Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти магістра

зі спеціальності 121 – Інженерія програмного забезпечення

На тему: Дослідження використання методів штучного інтелекту для формування навчально-професійної траєкторії старшокласників

Засвідчую, що в цій кваліфікаційній роботі немає запозичень із праць інших авторів без відповідних посилань.
Студент гр. ІПЗ-21м
_____ / М. Г. Ставцев /

Керівник
кваліфікаційної роботи _____ / Н. Х. Саітгареєв /

Економіко-
організаційна частина _____ / О.В. Шамрай /

Нормоконтроль _____ / Н. Х. Саітгареєв /

Завідувач кафедри _____ / А. М. Стрюк /

Кривий Ріг
2024

Криворізький національний університет

Факультет: Інформаційних технологій

Кафедра: Моделювання та програмного забезпечення

Ступінь вищої освіти: магістр

Спеціальність: 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри

_____ А. М. Стрюк

«___» _____ 20__ р.

ЗАВДАННЯ

на кваліфікаційну роботу

студенту групи ІПЗ-23-1м Ставцеву Максиму Геннадійовичу

1. Тема: Дослідження використання методів штучного інтелекту для формування навчально-професійної траєкторії старшокласників затверджено наказом по університету № 277с від «15» квітня 2024 р.
2. Термін подання студентом закінченої роботи: «01» грудня 2024р.
3. Вихідні дані по роботі: отримати порівняльний аналіз методів штучного інтелекту стосовно використання їх для формування навча навчально-професійної траєкторії старшокласників.
4. Зміст пояснювальної записки (перелік питань, що їх треба розробити): провести аналіз професійної області проблеми, порівняльна характеристику існуючих рішень, огляд та порівняння алгоритмів машинного навчання, придатних для вирішення задачі профорієнтації, розробка та тестування моделей класифікації, регресії та кластеризації, оцінка ефективності алгоритмів на основі встановлених метрик.
5. Перелік ілюстративного матеріалу: діаграми та схеми використання, основні алгоритми реалізованих моделей.

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	Огляд літератури за тематикою роботи	09.02.2024 – 15.03.2024
2	Проведення аналізу існуючих підходів та методів профорієнтації	16.03.2024 – 05.04.2024
3	Формулювання актуальності, мети та завдань роботи	06.04.2024 – 26.04.2024
4	Оформлення матеріалів першого розділу роботи	27.04.2024 – 17.05.2024
5	Вибір математичних моделей для класифікації, регресії та кластеризації	18.05.2024 – 07.06.2024
6	Підготовка та обробка даних для навчання моделей	08.06.2024 – 28.06.2024
7	Розробка програмної реалізації моделей машинного навчання	29.06.2024 – 27.07.2024
8	Тестування моделей, оцінка ефективності за критеріями точності та F1-міри	28.07.2024 – 20.08.2024
9	Оформлення матеріалів другого та третього розділів роботи	21.08.2024 – 10.09.2024
10	Аналіз результатів з точки зору наукової та практичної цінності	11.09.2024 – 01.10.2024
11	Узагальнення отриманих результатів та формування висновків	02.10.2024 – 22.10.2024
12	Остаточне оформлення пояснювальної записки	23.10.2024 – 12.11.2024
13	Підготовка ілюстративного матеріалу для захисту	13.11.2024 – 20.11.2024
14	Передзахист роботи	21.11.2024 – 28.11.2024

Дата видачі завдання: «09» лютого 2024 р.

Студент _____ / М. Г. Ставцев /

Керівник роботи _____ / Н. Х. Саітгареєв /

РЕФЕРАТ

ШТУЧНИЙ ІНТЕЛЕКТ, ПРОФОРІЄНТАЦІЯ, НАВЧАЛЬНО-ПРОФЕСІЙНА ТРАЄКТОРІЯ, АНАЛІЗ ОСОБИСТИХ ДАНИХ УЧНІВ, МЕТОДИ МАШИННОГО НАВЧАННЯ

Пояснювальна записка: 104 с., 5 табл., 23 рис., 1 дод., 18 джерел.

Метою кваліфікаційної роботи є дослідження та реалізація методів машинного навчання для аналізу індивідуальних характеристик старшокласників з метою формування рекомендацій щодо вибору їхньої навчально-професійної траєкторії.

У роботі проведено аналіз існуючих підходів до профорієнтації, зокрема інструментів, що базуються на традиційних методах психологічного тестування та алгоритмах штучного інтелекту. Виконано порівняння сучасних методів машинного навчання, таких як класифікація, регресія та кластеризація, з урахуванням специфіки задачі профорієнтації.

Виконано розгортання моделей на базі API, що забезпечує інтеграцію розроблених рішень у існуючі системи профорієнтації. Результати роботи можуть бути використані у школах та позашкільних закладах для автоматизації процесу профорієнтації та надання персоналізованих рекомендацій.

ABSTRACT

ARTIFICIAL INTELLIGENCE, CAREER GUIDANCE, EDUCATIONAL AND PROFESSIONAL TRAJECTORY, ANALYSIS OF STUDENTS' PERSONAL DATA, MACHINE LEARNING METHODS

Explanatory note: 104 p., 5 tables, 23 figures, 1 appendices, 18 sources.

The purpose of the qualification work is to study and implement machine learning methods for analyzing individual characteristics of high school students in order to form recommendations for choosing their educational and professional trajectory.

The work analyzes existing approaches to career guidance, in particular tools based on traditional psychological testing methods and artificial intelligence algorithms. A comparison of modern machine learning methods, such as classification, regression and clustering, is made, taking into account the specifics of the career guidance task.

API-based models have been deployed, which ensures the integration of the developed solutions into existing career guidance systems. The results of the work can be used in schools and extracurricular institutions to automate the career guidance process and provide personalized recommendations..

ЗМІСТ

ВСТУП 8
1 АНАЛІТИЧНИЙ ОГЛЯД ТА ВИЗНАЧЕННЯ ОСНОВНИХ НАПРЯМІВ ДОСЛІДЖЕННЯ 10
1.1 Аналіз теоретичних основ та проблеми вибору професійної траєкторії старшокласниками 10
1.2 Штучний інтелект і використання методів машинного навчання у галузі профорієнтації 13
1.3 Дослідження особливості предметної галузі 15
1.4 Аналіз існуючих досліджень з обраної теми 16
1.5 Актуальність задачі дослідження методів для створення інтелектуальної системи формування навчально-професійної траєкторі 18
1.6 Вимоги до дослідження методів машинного навчання для формування навчально-професійних траєкторій старшокласників 19
2 ВІДОМОСТІ ПРО ПРЕДМЕТ (ОБ'ЄКТ) ДОСЛІДЖЕННЯ 22
2.1 Формулювання предмета і об’єкта дослідження 22
2.2 Обґрунтування і вибір методів дослідження 24
2.3 Розробка математичних моделей, постановка задачі моделювання 26
2.4 Підготовка та обробка даних 28
2.5 Функціональна схема та основні алгоритми дослідження 30
2.6 Критерії оцінки ефективності методів 35
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ 37
3.1 Вибір інструментарію для реалізації методів машинного навчання 37
3.2 Підготовка даних для дослідження 40
3.3 Реалізація алгоритмів класифікації 43
3.4 Створення API 47

3.5 Керівництво для користувача по роботі з API 49
4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ ТА АНАЛІЗ ЕФЕКТИВНОСТІ 51
4.1 Методика проведення дослідження 51
4.2 Формування залежностей між параметрами об’єкту дослідження професійної області 53
4.3 Метрики оцінювання результатів дослідження 54
4.4 Формулювання отриманих результатів дослідження 56
4.5 Інтерпритація отриманих результатів з точки зору наукової та технічної цінності 60
5 АНАЛІЗ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ІННОВАЦІЇ 64
5.1 Розрахунок собівартості програмної інновації 64
5.2 Розрахунок ефективності впровадження програмної інновації 68
ВИСНОВКИ 71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 74
ДОДАТОК А 76

ВСТУП

Вибір професійного шляху є одним із найважливіших рішень у житті людини, що визначає майбутні можливості та кар'єрний розвиток. На етапі старшої школи цей вибір особливо складний, оскільки потребує не лише розуміння власних здібностей, інтересів та особистісних якостей, але й врахування тенденцій ринку праці та професійних перспектив. У сучасному світі, де швидкість розвитку технологій і зміна вимог до професій збільшується, старшокласники стикаються з численними викликами у прийнятті обґрунтованого рішення. Традиційні підходи до профорієнтації часто виявляються недостатньо ефективними, оскільки не враховують динамічні зміни ринку та індивідуальні особливості кожного учня. Ця проблема залишається гострою, потребуючи нових методів вирішення, заснованих на сучасних технологіях.

