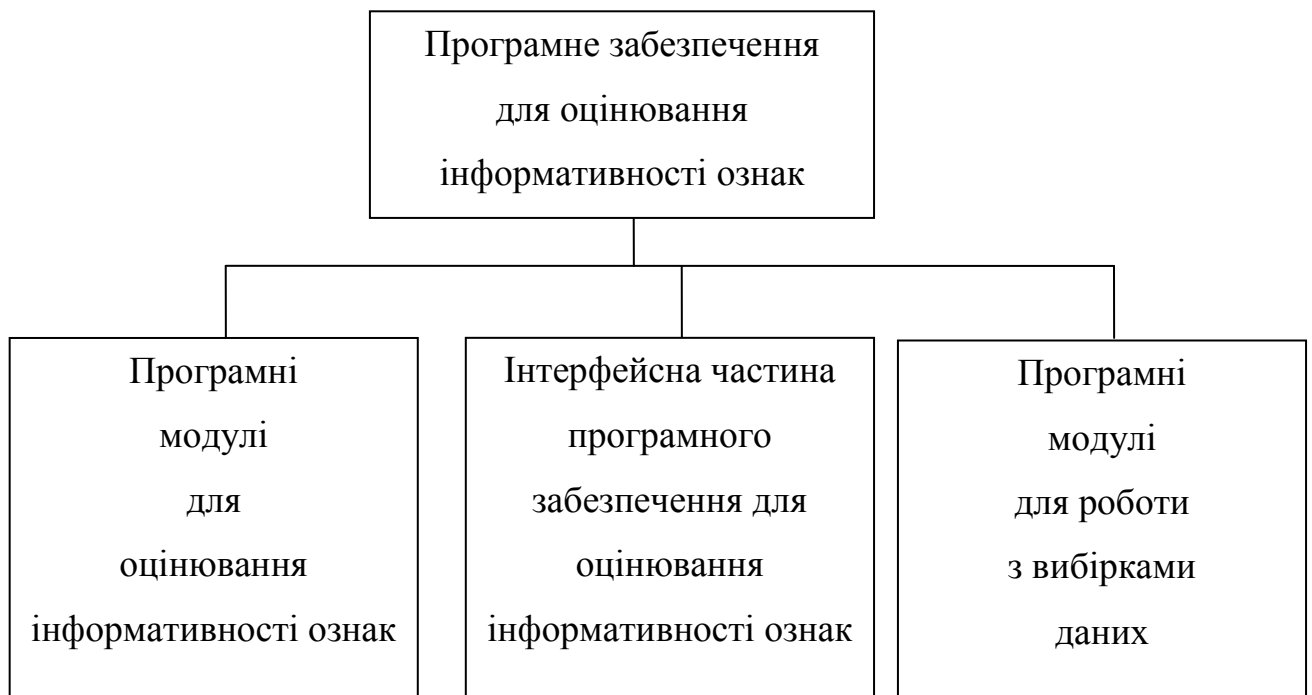


Рисунок 3.1 – Структура програмного забезпечення для оцінювання інформативності ознак

Як видно з рис. 3.1, програмне забезпечення для оцінювання інформативності ознак складається з таких основних частин:

– програмні модулі для оцінювання інформативності ознак – містять програмну реалізацію методів та критеріїв оцінювання інформативності ознак, зокрема, методу відбору ознак на основі синтезу моделей, метод випадкового лісу, методу головних компонентів, методу дискримінантного аналізу, методу локального лінійного вбудовування та методу розподіленого стохастичного вбудовування та ін.;

- інтерфейсна частина – містить структуру інтерфейсних форм користувача для роботи з програмою для оцінювання інформативності ознак;
- програмні модулі для роботи з вибірками даних – містять засоби для роботи з файлами, що містять вибірки вхідних даних, які необхідно проаналізувати та дослідити на наявність неінформативних ознак.

3.2 Функціонування програмного забезпечення для оцінювання інформативності ознак

Функціонування програмного забезпечення для підтримки процесу оцінювання інформативності ознак подамо за допомогою схеми, зображеної на рис. 3.2.

Робота програмного забезпечення для оцінювання інформативності ознак з введення користувачем файлу з вибіркою вхідних даних, яку необхідно проаналізувати та дослідити на наявність неінформативних ознак. На наступному етапі обирається метод відбору інформативних ознак (пропонується обрати метод з такої множини методів: метод відбору ознак на основі синтезу моделей, метод відбору ознак на основі випадкового лісу, метод головних компонентів, метод дискримінантного аналізу, метод локального лінійного вбудовування, метод розподіленого стохастичного вбудовування).

Після цього виконується оцінювання інформативності ознак на основі обраного методу. Далі користувачу виводяться результати відбору інформативних ознак, зокрема, інформативність кожної ознаки, а також короткі рекомендації щодо використання отриманих результатів. За необхідності користувач може зберегти отримані результати.

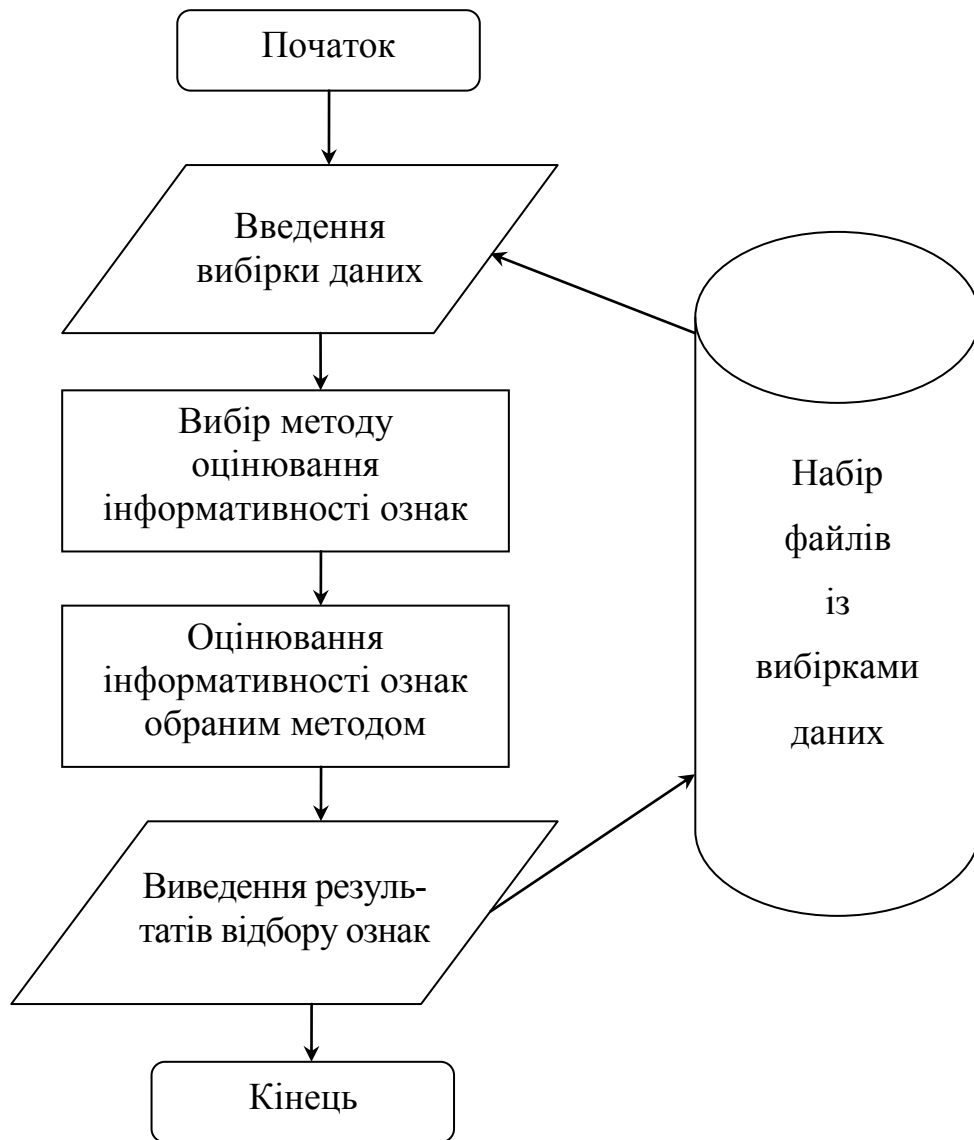


Рисунок 3.2 – Схема функціонування програмного забезпечення для підтримки процесу оцінювання інформативності ознак

3.3 Використання методів оцінювання інформативності ознак у програмному забезпеченні для відбору ознак

У програмному забезпеченні реалізовано різні методи оцінювання інформативності ознак [21]-[25].