На глобальному рівні для вирішення проблем профорієнтації все більше використовуються технології штучного інтелекту (ШІ), зокрема машинне навчання. У багатьох країнах відзначається зростання інтересу до персоналізованого підходу в освіті, що базується на аналізі індивідуальних характеристик учнів. Методи машинного навчання дозволяють обробляти великі обсяги даних про інтереси, академічні досягнення та психологічні характеристики учнів, сприяючи формуванню точних рекомендацій для вибору професії. У цьому контексті дослідження можливостей штучного інтелекту в підтримці профорієнтації старшокласників є надзвичайно актуальним.

Дана робота спрямована на дослідження застосування методів машинного навчання для аналізу даних старшокласників з метою формування оптимальних навчально-професійних траєкторій. Об’єктом дослідження є процес формування цієї траєкторії, що охоплює академічні досягнення, психологічні характеристики та інтереси учнів. Предметом дослідження є конкретні методи машинного навчання (класифікація, регресія, кластеризація), які дозволяють виявляти

закономірності в даних учнів і забезпечувати точні та індивідуальні рекомендації.

Метою даного дослідження є розробка та перевірка ефективності методів машинного навчання для персоналізованого профорієнтаційного консультування старшокласників. Це передбачає створення математичних моделей, здатних аналізувати індивідуальні дані і передбачати майбутні професійні траєкторії на основі інтересів і здібностей учнів, а також тенденцій на ринку праці. Результати дослідження можуть бути використані в школах, профорієнтаційних центрах та інших освітніх установах для підвищення ефективності профорієнтаційної роботи.

Для досягнення поставленої мети застосовуються методи машинного навчання, такі як класифікація (рішення дерев, метод опорних векторів), регресія та кластеризація. Класифікаційні методи дозволяють ідентифікувати можливі навчальні напрями для учнів, регресійні моделі — прогнозувати кар'єрні перспективи, а кластеризація — групувати учнів із подібними характеристиками для індивідуальних рекомендацій. Ці методи дозволяють використовувати великий обсяг даних для формування рекомендацій, що враховують індивідуальні потреби і здібності учнів.

Дане дослідження пов’язане з низкою робіт, присвячених застосуванню технологій машинного навчання для автоматизації профорієнтаційних процесів. Впровадження результатів дослідження сприятиме розвитку інноваційних методів освітньої підтримки, що забезпечують точність і гнучкість у профорієнтації, необхідну для сучасного ринку праці.

1 АНАЛІТИЧНИЙ ОГЛЯД ТА ВИЗНАЧЕННЯ ОСНОВНИХ НАПРЯМІВ ДОСЛІДЖЕННЯ

1.1 Аналіз теоретичних основ та проблеми вибору професійної траєкторії старшокласниками

Вибір професії є надзвичайно важливим і водночас складним процесом у житті кожної людини. Саме тому профорієнтація, як наука і практика, отримала багато уваги серед науковців, які намагаються зрозуміти, які фактори впливають на вибір професійної траєкторії, та як допомогти молоді зробити правильний вибір. Сучасні теорії профорієнтації висувають різні підходи, кожен з яких має свою логіку та пояснює процес ухвалення рішення по-своєму.

Однією з найбільш відомих теорій є теорія особистісних характеристик і професійної відповідності, розроблена Джоном Голландом. Він стверджував, що людина вибирає професію, яка відповідає її особистісним характеристикам. Голланд виділив шість основних типів особистостей, кожен з яких має певні професійні преференції. Наприклад, реалістичний тип тяжіє до роботи з технікою та реальними об'єктами, у той час як соціальний тип орієнтований на роботу з людьми. На думку Голланда, чим більше професійне середовище відповідає особистісному типу людини, тим більше вона буде задоволена роботою і досягне успіху [1].

Ще одна популярна теорія — теорія розвитку кар'єри, запропонована Дональдом Супером. Вона розглядає кар'єру як динамічний процес, що проходить через різні стадії розвитку, починаючи від дитинства і аж до зрілого віку. Супер вважав, що на кожному етапі життя людина стикається з певними завданнями та викликами, які формують її професійний вибір. На етапі старшої школи, за Супером, особливо важливо зрозуміти свої інтереси, навички та життєві пріоритети, оскільки це вплине на вибір майбутнього професійного шляху.

Теорія соціально-когнітивної кар'єри Лента, Брауна і Хакетта підкреслює, що професійний вибір формується не лише на основі особистісних характеристик, але й через взаємодію з навколишнім середовищем. Люди часто орієнтуються на свої переконання, очікування та думки про свої здібності, які можуть бути результатом досвіду, впливу соціального середовища, або ж внутрішніх переконань. Наприклад, якщо учень отримує підтримку від батьків або вчителів у розвитку своїх здібностей до науки, це може сприяти тому, що він вибере професію, пов'язану з науковими дослідженнями [2].

Ще одна цікава теорія — теорія компромісу та компромісу Лінди Готтфредсон, яка пояснює, як молоді люди обирають професію, керуючись своїми очікуваннями та обмеженнями, що накладає суспільство. Готтфредсон стверджує, що вибір професії часто є компромісом між бажаннями та можливостями. Під впливом суспільства, старшокласники можуть відмовлятися від певних професійних шляхів через соціальні обмеження або стереотипи, навіть якщо їм насправді було б цікаво розвиватися в цих напрямках.

Таким чином, сучасні теорії профорієнтації показують важливість індивідуального підходу до вибору професії, оскільки багато факторів, таких як особисті якості, вплив оточення та соціальні обмеження, відіграють свою роль у цьому процесі. Однак, попри різноманітність підходів, на практиці вибір професії для старшокласників все одно залишається непростим завданням через низку труднощів.

Процес вибору професії для старшокласників ускладнюється низкою проблем, які часто зумовлені як об'єктивними, так і суб'єктивними факторами. Однією з головних проблем є недостатня обізнаність про професії та ринок праці. Багато учнів просто не знають про можливості, які існують на сучасному ринку праці, особливо про нові професії, які з'явилися в результаті розвитку технологій. Інформація, яку вони отримують у школі або від батьків, може бути застарілою або обмеженою. Як результат, старшокласники не мають повної картини про те, які вимоги та перспективи мають професії, що часто стає причиною невдалого вибору.

Ще однією важливою проблемою є невизначеність щодо власних інтересів та здібностей. У підлітковому віці багато старшокласників ще не повністю усвідомлюють, що їм дійсно цікаво, і якими є їхні сильні сторони. Відсутність глибокого розуміння власних схильностей та здібностей може призвести до того, що учні обирають професії під впливом зовнішніх факторів, не маючи впевненості в тому, що саме вони дійсно хочуть робити в житті.

Важливу роль у виборі професії відіграє соціальний тиск та стереотипи. Багато старшокласників зіштовхуються з очікуваннями батьків, вчителів або друзів, які можуть схиляти їх до вибору певних професій або уникання інших через суспільні стереотипи. Наприклад, існують стереотипи щодо певних професій, як-от технічних спеціальностей, які традиційно вважаються "чоловічими", що може зупинити дівчат від вибору цих напрямків, навіть якщо вони мають здібності до них.

Також варто зазначити, що старшокласники часто відчувають страх перед змінами та невпевненість у майбутньому. Швидкі зміни на ринку праці, що обумовлені розвитком технологій, створюють відчуття нестабільності. Молоді люди побоюються, що професія, яку вони оберуть зараз, через кілька років може стати незатребуваною. Це додає додаткового стресу при ухваленні рішення.

Зрештою, варто відзначити, що у багатьох школах відсутня ефективна система підтримки, яка б допомогла учням у виборі професії. Стандартні профорієнтаційні тести, що використовуються у школах, часто не враховують індивідуальних особливостей учнів і не пропонують актуальної інформації про сучасний ринок праці. Це робить процес ухвалення рішення ще складнішим, оскільки старшокласники не отримують необхідної підтримки для обґрунтованого вибору.

Отже, вибір професії для старшокласників є складним багатофакторним процесом, який вимагає врахування індивідуальних якостей, знань про ринок праці та зовнішніх факторів, як-от соціальний тиск. Сучасні теорії профорієнтації надають рамкові підходи для розуміння цих складнощів, але на

практиці виникає необхідність у нових підходах, що здатні персоналізувати процес і врахувати унікальні особливості кожного учня.

1.2 Штучний інтелект і використання методів машинного навчання у галузі профорієнтації

Продовжуючи аналіз проблем вибору професії, варто звернутися до сучасних технологій, які можуть допомогти старшокласникам у цьому складному процесі. Однією з таких технологій є штучний інтелект (ШІ) — сукупність методів, що дозволяють комп'ютерам імітувати процеси мислення людини для вирішення складних завдань. ШІ стає дедалі більш актуальним інструментом у сфері профорієнтації, оскільки він здатний аналізувати великі обсяги даних, знаходити в них закономірності та надавати рекомендації, враховуючи індивідуальні особливості кожного учня.

Суть ШІ полягає у здатності не лише виконувати заздалегідь визначені інструкції, але й навчатися на основі попереднього досвіду, адаптуючи свої алгоритми до нових умов. Це дозволяє йому аналізувати інформацію та приймати рішення набагато гнучкіше і точніше, ніж традиційні алгоритми. На відміну від класичних програм, які виконують певні завдання за чітко визначеними правилами, ШІ може розвиватися і вдосконалювати свої методи в процесі навчання, що особливо важливо для вирішення завдань у таких динамічних сферах, як ринок праці та профорієнтація.

Штучний інтелект — це одна з найсучасніших і найперспективніших технологій, яка швидко проникає в наше життя та стає невід'ємною частиною багатьох сфер діяльності. Основна ідея ШІ полягає в тому, щоб імітувати здатність людини мислити та приймати рішення. Класичні комп'ютерні програми, створені людиною, діють за жорстко заданими алгоритмами і мають обмежені можливості для адаптації. ШІ, натомість, здатен навчатися та удосконалювати свої алгоритми на основі отриманих даних. Це дозволяє йому

знаходити рішення для складних завдань, які важко або навіть неможливо передбачити заздалегідь [3].

Однією з ключових галузей, у якій використовується ШІ, є машинне навчання — підхід, що дозволяє комп’ютерам самостійно вивчати закономірності в даних і приймати рішення на їхній основі. Замість того щоб програмувати алгоритм для кожного можливого сценарію, машинне навчання дозволяє системі "навчатися" на основі прикладів і коригувати свої рішення в процесі роботи. Таким чином, комп’ютер стає здатним до самонавчання і вдосконалення, що відкриває великі можливості для створення інтелектуальних систем у різних галузях, включно з профорієнтацією.

ШІ не просто обробляє дані за певними шаблонами — він може аналізувати їх та знаходити нові, раніше невідомі зв’язки між даними. Наприклад, у сфері освіти ШІ може допомогти старшокласникам вибрати професійний шлях, аналізуючи їхні інтереси та здібності й порівнюючи їх з вимогами різних професій. Завдяки методам класифікації, регресії та кластеризації, які допомагають ШІ аналізувати великі масиви даних і будувати прогнози, системи на основі ШІ можуть стати ефективними інструментами для вирішення завдань профорієнтації. Це дозволяє не лише враховувати академічні досягнення учнів, але й аналізувати їхні особистісні особливості, інтереси та тенденції ринку праці [4].

Отже, ШІ — це не просто технологія, а потужний інструмент, здатний значно розширити можливості людини. Імітуючи людське мислення, ШІ допомагає у прийнятті обґрунтованих рішень, аналізуючи дані на значно вищому рівні, ніж це під силу звичайним комп’ютерним алгоритмам. У профорієнтації ШІ може стати ключовим засобом для забезпечення точності та індивідуалізації рекомендацій, що робить процес вибору професійної траєкторії більш ефективним і відповідним для кожного учня.

1.3 Дослідження особливості предметної галузі

Дослідження особливостей предметної галузі у контексті застосування методів машинного навчання для формування навчально-професійних траєкторій старшокласників потребує всебічного аналізу кількох ключових аспектів: освітньої системи, психологічних характеристик старшокласників, тенденцій ринку праці та технологічних можливостей сучасного машинного навчання.

Насамперед, предметна галузь охоплює розуміння структури та особливостей освітньої системи, включаючи навчальні програми, спеціалізації та доступні професійні шляхи. Важливим є аналіз того, як освітні установи підготовляють учнів до вибору кар'єри, які інструменти та підходи використовуються для кар'єрного консультування, а також які виклики стоять перед учнями при прийнятті професійного рішення. Серед основних інструментів, що використовуються для кар'єрного консультування, є профорієнтаційні тести, опитування інтересів та здібностей, а також різні освітні портали, що допомагають учням оцінити свої нахили до певних професійних областей.

Другим важливим аспектом є психологічні особливості старшокласників, які мають великий вплив на їхні рішення щодо вибору професії. Цей аспект включає вивчення мотивації, інтересів, особистісних якостей та академічних здібностей учнів. Розуміння цих характеристик є критичним, адже на основі індивідуальних відмінностей можна точніше прогнозувати навчальні та професійні перспективи учнів. У цьому контексті досліджуються такі психологічні характеристики, як емоційний інтелект, рішучість і відкритість до нових досвідів. Академічні здібності, такі як математична схильність або вербальні навички, також є важливими, оскільки вони можуть вплинути на придатність учнів до певних наукових чи гуманітарних напрямів. Детальне вивчення цих аспектів дозволить точніше визначити навчально-професійні рекомендації, адаптовані до індивідуальних потреб кожного учня.

Третім аспектом, який відіграє вирішальну роль у створенні ефективних рекомендацій, є аналіз ринку праці. Вивчення поточних і майбутніх тенденцій у зайнятості, найбільш затребуваних навичок, а також економічних та технологічних змін у різних індустріях допомагає сформувати професійні рекомендації, які будуть не лише цікавими для старшокласників, а й перспективними з точки зору подальшої зайнятості. Це дозволяє забезпечити актуальність рекомендацій і допомогти учням вибрати професії, що відповідають потребам ринку праці та мають потенціал для розвитку.

Останнім, але не менш важливим аспектом є технологічні можливості сучасних методів машинного навчання, які можуть значно покращити якість кар'єрного консультування. Цей аспект включає оцінку потенціалу та обмежень таких технологій, їх здатність обробляти великі обсяги даних та забезпечувати високу точність прогнозування. Важливо дослідити, які саме методи машинного навчання підходять для аналізу даних про учнів, як забезпечити конфіденційність і захист інформації, а також як інтегрувати ці технології в освітні процеси, дотримуючись етичних норм.