Оцінювання інформативності ознак можна розділити на три основні групи методів: методи фільтрації, методи зв'язування (на основі побудови моделей) та методи вбудовування. Розглянемо основні особливості кожного з цих підходів [21]-[25].

Методи фільтрації ґрунтуються на статистичних техніках і аналізують кожну ознаку окремо. Вони дозволяють оцінювати та класифікувати ознаки за їхньою значимістю, наприклад, за мірою кореляції між ознакою та цільовою змінною [21]-[25].

Фільтраційні методи поділяються на глобальні та локальні. Глобальні методи оцінюють ймовірність кожної ознаки і відбирають певну кількість найбільш важливих ознак. Локальні методи також оцінюють ознаки, але перетворюють їх на одне число, а потім вибирають меншу кількість найважливіших ознак [21]-[25].

Методи фільтрації мають низькі обчислювальні витрати, оскільки їхня ефективність не залежить безпосередньо від загальної кількості даних. Вони швидкі у виконанні і можуть справлятися з великими обсягами даних, навіть коли кількість ознак перевищує кількість прикладів. Проте, їхнім недоліком є те, що вони оцінюють ознаки окремо, і тому відбір інформативних ознак не завжди веде до покращення точності моделі [21]-[25].

Методи зв'язування: основною особливістю цього типу методів є те, що класифікатор працює з підмножиною ознак, вибраних з первинного набору. Спочатку вибирається набір ознак, який найбільше підходить для навчання, а потім його ефективність перевіряється на тестовій вибірці (яка не береться до уваги при виборі підмножини) [21]-[25].

Існують дві основні стратегії для цього підходу: включення (попередній відбір) і виключення (пізній відбір). Перша стратегія починається з пустої підмножини, до якої поступово додаються різні ознаки, вибираючи найкращу комбінацію на кожному етапі. Друга стратегія починається з початкового набору ознак, з якого поступово видаляються менш важливі ознаки на основі оцінки класифікатора [21]-[25].

Метод зв'язування вибирає набори ознак за допомогою числових функцій і методів пошуку. Проте, кількість можливих комбінацій зростає експоненціально з кількістю ознак, що ускладнює процес пошуку [21]-[25].

Вбудовані методи: цей тип методів не відокремлює процес вибору ознак від процесу навчання моделі, а здійснює вибір в рамках навчання. Вбудовані методи вимагають менших обчислювальних ресурсів у порівнянні з методами зв'язування, але є більш точними ніж фільтраційні методи. Основна стратегія – постійне скорочення. Ідея полягає у створенні алгоритму, який не лише мінімізує помилки, але й обмежує кількість змінних. Оцінка продуктивності класифікатора здійснюється шляхом створення моделі, яка найбільше підходить для прогнозування [21]-[25].

Наприклад, у випадку використання лінійної регресії як класифікатора вибираються поліноміальні коефіцієнти, що відображають різницю між функцією та цільовими змінними. Стандартні помилки використовуються для оцінки якості вибраних коефіцієнтів. Параметри синтезованої моделі вибираються так, щоб сума оцінок, передбачених класифікатором, наближалася до реальних значень [21]-[25].

Методи зв'язування та вбудовані методи можуть потребувати більше часу на обчислення порівняно з фільтраційними методами і часто використовуються в комбінації з конкретними типами класифікаторів або розпізнавальних моделей [21]-[25].

Останнім часом часто використовують бази даних з сотнями або навіть тисячами ознак. Якщо кількість ознак дорівнює або перевищує кількість спостережень, модель машинного навчання може не забезпечити прийнятну точність на тестових даних. Щоб уникнути цієї проблеми, необхідно застосовувати стандартизацію або зменшення розмірності (відбір ознак). У машинному навчанні розмір набору даних визначається кількістю змінних, що його складають [21]-[25].

Зменшення кількості ознак може допомогти зменшити ризик неякісного навчання моделі, але методи відбору ознак можуть надати інші переваги, такі як: підвищення точності моделі, зменшення складності, прискорення процесу навчання, покращення опису даних, збільшення рівня розуміння моделі [21]-[25].

Метод відбору ознак спрямований на зменшення кількості змінних у наборі даних шляхом видалення непотрібних ознак. Це дозволяє зберегти більшість інформації, що міститься в оригінальних даних, і забезпечити кращу структуру для подальшого аналізу [21]-[25]. Відмінність між відбором і видаленням ознак полягає в тому, що відбір ознак фокусується на оцінці важливості кожної ознаки і видаленні менш значущих, створюючи при цьому нові, більш релевантні ознаки [21]-[25].

Аналіз головних компонентів є широко використовуваним методом зменшення лінійної розмірності. Використовуючи аналіз головних компонентів, обробляються вихідні дані та шукається комбінація ознак, яка максимізує варіацію в даних, при цьому зменшуючи їхню загальну кількість. Аналіз головних компонентів досягає цього шляхом виявлення основних напрямків варіації в даних і проєктування їх на ці напрямки. Важливість кожного напрямку визначається на основі того, як сильно дані розподілені вздовж цього напрямку. Аналіз головних компонентів є методом, який орієнтується на варіацію даних, а не на їхній зміст, що може призвести до неточної класифікації [21]-[25].

Лінійний дискримінантний аналіз є розширенням методів зменшення розмірності, використовуваних у класифікаторах і машинному навчанні. Лінійний дискримінантний аналіз намагається максимізувати різницю між середніми значеннями різних класів і мінімізувати розсіювання всередині кожного класу. Це допомагає покращити розмежування класів, коли дані проєктуються на менш вимірний простір. Лінійний дискримінантний аналіз є ефективним для покращення розподілу між класами, але його результати можуть бути неточними, якщо дані не відповідають нормальному розподілу ймовірності [21]-[25].

Метод локального лінійного вбудовування (Local Linear Embedding) є методом зменшення розмірності, який базується на багатьох дослідженнях. Метод локального лінійного вбудовування створює новий розмір даних шляхом локального відображення. На відміну від методів, таких як аналіз

головних компонентів, які добре працюють для лінійних зв'язків між ознаками, метод локального лінійного вбудовування краще підходить для нелінійних випадків. Метод локального лінійного вбудовування зберігає локальні відстані між сусідніми точками, що дозволяє зберігати їхню структуру у новому, зменшеному просторі, подібно до серії аналізу головних компонентів, але з акцентом на нелінійні залежності [21]-[25].

Отже, методи відбору ознак допомагають зменшити кількість ознак (факторів), які використовуються в моделі машинного навчання, щоб підвищити її ефективність і зменшити складність. Коли використовується метод на основі моделей для відбору ознак з деревом рішень, основними параметрами, які можна налаштовувати, є важливість ознак, максимальна глибина дерева, мінімальна кількість зразків для поділу та інші. Древа рішень надають інформативність кожній ознаці на основі того, наскільки добре вона допомагає розділяти дані [21]-[25].

Метод нелінійного зменшення розмірності часто використовується для візуалізації великих наборів даних в таких областях, як обробка природної мови та аналіз текстів [21]-[25]. Цей метод працює з ймовірнісними розподілами, які включають пари подібності між вхідними даними. Метод нелінійного зменшення використовує ймовірність для оцінки різниці між розподілами, зменшуючи її завдяки нахилу [21]-[25].

У методі нелінійного зменшення розмірності розподіл даних у високій розмірності моделюється як гаусівський розподіл, а у нижчій розмірності як розподіл Стюдента. Це дозволяє зменшити різницю у відстанях між сусідніми точками, наближаючи їх до меншої розмірності [21]-[25].