Таким чином, ці чотири аспекти — освітня система, психологічні характеристики учнів, ринок праці та можливості машинного навчання — утворюють комплексну основу для аналізу предметної галузі та розробки інноваційних інструментів профорієнтаційної підтримки. Успішне поєднання цих елементів має великий потенціал для підвищення ефективності кар'єрного орієнтування, допомагаючи старшокласникам обрати професії, які принесуть їм не лише професійний успіх, а й особистісне задоволення.

1.4 Аналіз існуючих досліджень з обраної теми

Найбільш підходящим, для аналізу вже існуючих досліджень з обраної тематики, є дослідження «ПРОФЕСІЙНЕ САМОВИЗНАЧЕННЯ СТАРШОКЛАСНИКІВ ЯК ПЕРЕДУМОВА УСПІШНОЇ ПРОФЕСІЙНОЇ

САМОРЕАЛІЗАЦІЇ ТА КАР’ЄРНОГО РОЗВИТКУ» за авторства Чечко І.І, члена Чернігівського обласного центру зайнятості [5].

Дане дослідження зосереджене на соціально-психологічному вимірі підготовки старшокласників до вступу у вищі навчальні заклади. Воно вивчає, як старшокласники роблять вибір своєї майбутньої освіти і професії, розглядаючи це як процес професійного самовизначення. Автори дослідження аналізують різні соціальні та психологічні фактори, які можуть впливати на рішення старшокласників, включаючи їх особистісні характеристики, сімейне оточення, шкільне середовище та вплив однолітків.

До сильних сторін цього дослідження слід віднести:

− всебічний аналіз, адже стаття надає детальний огляд взаємодії соціальних, психологічних і ринкових динамік, які впливають на кар'єрні вибори студентів;

− теоретичну глибину дослідження, оскільки воно розглядає кілька теоретичних рамок, надаючи міцну академічну основу обговоренню професійного самовизначення.

Однак, слабкими сторонами цього дослідження є:

− відсутність технологічної інтеграції, оскільки дослідження не включає сучасні технологічні інструменти, такі як машинне навчання, які могли б підвищити точність і персоналізацію професійного наставництва;

− обмежений обсяг практичних застосувань. Хоча дослідження міцне у теорії, йому невистачає опису більш прямих застосувань або прикладів того, як ці теорії можуть бути практично втілені в освітніх системах для допомоги в прийнятті кар'єрних рішень.

Підсумовуючи, хоча наукова стаття надає цінні інсайти щодо факторів, які впливають на кар'єрні вибори серед старшокласників, інтеграція машинного навчання та надання більш практичної допомоги могли б зробити її більш актуальною для сучасних освітніх потреб. Цей підхід зміг би згладити розрив між теоретичними знаннями та практичним застосуванням, що є необхідним для

вирішення динамічних та різноманітних потреб студентів у швидко змінюваному світі.

1.5 Актуальність задачі дослідження методів для створення інтелектуальної системи формування навчально-професійної траєкторі

Сучасна освітня система все більше потребує персоналізованих рішень, особливо в сфері профорієнтації старшокласників, які стоять перед вибором майбутнього професійного напряму. Стандартні методи, що застосовуються для профорієнтації, часто обмежені загальними рекомендаціями та не враховують індивідуальні якості, навички та досягнення кожного учня. У зв’язку з цим виникає потреба у створенні інтелектуальної системи, яка б надавала точні та персоналізовані рекомендації щодо вибору професійної траєкторії.

Старшокласники нерідко стикаються з труднощами у визначенні своїх сильних сторін та інтересів, що ускладнює вибір професії. Стандартні тести або загальні консультації не враховують унікальні риси особистості кожного учня, що робить процес профорієнтації недостатньо точним. Інтелектуальна система, яка використовує сучасні методи машинного навчання, може глибоко аналізувати дані про особисті якості, академічні досягнення та схильності учнів і надавати рекомендації, що відповідають їхнім індивідуальним особливостям. Такий підхід забезпечує підвищену точність у виборі напряму навчання та роботи.

Серед методів машинного навчання, які можуть розглядатися для використання у системах профорієнтації, важливими є:

- класифікація - метод, що дозволяє віднести учня до певної професійної категорії, спираючись на його особистісні якості та успішність у різних предметах. Дослідження класифікаційних алгоритмів, таких як дерева рішень та метод опорних векторів, дасть змогу визначити їхню придатність для розв’язання профорієнтаційних завдань;

– регресія - використовується для прогнозування кількісних показників, які можуть бути корисними при визначенні відповідності учня певному професійному напряму;

– кластеризація - дає можливість групувати учнів з подібними характеристиками для формування більш точних і диференційованих рекомендацій.

Застосування цих методів дозволяє обробляти великі обсяги даних, виявляти закономірності та забезпечувати точні результати, що є важливим для формування персоналізованих рекомендацій у процесі профорієнтації. Аналіз та порівняння цих методів у межах дослідження допоможуть визначити їхню ефективність та адаптивність до індивідуальних потреб учнів.

Персоналізований підхід до профорієнтації має велике значення для підвищення точності рекомендацій. Інтелектуальна система, здатна враховувати унікальні якості кожного учня, такі як успішність у певних предметах, природні схильності, рівень мотивації та інші особистісні характеристики, забезпечує створення рекомендацій, що максимально відповідають потенціалу і потребам учня. Це знижує ризик неправильного вибору професії та підвищує імовірність майбутнього професійного задоволення.

Таким чином, дослідження методів машинного навчання для застосування у профорієнтаційних системах є надзвичайно актуальним. Визначення найбільш підходящих алгоритмів дозволяє створити надійну основу для побудови ефективної інтелектуальної системи, яка допомагатиме учням у виборі професійного напряму, враховуючи їхні індивідуальні якості.

1.6 Вимоги до дослідження методів машинного навчання для формування навчально-професійних траєкторій старшокласників

Перед проведенням дослідження, яке зосереджена на використанні методів машинного навчання для формування навчально-професійних траєкторій старшокласників, важливо визначити чіткі вимоги. Це допоможе забезпечити

ефективність дослідження та розробку корисних інструментів. До основниих вимог дослідження слід віднести:

а) збір даних:

1) забезпечити доступ до достатньої кількості релевантних даних про старшокласників, включаючи їх академічні результати, психологічні профілі, інтереси та кар'єрні уподобання;

2) зібрати дані з різних географічних регіонів та соціально-економічних груп для репрезентативності та уникнення упереджень;

б) обробка даних:

1) визначити процедури для очищення даних, нормалізації та обробки пропущених значень для забезпечення якості і коректності вхідних даних для аналізу;

2) забезпечити захист персональних даних відповідно до норм конфіденційності та законодавства про захист даних;

в) методологія машинного навчання:

1) оптимізувати вибір алгоритмів машинного навчання відповідно до специфіки задачі профорієнтації. Зокрема, передбачити використання методів класифікації для формування рекомендацій щодо професій, регресії для оцінки потенційної успішності учнів та кластеризації для групування старшокласників із подібними профілями;

2) розробити та навчити моделі, що використовують зібрані дані для формування рекомендацій професійних шляхів, забезпечуючи точність у прогнозуванні;

3) забезпечити регулярне тестування та валідацію моделей для оцінки їхньої точності та ефективності, що дозволить уникнути помилок у профорієнтаційних рекомендаціях;

г) етичні міркування:

1) впровадити заходи для забезпечення відсутності упереджень в алгоритмах, щоб не відтворювати існуючі соціальні нерівності та уникнути

дискримінації за статтю, соціально-економічним статусом або іншими характеристиками;

2) забезпечити прозорість процесів машинного навчання, щоб висновки моделей були зрозумілими для користувачів і всіх зацікавлених сторін;

д) впровадження та моніторинг:

1) розробити стратегії інтеграції розроблених інструментів у шкільні та післяшкільні освітні програми, щоб система могла стати частиною навчального процесу;

2) встановити процедури моніторингу використання системи та збору зворотного зв’язку з метою подальшого удосконалення технологій і підвищення якості рекомендацій.