3.4 Проєктування інтерфейсу програмного забезпечення для оцінювання інформативності ознак

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підтримки процесу оцінювання інформативності ознак

виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для оцінювання інформативності ознак враховані функціональні вимоги до програми:

- підтримка можливості завантаження вибірки даних, що містить історичну інформацію про досліджувані об’єкти або процеси;
- підтримка можливості вибору методу оцінювання інформативності ознак;
- можливість оцінювання інформативності ознак обраним користувачем методом;
- можливість виведення результатів відбору ознак;
- підтримка можливості оцінювання ефективності методів відбору ознак.

Інтерфейс головної форми програмного забезпечення для оцінювання інформативності ознак наведено на рис. 3.3.

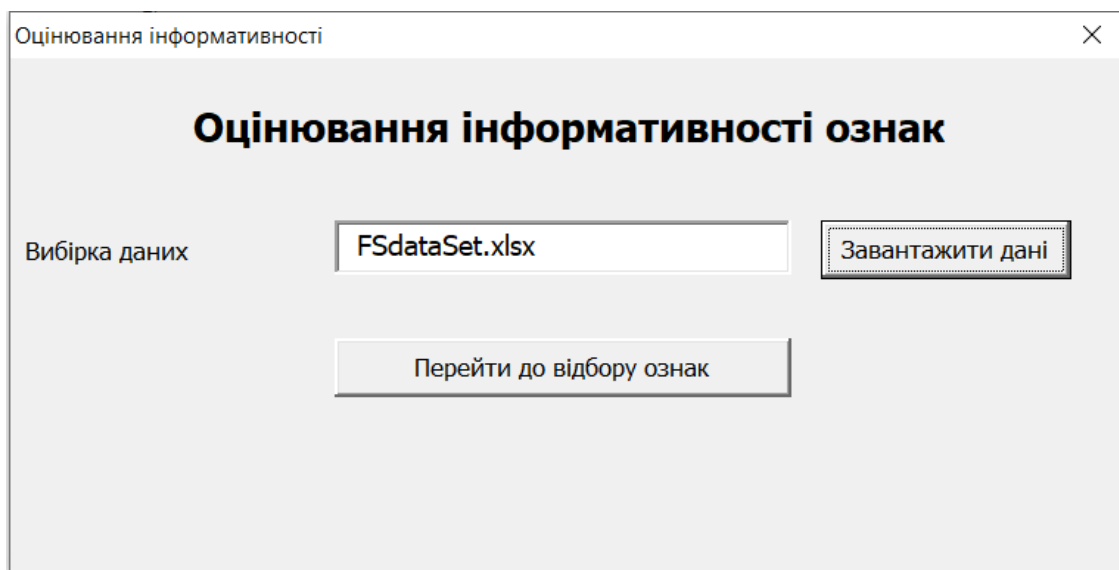


Рисунок 3.3 – Інтерфейс головної форми програмного забезпечення для оцінювання інформативності ознак

Форма вибору методу оцінювання інформативності ознак наведена на рис. 3.4.

Оцінювання інформативності

Оцінювання інформативності ознак

Оберіть метод оцінювання інформативності:

Введіть параметри методу оцінювання інформативності:

Тип моделі:

Параметри моделі:

Максимальна глибина

Мінімальна кількість зразків у вузлі:

Критерій якості розподілу зразків у вузлі:

Рисунок 3.4 – Форма вибору методу оцінювання інформативності ознак

Форму з результатами оцінювання інформативності ознак наведено на рис. 3.5.

Результати оцінювання інформативності

Оцінювання інформативності ознак

Результати оцінювання інформативності:

Вхідні дані:
 Вибірка даних: FSdataSet.xlsx
 Кількість ознак у вибірці даних: 15 ознак
 Метод, що використовувався для оцінювання інформативності: Метод на основі моделі
 Параметри методу: максимальна глибина - 5, мінімальна кількість зразків у вузлі - 10, критерій якості розподілу зразків у вузлі - Ентропія

Результати оцінювання інформативності:
 $I(x_5)=0.21$, $I(x_2)=0.18$, $I(x_8)=0.15$, $I(x_3)=0.12$, $I(x_{11})=0.11$, $I(x_7)=0.09$, $I(x_1)=0.08$, $I(x_{10})=0.07$, $I(x_6)=0.06$, $I(x_{12})=0.05$, $I(x_4)=0.05$, $I(x_9)=0.04$, $I(x_{14})=0.03$, $I(x_{13})=0.02$, $I(x_{15})=0.01$

Ознака 5 має найвищу інформативність (0.21), що вказує на те, що ця ознака найбільше впливає на результат моделі. Ознака 2 і Ознака 8 також мають високу інформативність, але меншу, ніж Ознака 5.
 Ознаки з меншою інформативністю (наприклад, Ознака 13 і Ознака 15) можуть бути менш корисними для моделі і можуть бути потенційно виключені або перевірені на предмет їхньої додаткової інформаційної цінності.

Рисунок 3.5 – Форма з результатами оцінювання інформативності ознак

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для підтримки процесу оцінювання інформативності ознак.

Запропоновано структуру програмного забезпечення для підтримки процесу оцінювання інформативності ознак. Визначено, що програмне забезпечення для оцінювання інформативності ознак складається з таких основних частин: програмні модулі для оцінювання інформативності ознак (містять програмну реалізацію методів та критеріїв оцінювання інформативності ознак), інтерфейсна частина (містить форми користувача для роботи з програмою), програмні модулі для роботи з вибірками даних (містять засоби для роботи з файлами, що містять вибірки вхідних даних).

Описано особливості реалізації програмного забезпечення для оцінювання інформативності ознак. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підтримки процесу оцінювання інформативності ознак.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу розпізнавання образів, зокрема, для відбору інформативних ознак. Програма написана на мові Python, яка має зрозумілий та простий синтаксис, що полегшує написання і читання програмного коду, крім того Python пропонує велику кількість бібліотек для обробки даних і машинного навчання, які надають інструменти для оцінювання важливості ознак, що значно спрощує реалізацію алгоритмів і моделей. В якості середовища розробки для створення програмного забезпечення для оцінювання інформативності ознак обрано PyCharm, що надає потужні інструменти для розробки, налагодження та тестування Python-коду, зокрема, інструменти для автоматичного завершення коду, перевірки синтаксису та інші функції, які полегшують розробку програмного забезпечення.

Програмне забезпечення для підтримки процесу оцінювання інформативності ознак забезпечує виконання таких функцій:

- підтримка можливості завантаження вибірки даних, що містить історичну інформацію про досліджувані об'єкти або процеси;
- підтримка можливості вибору методу оцінювання інформативності ознак;
- можливість оцінювання інформативності ознак обраним користувачем методом;
- можливість виведення результатів відбору ознак;
- підтримка можливості оцінювання ефективності методів відбору ознак.

Для експлуатації програмного забезпечення для підтримки процесу оцінювання інформативності ознак необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 2 гігабайти і більше;
- не менше 100 Мб вільного простору на жорсткому диску;
- процесор із тактовою частотою не менше 2 ГГц;
- встановлена операційна система.

Функціональні характеристики розробленого програмного забезпечення для оцінювання інформативності ознак наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для оцінювання інформативності ознак наведено у додатку Б.

4.2 Характеристики програми для оцінювання інформативності ознак

Програмне забезпечення для підтримки процесу оцінювання інформативності ознак складається з модулів для оцінювання інформативності ознак, інтерфейсної частини та програмних модулів для роботи з вибірками даних.

Робота програмного забезпечення для оцінювання інформативності ознак з введення користувачем файлу з вибіркою вхідних даних, яку необхідно проаналізувати та дослідити на наявність неінформативних ознак. На наступному етапі обирається метод відбору інформативних ознак (пропонується обрати метод з такої множини методів: метод відбору ознак на основі випадкового лісу, метод головних компонентів, метод дискримінантного аналізу, метод локального лінійного вбудовування, метод розподіленого стохастичного вбудовування).

Після цього виконується оцінювання інформативності ознак на основі обраного методу. Далі користувачу виводяться результати відбору інформативних ознак, зокрема, точність моделі, синтезованої на основі відібраного набору ознак. За необхідності користувач може зберегти отримані результати.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для підтримки процесу оцінювання інформативності ознак необхідно запустити основний ру-файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для підтримки процесу оцінювання інформативності ознак є вибірка даних, а також метод оцінювання інформативності ознак.

Вихідними даними програмного забезпечення для підтримки процесу оцінювання інформативності ознак є інформація щодо інформативності ознак, а також короткі пояснення результатів оцінювання інформативності ознак.

4.3.3 Повідомлення

Користувач програмного забезпечення для підтримки процесу оцінювання інформативності ознак може отримати такі повідомлення: повідомлення про некоректні вхідні дані, повідомлення про завершення процесу оцінювання ознак та ін.

4.4 Виконання програмного забезпечення для підтримки процесу оцінювання інформативності ознак

Після запуску програмного забезпечення для підтримки процесу оцінювання інформативності ознак користувачу відображається головна форма (рис. 4.1), де користувач може завантажити вибірку даних, що містить інформацію про досліджувані об'єкти або процеси.

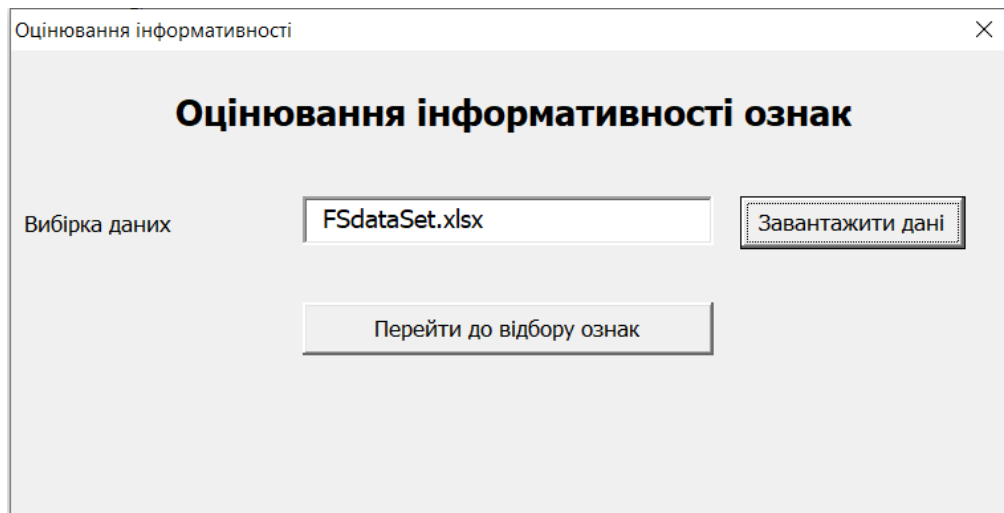


Рисунок 4.1 – Інтерфейс головної форми програмного забезпечення для оцінювання інформативності ознак

Після завантаження вибірки даних для аналізу та пошуку інформативних ознак, користувач має можливість вибору методу оцінювання інформативності ознак. Метод обирається зі списку. В залежності від обраного методу, користувач додатково вводить параметри методу або залишає ті, що програма пропонує за замовченням (рис. 4.2).

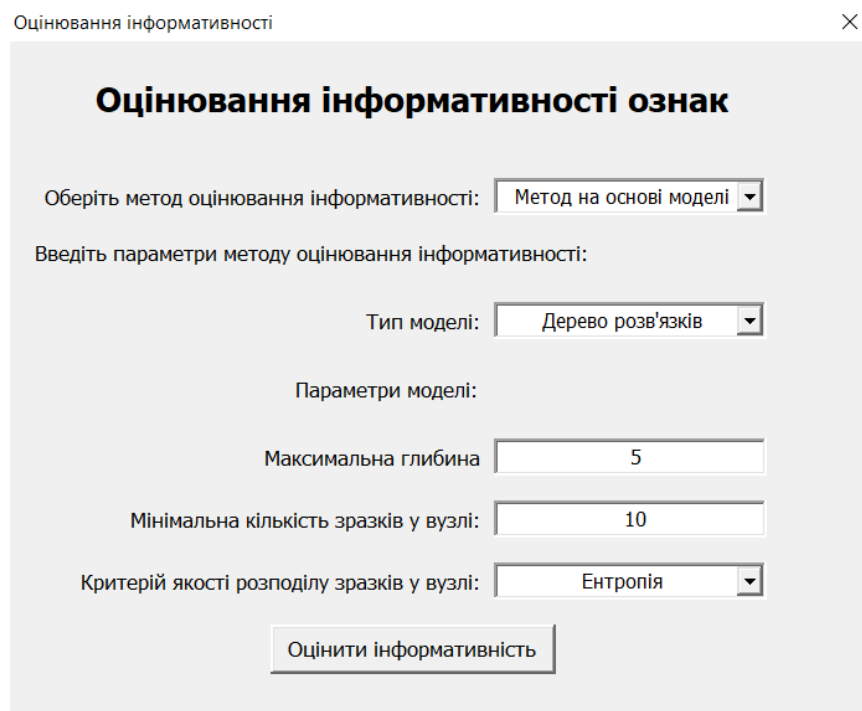


Рисунок 4.2 – Вибір методу оцінювання інформативності ознак

Після натискання на кнопку виводиться форма з результатами оцінювання інформативності ознак (рис. 4.3). У цій формі виводиться інформація про вхідні дані для аналізу (вибірка даних, її параметри, метод, що використовувався для оцінювання інформативності ознак) та результати оцінювання інформативності ознак. Як результати виводяться числові оцінки інформативності кожної ознаки (ознаки відсортовано від найбільш значущої до найменш інформативної), а також надається коротке пояснення результатів оцінювання інформативності ознак.