2 ВІДОМОСТІ ПРО ПРЕДМЕТ (ОБ'ЄКТ) ДОСЛІДЖЕННЯ

2.1 Формулювання предмета і об’єкта дослідження

Формування навчально-професійної траєкторії старшокласників є багатогранним процесом, який поєднує аналіз індивідуальних особливостей учнів з використанням сучасних технологій. У цьому дослідженні особлива увага приділяється використанню методів машинного навчання для побудови рекомендацій, які враховують унікальні риси кожного учня. Для чіткого розуміння дослідницького підходу необхідно визначити предмет і об’єкт дослідження, які формують основу для проведення роботи.

Предмет дослідження визначає конкретні аспекти чи властивості об’єкта, які досліджуються для вирішення проблеми чи досягнення цілей роботи, тому предметом нашого дослідження є процес формування навчально-професійної траєкторії старшокласників. Цей процес передбачає аналіз даних про учнів, створення моделей на основі методів машинного навчання та побудову рекомендацій, які допомагають старшокласникам зробити обґрунтований вибір майбутнього професійного напряму. Формування професійної траєкторії охоплює як аналіз академічних досягнень і особистісних характеристик учнів, так і їх інтересів та кар’єрних уподобань.

Об’єкт дослідження — це частина реальності, явище чи процес, який вивчається для досягнення наукових результатів. Він є тією базою, що піддається аналізу. Відповідно в нашому дослідженні об’єктом є індивідуальні якості, навички, досягнення та інтереси старшокласників, які слугують основою для формування рекомендацій. Ці характеристики є ключовими даними для навчання моделей машинного навчання, які виявляють закономірності та створюють персоналізовані рекомендації. До основних аспектів, що аналізуються в межах об’єкта дослідження, належать:

- академічні досягнення: оцінки, результати олімпіад, прогрес у навчанні, які свідчать про схильності до певних предметів чи галузей знань;

- навички та здібності: технічні та м’які навички, когнітивні здібності, які визначають потенціал учня у певних професійних сферах;
- особистісні характеристики: психологічні риси (рішучість, відкритість до нового, емоційний інтелект), які впливають на вибір професії;
- інтереси та кар’єрні уподобання: захоплення, хобі та кар’єрні цілі, що формуються під впливом сім’ї, середовища чи самоаналізу;
- соціокультурні фактори: соціально-економічне середовище та регіональні особливості, які можуть визначати доступність певних професій чи навчальних напрямів.

Предмет і об’єкт дослідження тісно пов’язані між собою. Процес формування навчально-професійної траєкторії (предмет) базується на аналізі даних про учнів (об’єкт). Саме індивідуальні характеристики старшокласників є основою для створення точних та персоналізованих рекомендацій. Методи машинного навчання дозволяють автоматизувати цей процес, забезпечуючи високу точність та адаптивність рекомендацій.

У межах дослідження використання методів машинного навчання для аналізу об’єкта (індивідуальних характеристик учнів) та вирішення завдання формування професійної траєкторії (предмету) є ключовим етапом. Зокрема, такі алгоритми, як класифікація, регресія та кластеризація, сприяють ефективному опрацюванню даних та підвищують якість рекомендацій.

Таким чином, предмет дослідження — це процес формування навчально-професійної траєкторії, що враховує особливості учнів, а об’єктом є характеристики цих учнів, які формують основу для аналізу та побудови рекомендацій. Їхнє дослідження забезпечує теоретичну та практичну базу для розробки сучасних інтелектуальних систем, які сприятимуть якісному вирішенню завдань профорієнтації.

2.2 Обґрунтування і вибір методів дослідження

Продовжуючи аналіз предметної галузі, необхідно перейти до розгляду методів машинного навчання, які є основою для формування навчально-професійних траєкторій старшокласників. Оскільки завдання профорієнтації потребує не лише аналізу індивідуальних характеристик учнів, але й точного прогнозування та класифікації, важливо вибрати оптимальні алгоритми, здатні враховувати багатовимірність і складність вхідних даних. Розуміння можливостей кожного методу та його доцільності у вирішенні поставленого завдання дозволяє забезпечити ефективність дослідження та формування персоналізованих рекомендацій.

Класифікація — це метод машинного навчання, який використовується для прогнозування категорії або класу об'єкта на основі вхідних даних. Алгоритми класифікації працюють у просторі ознак, який визначається характеристиками досліджуємого об’єкту. До основних алгоритмів класифікації відносятся:

- дерева рішень (Decision Trees) є інтерпретованим методом, який будує ієрархію правил на основі значень вхідних ознак. Алгоритм вибирає розбиття даних, яке мінімізує ентропію (або іншу функцію втрат) у кожному вузлі дерева. Цей підхід є ефективним для створення рекомендацій, які зрозумілі для користувача. Наприклад, якщо учень демонструє високі оцінки з математики та низькі з гуманітарних дисциплін, дерево рішень може віднести його до технічних спеціальностей;
- метод опорних векторів (SVM) шукає гіперплощину, яка найкраще розділяє дані у просторі ознак. SVM оптимізує маржинальний простір між класами, що дозволяє зменшити ймовірність помилкової класифікації. Цей метод є особливо ефективним для даних із високою розмірністю, що відповідає задачам профорієнтації, де враховується велика кількість факторів;
- нейронні мережі дозволяють моделювати нелінійні залежності між характеристиками учня та його професійною орієнтацією. Вони використовують багатошарову архітектуру, яка складається з вхідного, прихованих та вихідного шарів. Кожен нейрон виконує обчислення за допомогою активаційної функції,

наприклад, ReLU або сигмоїдної. Нейронні мережі здатні обробляти різнорідні дані, такі як оцінки, текстові відповіді чи психологічні опитування, що робить їх корисними у нашій темі.

Регресія використовується для прогнозування числових значень на основі заданих вхідних ознак. У завданні профорієнтації цей підхід може допомогти оцініти ймовірність успіху учня у певній професійній сфері, виходячи з його академічних результатів та особистісних якостей. Серед алгоритмів регресії можна виділити:

− лінійну регресію, яка є базовим методом, який моделює залежність між вхідними змінними та вихідним числовим показником. Модель обчислює вагові коефіцієнти, які мінімізують середньоквадратичну похибку. Для нашого завдання це може бути прогноз рівня успішності учня у певній галузі;

− поліноміальну регресію, яка є розширенням лінійної моделі, яка дозволяє враховувати нелінійні залежності між змінними. Наприклад, взаємозв’язок між рівнем креативності учня та його придатністю до дизайнерської професії може бути змодельований за допомогою цієї регресії;

− рідж-регресія, що використовує регуляризацію для зменшення перенавчання моделі. Це особливо корисно у нашому дослідженні, де може бути велика кількість ознак (наприклад, характеристики учнів із кількох тестів).

Кластеризація — це метод навчання без учителя, який використовується для групування об'єктів на основі їхніх схожих характеристик. У нашому дослідженні кластеризація дозволить створити групи учнів сгрупувавши їх за подібними характеристиками, наприклад, "групи технічного спрямування", "групи креативних професій". В кластеризації можна виділити наступні алгоритми:

− метод K-середніх (K-Means), що є одним із найпоширеніших підходів, який ітеративно визначає центри кластерів, що мінімізують суму відстаней між точками даних і центром. У профорієнтації цей метод може групувати учнів за схожими академічними досягненнями та інтересами;

− ієрархічна кластеризація яка дозволяє побудувати деревовидну структуру груп, що корисно для сегментування учнів за кількома рівнями (наприклад, початковий розподіл за професійними напрямами, а далі — за конкретними спеціалізаціями);

− dbscan алгоритм, який визначає кластери з довільною формою, базуючись на щільності точок. Цей метод підходить для аналізу даних, які мають нерівномірний розподіл (наприклад, психологічні характеристики учнів).