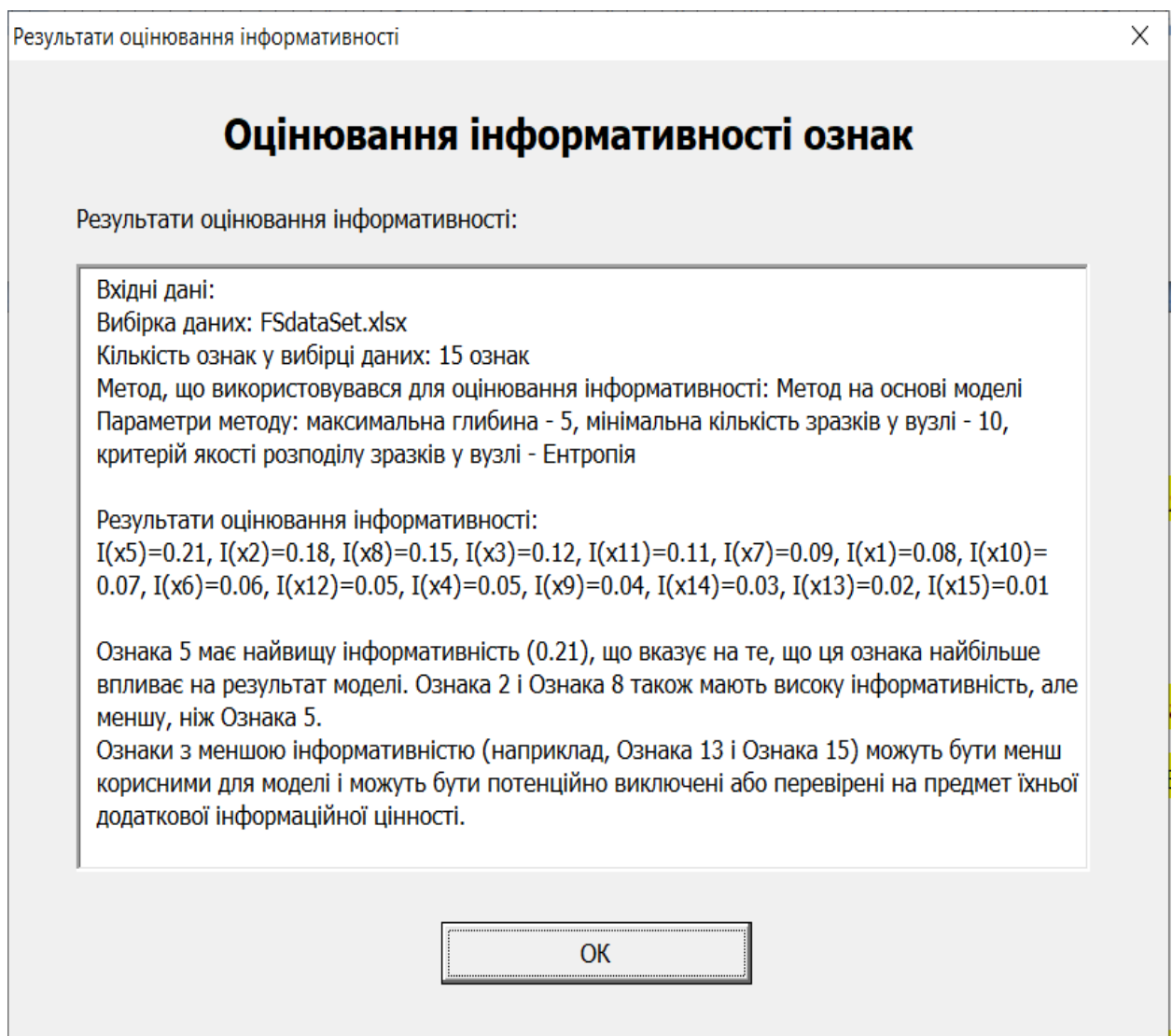


Рисунок 4.3 – Результати оцінювання інформативності ознак

4.5 Тестування програмного забезпечення для підтримки процесу оцінювання інформативності ознак

Виконано тестування програмного забезпечення для оцінювання інформативності ознак.

Виявлено, що програма для підтримки процесу оцінювання інформативності ознак функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу оцінювання інформативності ознак.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою оцінювання інформативності ознак.

4.6 Висновки за розділом 4

Описано програмне забезпечення для оцінювання інформативності ознак. Виконано тестування розробленого програмного забезпечення для оцінювання інформативності ознак. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу оцінювання інформативності ознак.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес синтезу розпізнавальних моделей.

Визначено, що модель є способом опису системи з метою розпізнавання, прогнозування і аналізу стану реальних об'єктів, процесів або явищ; висока складність моделі не завжди гарантує покращення якості розпізнавання, іноді простіші моделі можуть дати кращі результати. Проаналізовано процес моделювання. Відзначено, що важливим етапом в процесі моделювання є визначення змінних і параметрів, на цьому етапі важливо оцінити інформативність кожної ознак з метою недопущення включення в модель неінформативних ознак або надлишкових ознак. Відбір ознак є процесом зменшення кількості вхідних даних при створенні прогнозної або розпізнавальної моделі, зменшення кількості вхідних змінних може допомогти знизити обчислювальні витрати на моделювання та іноді поліпшити ефективність моделі. Проаналізовано методи відбору ознак. Відзначено, що не існує універсального методу для відбору ознак, тому потрібно знайти найбільш ефективний підхід для конкретної задачі розпізнавання або регресії шляхом систематичних експериментів, необхідно експериментувати з різними моделями, які працюють з різними підмножинами вхідних ознак, відбираючи їх за допомогою різних методів, і визначаючи, який підхід найкраще підходить для конкретної проблеми. Відповідно, для вирішення таких завдань необхідно розробити програмне забезпечення, що дозволяє виконувати оцінювання інформативності ознак за допомогою різних методів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу синтезу розпізнавальних моделей, зокрема, для оцінювання інформативності ознак. Проте деякі програмні засоби для синтезу розпізнавальних моделей можуть не підтримувати широкий спектр методів оцінки інформативності ознак, що

може обмежувати можливості аналізу і вибору найбільш релевантних ознак. Іншим недоліком є те, що деякі програмні рішення можуть бути дорогими або вимагати значних обчислювальних ресурсів, що може бути недоступно для невеликих компаній або індивідуальних дослідників. Деяке програмне забезпечення може не завжди ефективно справлятися з незбалансованими наборами даних, де певні ознаки домінують над іншими, що може вплинути на точність оцінки інформативності. Тому актуальною є розробка програмного забезпечення для оцінювання інформативності ознак.

Сформульовано функціональні вимоги до програмного забезпечення для оцінювання інформативності ознак.

Для реалізації програмного забезпечення для підтримки процесу для оцінювання інформативності ознак обрано мову програмування Python, яка має зрозумілий та простий синтаксис, що полегшує написання і читання програмного коду. Python пропонує велику кількість бібліотек для обробки даних і машинного навчання, таких як NumPy, pandas, Scikit-learn, TensorFlow, Keras. Ці бібліотеки надають інструменти для оцінювання важливості ознак, що значно спрощує реалізацію алгоритмів і моделей. Крім того, Python дозволяє швидко тестувати та перевіряти фрагменти коду в інтерактивному режимі, що зручно для експериментів з різними методами оцінювання ознак.

Для створення програмного забезпечення для підтримки процесу оцінювання інформативності ознак обрано середовище розробки PyCharm, що надає потужні інструменти для розробки, налагодження та тестування Python-коду, зокрема, інструменти для автоматичного завершення коду, перевірки синтаксису та інші функції, які полегшують розробку програмного забезпечення. PyCharm підтримує інтеграцію з системами контролю версій, що полегшує співпрацю в командах і керування версіями коду.

Наукова новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати

програмне забезпечення для відбору інформативних ознак та синтезу розпізнавальних моделей.

Практична цінність результатів роботи полягає у тому, що розроблено програмне забезпечення для оцінювання інформативності ознак.

Запропоновано структуру програмного забезпечення для підтримки процесу оцінювання інформативності ознак. Визначено, що програмне забезпечення для оцінювання інформативності ознак складається з таких основних частин: програмні модулі для оцінювання інформативності ознак (містять програмну реалізацію методів та критеріїв оцінювання інформативності ознак), інтерфейсна частина (містить форми користувача для роботи з програмою), програмні модулі для роботи з вибірками даних (містять засоби для роботи з файлами, що містять вибірки вхідних даних).

Описано особливості реалізації програмного забезпечення для оцінювання інформативності ознак. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підтримки процесу оцінювання інформативності ознак.

Виконано тестування розробленого програмного забезпечення для оцінювання інформативності ознак. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу оцінювання інформативності ознак.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Introduction to Mathematical Modelling [Electronic resource]. – Access mode: <https://ubcmath.github.io/MATH360/process/overview.html>.
2. How to Make a Mathematical Model [Electronic resource]. – Access mode: <https://www.wikihow.com/Make-a-Mathematical-Model>.
3. Ten simple rules for tackling your first mathematical models [Electronic resource]. – Access mode: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008539>.
4. Introduction to Modeling [Electronic resource]. – Access mode: https://math.libretexts.org/Courses/Mt._San_Jacinto_College/Ideas_of_Mathematics/07%3A_Modeling/7.01%3A_Introduction_to_Modeling.
5. How and why to build a mathematical model: A case study using prion aggregation [Electronic resource]. – Access mode: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7152750/>.
6. An introduction to mathematical modelling [Electronic resource]. – Access mode: <https://medium.com/swlh/constructing-a-mathematical-model-understanding-and-controlling-the-natural-world-with-maths-b222939ee8be>.
7. What Is Mathematical Modelling [Electronic resource]. – Access mode: <https://www.mathscareers.org.uk/what-is-mathematical-modelling/>.
8. How to Choose a Feature Selection Method For Machine Learning [Electronic resource]. – Access mode: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>.
9. Feature Selection Techniques in Machine Learning [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning/>.
10. Feature Selection Tutorial with Python Examples [Electronic resource]. – Access mode: <https://arxiv.org/pdf/2106.06437>.