Таким чином, виходячи з аналізу алгоритмів та методів машинного навчання, в нашому дослідженні класифікація дозволить створювати персоналізовані рекомендації на основі багатовимірного аналізу характеристик учня, регресійні моделі дозволять забезпечити точний прогноз та оцінити перспективність обраного напрямку навчання а кластеризація забезпечить ефективний спосіб сегментації учнів для побудови узагальнених рекомендацій або визначення трендів серед груп.

Поєднання цих підходів забезпечить комплексний підхід до формування навчально-професійної траєкторії, враховуючи як індивідуальні особливості, так і загальні тенденції.

2.3 Розробка математичних моделей, постановка задачі моделювання

Математична модель – наближений опис об'єкта моделювання, виражений за допомогою математичної символіки. Математичні моделі є невід’ємною частиною математики. Значний поштовх до розвитку математичного моделювання додало створення електронно-обчислювальних машин (ЕОМ). Застосування ЕОМ дозволило проаналізувати і застосувати на практиці багато математичні моделей та формул, які раніше не піддавалися аналітичному дослідженню. Реалізовану на комп'ютері математичну модель називають комп'ютерною математичною моделлю, а проведення цілеспрямованих

розрахунків за допомогою комп'ютерної моделі називається обчислювальним експериментом. [6]

У рамках нашого дослідження будуть використані методи машинного навчання для формування навчально-професійних траєкторій старшокласників. Для досягнення мети нами буде розглянуто три основні підходи: класифікація, регресія та кластеризація.

В якості математичних моделей класифікації, нами були обрані моделі розрахунку ентропії та інтдексу Джині, які відображені в формулах 2.1 та 2.2

$$H(S) = -\sum_{i=1}^{n} p_i log_2 (p_i), \quad (2.3)$$

де p_i — частка об'єктів класу i у вибірці S;

$$G(S) = 1 - \sum_{i=1}^{n} p_i^2, \quad (2.1)$$

де p_i — частка об'єктів класу i у вибірці S;

В якості математичних моделей регресії, нами була обрана стандартна модель лінійної регресії, яка відображені в формулах 2.3:

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \epsilon, \quad (2.2)$$

де y – прогнозоване значення;

β_i - вагові коефіцієнти;

x_i - вхідні ознаки;

ϵ - похибка;

В якості математичних моделей регресії, нами була обрана стандартна модель K-Means, яка відображені в формулах 2.4:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x_i - \mu_k||^2, \quad (2.3)$$

де k - кількість кластерів;

C_i – кластер i;

μ_i - центр кластера i;

$||x_i - \mu_k||$ - відстань між точкою x і центром кластера;

2.4 Підготовка та обробка даних

Підготовка даних є одним із найважливіших етапів дослідження, оскільки від їх якості залежить точність і надійність роботи моделей машинного навчання. Процес підготовки даних включає кілька важливих етапів, кожен із яких спрямований на створення повноцінної, збалансованої та репрезентативної вибірки, придатної для аналізу та моделювання.

Першим етапом є збір даних, що включає отримання інформації про старшокласників із різних джерел. Це можуть бути академічні записи, результати тестів, анкети, психологічні опитування та інформація про інтереси. На цьому етапі важливо забезпечити різноманітність джерел даних, аби охопити учнів із різних регіонів, соціальних груп та з різними рівнями академічної підготовки. Такий підхід дозволяє зробити вибірку більш репрезентативною, що сприятиме підвищенню універсальності та точності моделей.

Після збору даних починається процес очищення. Це необхідно для виявлення та усунення неточностей, таких як дублікати записів, некоректні або аномальні значення. Наприклад, якщо в записах є оцінка, яка перевищує максимальну можливу для певної шкали, або результати, що не відповідають логіці (наприклад, високий рівень математичних знань у поєднанні з низькими оцінками з базової математики), такі дані повинні бути скориговані або видалені. Очищення допомагає уникнути впливу помилкової інформації на результати роботи моделей.

Наступним важливим етапом є нормалізація даних, що забезпечує узгодженість різних показників. У нашому випадку це може означати приведення оцінок до єдиної шкали або стандартизацію показників психологічних тестів. Наприклад, якщо оцінки представлені як у 5-бальній, так і у 12-бальній системі, їх потрібно перевести в єдину шкалу, щоб забезпечити

рівноцінний вплив цих даних на модель. Нормалізація також дозволяє уникнути ситуації, коли характеристики з більшими числовими значеннями непропорційно впливають на результати аналізу.

Особливу увагу слід приділити обробці пропущених значень, адже відсутність даних у вибірці може суттєво вплинути на точність моделювання. Пропущені значення можуть бути заповнені кількома способами. Наприклад, середнє або медіанне значення може бути використане для числових характеристик, тоді як для категорійних даних може застосовуватися найчастіший клас. У деяких випадках пропущені значення можна залишити як окрему категорію, якщо вони мають потенційне значення для аналізу (наприклад, пропущені оцінки з певного предмету можуть вказувати на слабкий інтерес до цього напряму). Вибір підходу до обробки пропущених даних залежить від специфіки дослідження та ролі цих даних у моделях.

Для забезпечення репрезентативності даних потрібно дотримуватися певних критеріїв. По-перше, вибірка повинна охоплювати учнів із різними характеристиками, такими як академічний рівень, інтереси, соціально-економічний статус тощо. Це дозволяє моделі бути універсальною та ефективною для різних груп учнів. По-друге, важливо забезпечити збалансованість вибірки, особливо у випадках, коли аналізуються категорійні дані. Наприклад, якщо більшість даних належить учням, схильним до технічних спеціальностей, а інші напрями представлені слабо, модель може працювати упереджено. Для цього використовуються методи балансування, такі як збільшення кількості даних для менш представлених груп.

Нарешті, дані повинні бути адаптованими до специфіки дослідження. Наприклад, характеристики, які не мають відношення до задачі профорієнтації (такі як місце проживання, якщо воно не впливає на доступність професій), можуть бути виключені з аналізу. Це допомагає зменшити шум у даних та зосередитися на ключових показниках, які впливають на формування навчально-професійної траєкторії [7].

Таким чином, підготовка та обробка даних — це багатоетапний процес, який включає збір, очищення, нормалізацію та обробку пропущених значень. Усі ці дії спрямовані на створення якісної вибірки, яка відповідає вимогам дослідження та дозволяє забезпечити ефективність роботи моделей машинного навчання. Репрезентативність, точність і узгодженість даних є ключовими факторами успішного моделювання та отримання корисних результатів.

2.5 Функціональна схема та основні алгоритми дослідження

Згідно аналізу предметної області та математичних моделей, нами було розроблено схему моделювання яка зображена на рисунку 2.1:

Рисунок 2.1 – Схема моделювання дослідження

На моделі зображено, як вхідні дані — такі як оцінки учнів, їхні інтереси, професійні уподобання та результати тестів — проходять етап обробки, включаючи нормалізацію, очищення та підготовку до аналізу. Далі ці дані використовуються трьома моделями. Класифікаційна модель призначає

ймовірну професію на основі даних, регресійна модель оцінює рівень відповідності учня обраній професії, а кластеризація дозволяє групувати учнів за схожістю у професійних уподобаннях. У кінцевому етапі результати представлені у вигляді графіків, профілів і рекомендацій, що допомагає учням та їхнім наставникам краще зрозуміти, у яких професіях вони мають найбільші шанси на успіх. Логіка роботи побудована таким чином, щоб адаптуватися до різноманітних сценаріїв використання та забезпечувати гнучкість у масштабуванні системи.

Далі, нами було розроблено структурну схему майбутнього застосунку, що відображена на рисунку 2.2:

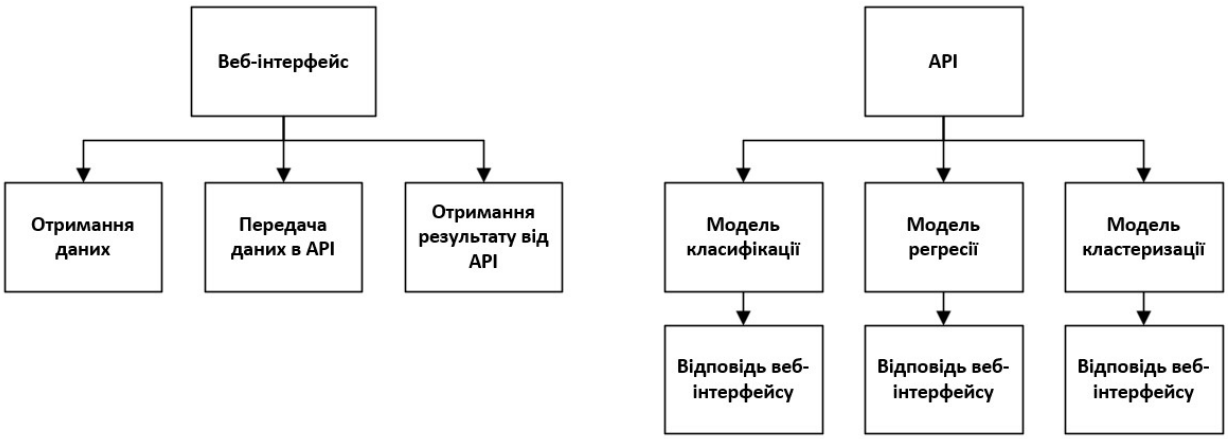

Рисунок 2.2 – Структурна схема

Представлена схема демонструє основні компоненти та процеси роботи системи автоматизації профорієнтації. Вона складається з двох головних модулів: веб-інтерфейсу та API, які тісно взаємодіють між собою для забезпечення функціональності системи.

Модуль веб-інтерфейсу виконує роль взаємодії з користувачем. Користувач через інтерфейс вводить дані про учнів, які потім передаються в API для обробки. Після цього веб-інтерфейс отримує результат від API (наприклад,

рекомендації чи прогнози) та надає ці результати користувачеві у зручному форматі.

Модуль API відповідає за обробку отриманих даних. Він складається з трьох основних моделей: класифікації, регресії та кластеризації. Кожна з моделей обробляє отримані дані відповідно до своєї мети: класифікація визначає професійні інтереси, регресія прогнозує ймовірність успіху в обраній сфері, а кластеризація групує учнів за спільними характеристиками. Після обробки результати передаються назад у веб-інтерфейс для відображення.

На основі розробленої структурної схеми, нами було сформовано функціональну схему роботи веб-інтрефейсу, що зображена на рисунку 2.3:

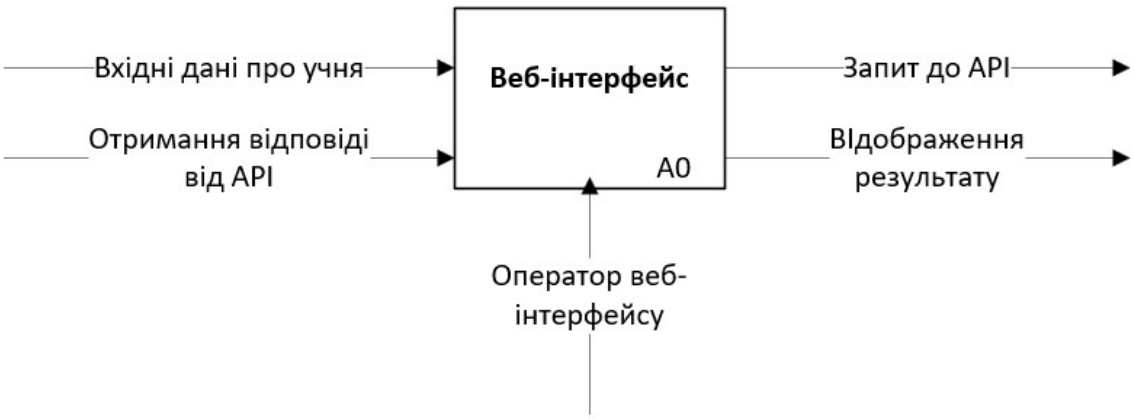

Рисунок 2.3 – Функціональна схема веб-інтерфейсу

Ця схема ілюструє функціонування веб-інтерфейсу в системі автоматизації профорієнтації. Веб-інтерфейс є центральною ланкою між оператором (користувачем) та API. Додатково нами була розроблена детальна функціональна схема, яка описує більш детальну передачу даних. Схема зображена на рисунку 2.4.

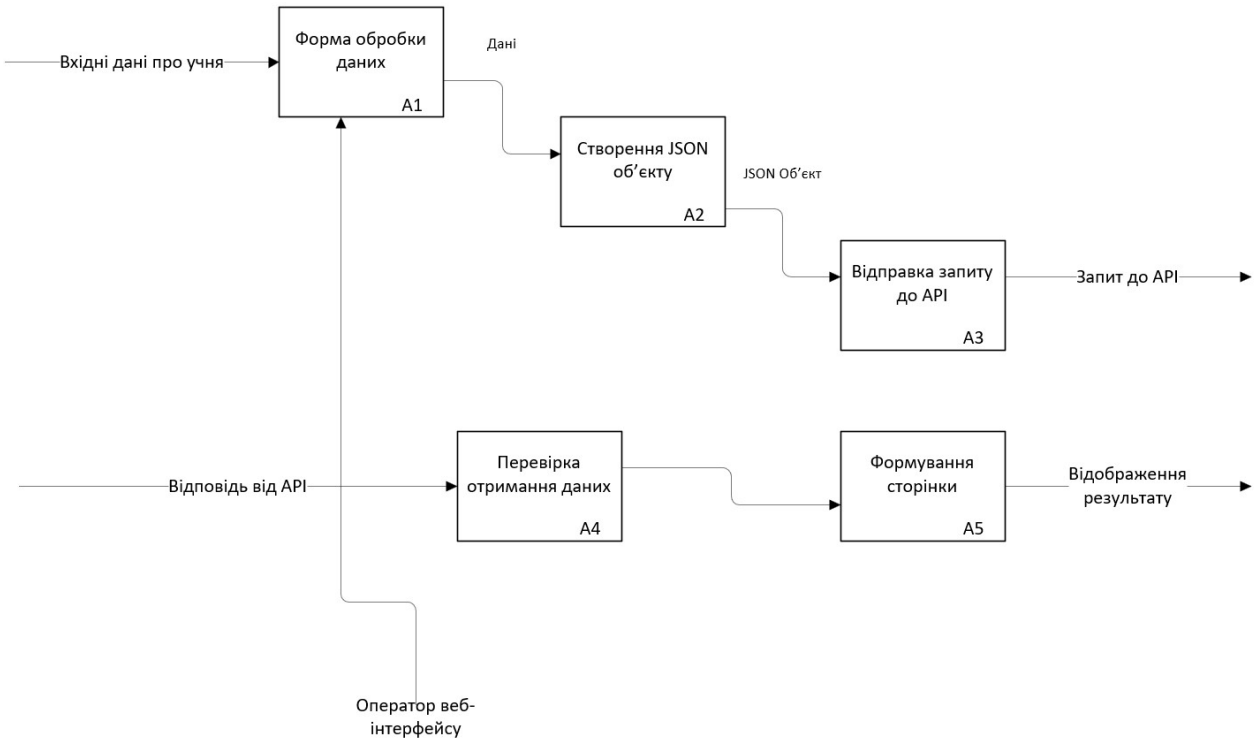

Рисунок 2.4 – Детальна функціональна схема веб-інтерфейсу

Користувач вводить вхідні дані про учня через інтерфейс. Ці дані надсилаються у вигляді запиту до API для обробки. Після отримання відповіді від API веб-інтерфейс відображає результати обробки оператору у зручному форматі. Така структура забезпечує зручність використання системи та швидкий доступ до результатів аналізу.