11. Feature Selection – Ten Effective Techniques with Examples [Electronic resource]. – Access mode: <https://www.machinelearningplus.com/machine-learning/feature-selection/>.
12. Best Image Recognition Software [Electronic resource]. – Access mode: <https://www.g2.com/categories/image-recognition>.
13. Best Machine Learning Software List [Electronic resource]. – Access mode: <https://theqalead.com/tools/best-machine-learning-software/>.
14. SuperAnnotate [Electronic resource]. – Access mode: <https://www.superannotate.com/>.
15. Dataloop [Electronic resource]. – Access mode: <https://dataloop.ai/>.
16. Neural Designer [Electronic resource]. – Access mode: <https://www.neuraldesigner.com/>.
17. Le tutoriel Python [Electronic resource]. – Access mode: <https://docs.python.org/fr/3/tutorial/index.html>.
18. Installing Python modules [Electronic resource]. – Access mode: <https://help.pythonanywhere.com/pages/InstallingNewModules>.
19. PyCharm. Python pour la science des données et le développement web [Electronic resource]. – Access mode: <https://www.jetbrains.com/fr-fr/pycharm/>.
20. PyCharm: all about the most popular Python IDE [Electronic resource]. – Access mode: <https://datascientest.com/en/pycharm-all-about-the-most-popular-python-ide>.
21. An Introduction to Feature Selection [Electronic resource]. – Access mode: <https://machinelearningmastery.com/an-introduction-to-feature-selection/>.
22. The 5 Feature Selection Algorithms every Data Scientist should know [Electronic resource]. – Access mode: <https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2>.
23. Feature Selection In Machine Learning: All You Need to Know [Electronic resource]. – Access mode: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/feature-selection-in-machine-learning>.

24. Feature selection techniques for machine learning: a survey of more than two decades of research [Electronic resource]. – Access mode: <https://link.springer.com/article/10.1007/s10115-023-02010-5>.

25. Feature Selection for Machine Learning: 3 Categories and 12 Methods [Electronic resource]. – Access mode: <https://towardsdatascience.com/feature-selection-for-machine-learning-3-categories-and-12-methods-6a4403f86543>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для підтримки процесу відбору інформативних ознак.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація методів оцінювання значущості описових характеристик об'єктів», затверджене наказом Національного університету «Запорізька політехніка» № 498 від 29 листопада 2024 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу оцінювання інформативності ознак.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підтримки процесу оцінювання інформативності ознак забезпечує виконання таких функцій:

- підтримка можливості завантаження вибірки даних, що містить історичну інформацію про досліджувані об'єкти або процеси;
- підтримка можливості вибору методу оцінювання інформативності ознак;
- можливість оцінювання інформативності ознак обраним користувачем методом;
- можливість виведення результатів відбору ознак;

– підтримка можливості оцінювання ефективності методів відбору ознак.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для підтримки процесу оцінювання інформативності ознак повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для оцінювання інформативності ознак повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для оцінювання інформативності ознак необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для оцінювання інформативності ознак необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 2 гігабайти і більше;
- не менше 100 Мб вільного простору на жорсткому диску;
- процесор із тактовою частотою не менше 2 ГГц;
- встановлена операційна система.

A.3.6 Вимоги до маркування і пакування

Програма для підтримки процесу оцінювання інформативності ознак може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для оцінювання інформативності ознак».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для підтримки процесу оцінювання інформативності ознак є вибірка даних, а також метод оцінювання інформативності ознак.

Вихідними даними програмного забезпечення для підтримки процесу оцінювання інформативності ознак є інформація щодо інформативності ознак, а також короткі пояснення результатів оцінювання інформативності ознак.

ДОДАТОК Б**Фрагмент тексту програми**

```

drf = pd.read_csv(dataset)
prcnt_miss = drf.isnull().sum() * 100 / len(drf)
miss_znach = pd.DataFrame({'prcnt_miss': prcnt_miss})
miss_znach.sort_znach(by = 'prcnt_miss' , ascending=False)

X = drf.drop(['cls'], axis = 1)
Y = drf['cls']
X = pd.get_dummies(X, prefix_sep='_')
Y = LabelEncoder().fit_transform(Y)

def lis_test(X, Y):
    Xtr, Xtst, Ytr, Ytst = trntstsplt(X, Y, tstsz = 0.30,
random_state = 101)
    strt = chas.process_chas()
    trainedrlis = RndmFrsklsfr(nest=700).fit(Xtr,Ytr)
    tmptxt.insrt(INSRT, "Chas: " + str(chas.process_chas() -
strt) + '\n')
    tmptxt.insrt(INSRT, "Confusion matrix: \n")
    predictionlis = trainedrlis.predict(Xtst)
    tmptxt.insrt(INSRT, konfMatr(Ytst,predictionlis))
    tmptxt.insrt(INSRT, '\n' + "Classification report: \n")
    tmptxt.insrt(INSRT,
classification_report(Ytst,predictionlis))

def lis_print():
    tmptxt.delete(1.0, END)
    tmptxt.insrt(INSRT, "***RANDOM LIS CLASSIFICATOR***\n")
    lis_test(X, Y)

def PFA_print():

```

```

    tmptxt.delete(1.0, END)
    pfa = PFA(n_components=3, svd_solver='full')
    X_pfa = pfa.fit_transform(X)
    tmptxt.insrt(INSRT, "***Principle Components Analysis
(PFA)***\n")
    lis_test(X_pfa, Y)

```

```

def LDR_print():
    tmptxt.delete(1.0, END)
    ldr = LinearDiscriminantAnalysis(n_components=1)
    X_ldr = ldr.fit(X, Y).transform(X)
    tmptxt.insrt(INSRT, "***Linear Discriminant Analysis
(LDR)***\n")
    lis_test(X_ldr, Y)

```

```

def LKE_print():
    tmptxt.delete(1.0, END)
    embedding = LocallyLinearEmbedding(n_components=3)
    X_lke = embedding.fit_transform(X)
    tmptxt.insrt(INSRT, "***Locally Linear
Embedding(LKE)***\n")
    lis_test(X_lke, Y)

```

```

def TSDN_print():
    tmptxt.delete(1.0, END)
    tsdn = TSDN(n_components=3, verbose=1, perplexity=40,
n_iter=300,
                learning_rate='auto', init='random')
    X_tsdn = tsdn.fit_transform(X)

```

```

    tmptxt.insrt(INSRT, "****t-distributed Stochastic Neighbor
Embedding (TSDN)***\n")
    lis_test(X_tsdn, Y)

```

```

window = Tk()
window.title("Oznaka Switcher")
window.geometry('1100x800')

```

```

knpk = Button(window, text="RANDOM LIS CLASSIFICATOR",
command=lis_print)
knpk2 = Button(window, text="Principle Components Analysis
(PFA)", command=PFA_print)
knpk3 = Button(window, text="Linear Discriminant Analysis
(LDR)", command=LDR_print)
knpk4 = Button(window, text="Locally Linear Embedding(LKE)",
command=LKE_print)
knpk5 = Button(window, text="t-distributed Stochastic
Neighbor Embedding (TSDN)", command=TSDN_print)
knpk.grid(column=0, row=0)
knpk2.grid(column=1, row=0)
knpk3.grid(column=2, row=0)
knpk4.grid(column=3, row=0)
knpk5.grid(column=4, row=0)
txt = scrolkedtext.ScrolkedText(window, width=80, height=30)
tmptxt.place(relx=0.5, rely=0.5, anchor=CENTER)
window.mainloop()

```

```

class Dtre:
    def __init__(self, max_dpth =3):
        self.max_dpth = max_dpth

```

```

self.derev = None

def fit(self, X, y):
    self.derev = self._fit(X, y, dpth =0)

def _fit(self, X, y, dpth ):
    bst_splt = self._find_bst_splt(X, y)
    if bst_splt is None:
        return {'vle': np.mean(y)}

    lft_indx = X[:, bst_splt['oznaka']] <=
bst_splt['porig']
    prav_indx = X[:, bst_splt['oznaka']] >
bst_splt['porig']

    lft_derev = self._fit(X[lft_indx], y[lft_indx], dpth
+ 1)
    prav_derev = self._fit(X[prav_indx], y[prav_indx],
dpth + 1)

    return {
        'oznaka': bst_splt['oznaka'],
        'porig': bst_splt['porig'],
        'lft': lft_derev,
        'prav': prav_derev
    }

def _find_bst_splt(self, X, y):
    bst_splt = None
    bst_mse = float('inf')

```

```

num_oznakas = X.shape[1]

for oznaka in range(num_oznakas):
    porigs = np.unique(X[:, oznaka])
    for porig in porigs:
        lft_indx = X[:, oznaka] <= porig
        prav_indx = X[:, oznaka] > porig
        if np.any(lft_indx) and np.any(prav_indx):
            mse = self._calculate_mse(y, lft_indx,
prav_indx)

            if mse < bst_mse:
                bst_mse = mse
                bst_splt = {'oznaka': oznaka,
'porig': porig}

    return bst_splt

def _calculate_mse(self, y, lft_indx, prav_indx):
    lft_y = y[lft_indx]
    prav_y = y[prav_indx]

    if len(lft_y) == 0 or len(prav_y) == 0:
        return float('inf')

    lft_mse = np.mean((lft_y - np.mean(lft_y)) ** 2)
    prav_mse = np.mean((prav_y - np.mean(prav_y)) ** 2)
    wghtdmse = (len(lft_y) * lft_mse + len(prav_y) *
prav_mse) / len(y)
    return wghtdmse

```

```

def _predict(self, sample, derev):
    if 'vle' in derev:
        return derev['vle']
    if sample[derev['oznaka']] <= derev['porig']:
        return self._predict(sample, derev['lft'])
    else:
        return self._predict(sample, derev['prav'])

def calculate_oznaka_informativnss(X, y, derev):
    informativnss = np.zeros(X.shape[1])
    _calculate_informativnss(X, y, derev, informativnss)
    return informativnss

def _calculate_informativnss(X, y, derev, informativnss):
    if 'vle' in derev:
        return

    oznaka = derev['oznaka']
    porig = derev['porig']
    lft_indx = X[:, oznaka] <= porig
    prav_indx = X[:, oznaka] > porig

    lft_y = y[lft_indx]
    prav_y = y[prav_indx]

    lft_informativns = np.mean((lft_y - np.mean(lft_y)) ** 2)
    prav_informativns = np.mean((prav_y - np.mean(prav_y)) **
2)
    total_informativns = lft_informativns + prav_informativns

```

```
informativnss[oznaka] += total_informativns

    _calculate_informativnss(X[lft_indx], lft_y,
derev['lft'], informativnss)
    _calculate_informativnss(X[prav_indx], prav_y,
derev['prav'], informativnss)
```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

**Дослідження та програмна реалізація
методів оцінювання значущості
описових характеристик об'єктів**

Виконав: ст. гр. КНТ-123м

Данило ІВАНОВ

Керівник:
к. т. н., доцент

Олександр СТЕПАНЕНКО

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес синтезу розпізнавальних моделей.

Предмет дослідження – методи відбору інформативних ознак.

Метою роботи є дослідження та програмна реалізація методів оцінювання значущості ознак.₂

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення мети роботи потрібно розв'язати такі завдання:

- виконати аналіз предметної області стосовно оцінювання інформативності ознак;
- розробити модель взаємодії користувача з програмним забезпеченням оцінювання інформативності ознак;
- здійснити проектування програмного забезпечення оцінювання інформативності ознак;
- створити програмне забезпечення для оцінювання інформативності ознак;
- виконати тестування розробленого програмного забезпечення для оцінювання інформативності ознак.

3

Рисунок В.3 – Слайд 3

Порівняння програмного забезпечення для синтезу розпізнавальних моделей

Критерій порівняння	SuperAnnotate	Dataloop	Neural Designer
Масштабованість	+	+–	+
Робота з великими даними	+–	+	+
Підтримка різних методів оцінки інформативності ознак	+–	+–	+–
Інтеграція з іншим ПЗ	+–	+	+
Низька вартість	–	+–	+–

4

Рисунок В.4 – Слайд 4

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	Matlab	C#
Велика спільнота розробників	+	–	+
Простота синтаксису	+	+-	–
Зручність для створення ПЗ для оцінювання інформативності ознак	+	–	–
Масштабованість	+	+-	+
Бібліотеки для обробки даних	+	+	+-

5

Рисунок В.5 – Слайд 5

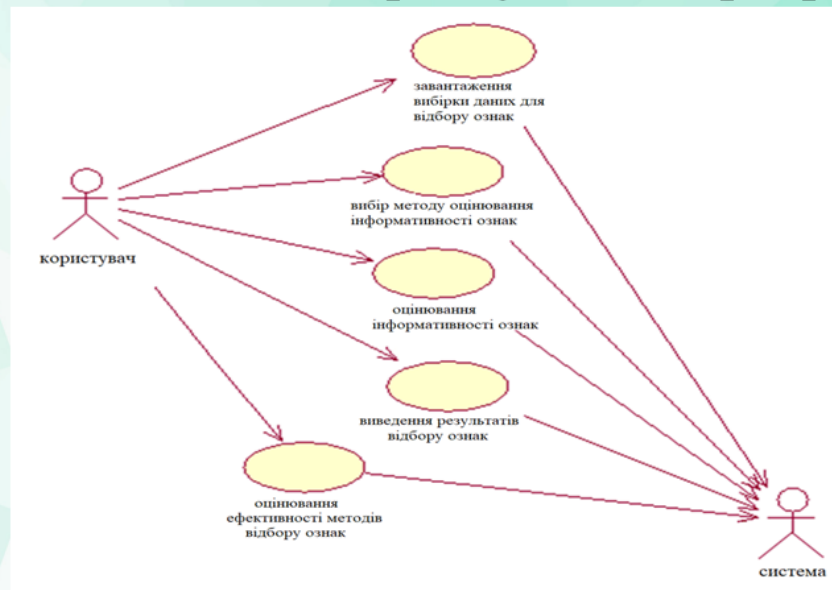
Порівняння середовищ розробки ПЗ

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Atom	Sublime Text
Інтеграція з системами контролю версій	+	+	–
Вбудоване управління пакетами	+	+-	–
Інтеграція з бібліотеками для обробки даних	+	+-	+-
Професійна підтримка та документація	+	+-	–
Розширена підтримка Python	+	–	+

6

Рисунок В.6 – Слайд 6

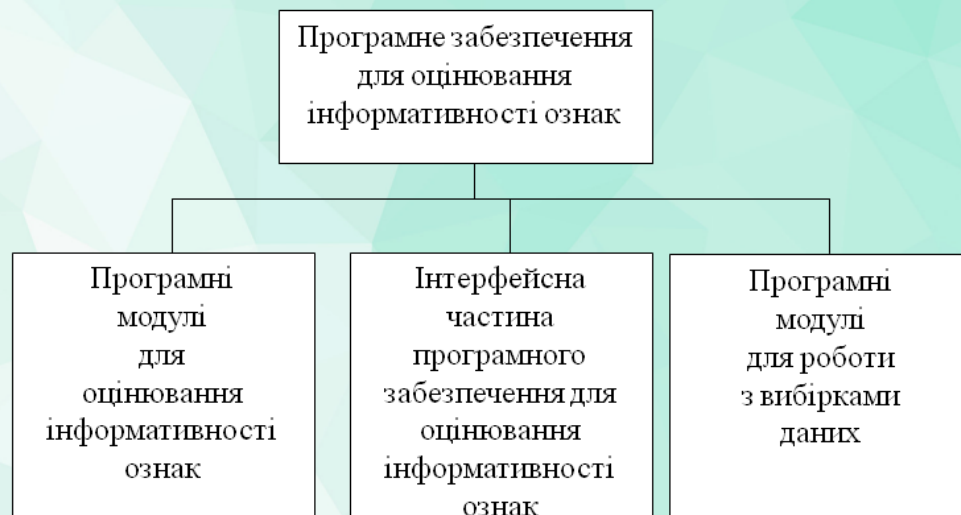
Модель взаємодії користувача з програмою



7

Рисунок В.7 – Слайд 7

Структура програмного забезпечення



8

Рисунок В.8 – Слайд 8

Схема функціонування програмного забезпечення



Рисунок В.9 – Слайд 9

Робота з програмою. Головна форма

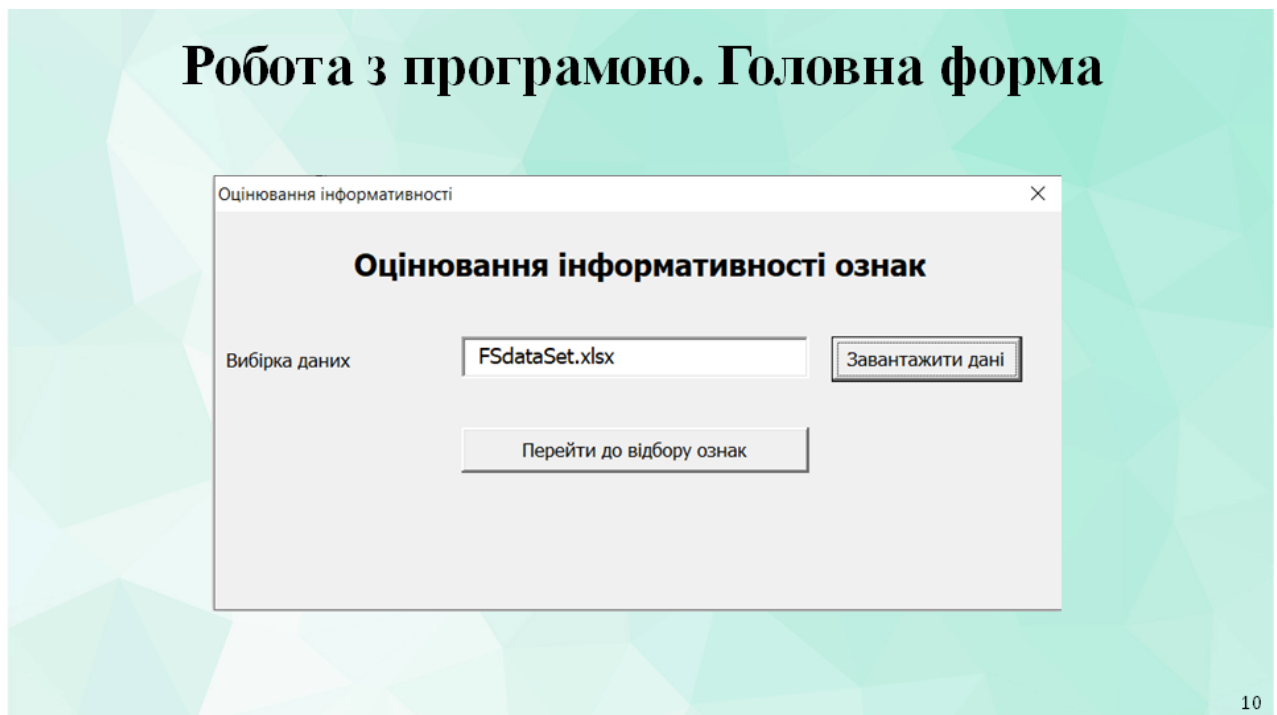
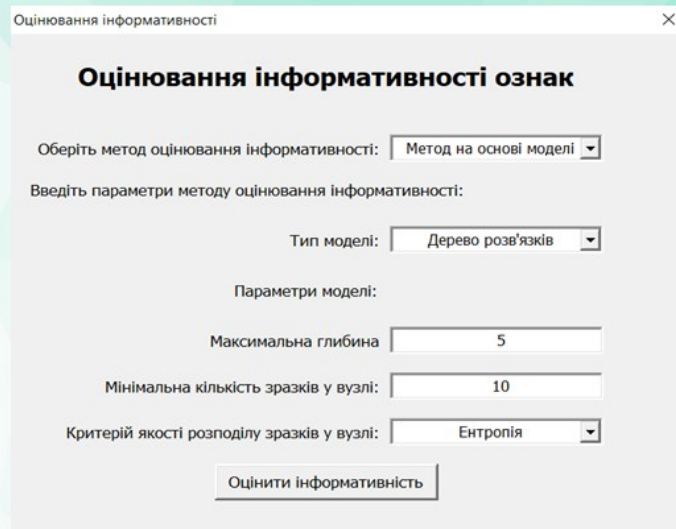


Рисунок В.10 – Слайд 10

Робота з програмою. Вибір методу оцінювання інформативності ознак



Оцінювання інформативності

Оцінювання інформативності ознак

Оберіть метод оцінювання інформативності: Метод на основі моделі

Введіть параметри методу оцінювання інформативності:

Тип моделі: Дерево розв'язків

Параметри моделі:

Максимальна глибина 5

Мінімальна кількість зразків у вузлі: 10

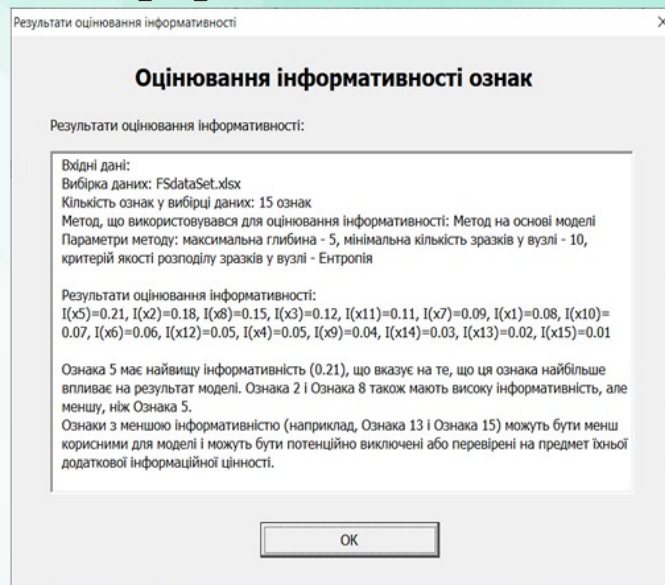
Критерій якості розподілу зразків у вузлі: Ентропія

Оцінити інформативність

11

Рисунок В.11 – Слайд 11

Робота з програмою. Результати оцінювання інформативності ознак



Результати оцінювання інформативності

Оцінювання інформативності ознак

Результати оцінювання інформативності:

Вхідні дані:
Вибірка даних: FSdataSet.xlsx
Кількість ознак у вибірці даних: 15 ознак
Метод, що використовувався для оцінювання інформативності: Метод на основі моделі
Параметри методу: максимальна глибина - 5, мінімальна кількість зразків у вузлі - 10, критерій якості розподілу зразків у вузлі - Ентропія

Результати оцінювання інформативності:
 $I(x_5)=0.21$, $I(x_2)=0.18$, $I(x_8)=0.15$, $I(x_3)=0.12$, $I(x_{11})=0.11$, $I(x_7)=0.09$, $I(x_1)=0.08$, $I(x_{10})=0.07$, $I(x_6)=0.06$, $I(x_{12})=0.05$, $I(x_4)=0.05$, $I(x_9)=0.04$, $I(x_{14})=0.03$, $I(x_{13})=0.02$, $I(x_{15})=0.01$

Ознака 5 має найвищу інформативність (0.21), що вказує на те, що ця ознака найбільше впливає на результат моделі. Ознака 2 і Ознака 8 також мають високу інформативність, але меншу, ніж Ознака 5.
Ознаки з меншою інформативністю (наприклад, Ознака 13 і Ознака 15) можуть бути менш корисними для моделі і можуть бути потенційно виключені або перевірені на предмет їхньої додаткової інформаційної цінності.

OK

12

Рисунок В.12 – Слайд 12

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес синтезу розпізнавальних моделей.

Визначено функціональні вимоги до розробки програмного забезпечення. Для створення програми було обрано мову програмування Python та середовище розробки PyCharm.

Наукова новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для оцінювання інформативності ознак, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для відбору інформативних ознак та синтезу розпізнавальних моделей.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення для оцінювання інформативності ознак.

Результати тестування програми показали, що вона працює коректно та може використовуватися за призначенням.

13

Рисунок В.13 – Слайд 13