Відповідно до веб-інтерфейсу, нами також була розроблена функціональна схема роботи API інтерфейсу, що зображеня на рисунку 2.5. Ця схема демонструє структуру API, що забезпечує обробку запитів від веб-інтерфейсу. API є центральним компонентом, який використовує три моделі: класифікації, регресії та кластеризації. Вхідний запит, отриманий від веб-інтерфейсу, передається відповідній моделі в залежності від типу задачі. Після обробки даних кожна модель генерує відповідь, яка повертається до API. API, у свою чергу, формує єдину відповідь і передає її назад веб-інтерфейсу для відображення результатів користувачу. Така архітектура забезпечує гнучкість та масштабованість системи.

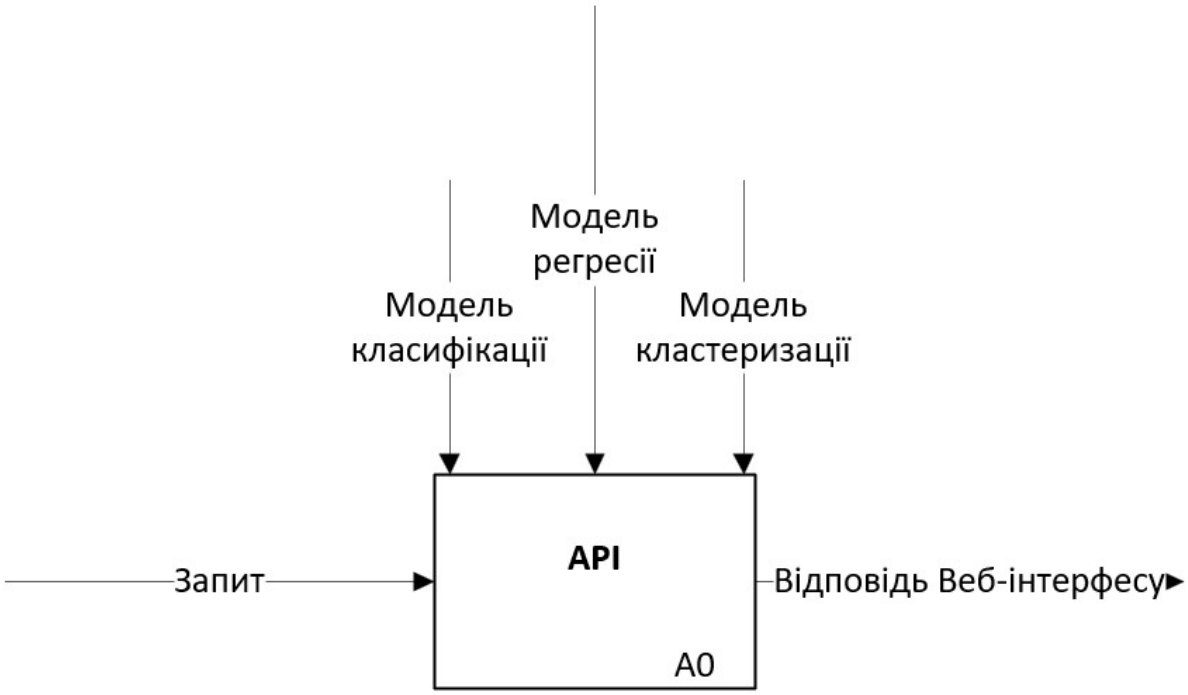

Рисунок 2.5 – Функціональна схема API

Додатково нами була розроблена детальна функціональна схема, яка описує більш детальну передачу даних. Схема зображена на рисунку 2.6.

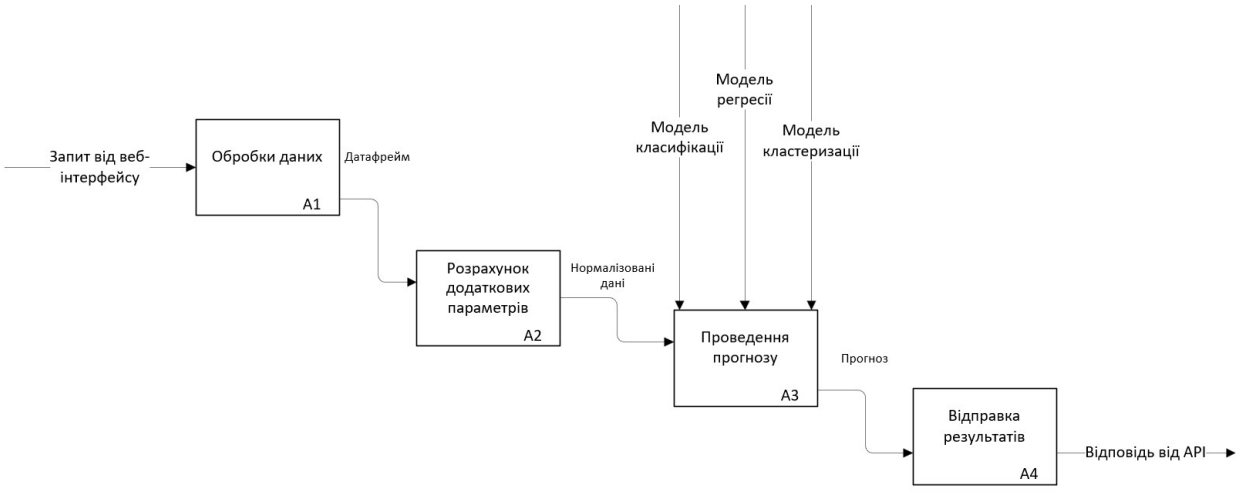

Рисунок 2.6 – Деталізована функціональна схема API

2.6 Критерії оцінки ефективності методів

Оцінка ефективності методів машинного навчання є надзвичайно важливим етапом у дослідженні, оскільки саме за допомогою певних критеріїв можна визначити, наскільки добре алгоритми виконують поставлені задачі. Для цього використовуються метрики, які допомагають аналізувати якість роботи моделей залежно від типу завдання. У нашому випадку, це завдання класифікації, регресії та кластеризації, і кожен із цих методів має свої специфічні метрики оцінки.

Для класифікації основною метрикою є точність (Accuracy). Вона визначає, яку частку всіх об’єктів алгоритм правильно класифікував. Однак, у випадках, коли класи у вибірці представлені нерівномірно, точність може бути недостатньо інформативною. Наприклад, якщо більшість учнів належать до однієї категорії, модель може просто завжди обирати цей клас і показувати високу точність, хоча її реальна ефективність буде низькою. У таких випадках застосовуються додаткові метрики, як-от Precision (точність позитивного класу) та Recall (повнота). Precision показує, яку частку об’єктів алгоритм правильно класифікував до певного класу серед усіх, які він до цього класу відніс. Recall, своєю чергою, показує, наскільки добре модель знайшла всі об’єкти конкретного класу серед реальних даних. F1-міра є середньозваженим значенням Precision і Recall, і вона дозволяє оцінити модель у ситуаціях, коли обидва ці аспекти важливі. Ще однією важливою метрикою для класифікації є ROC-AUC, яка вимірює якість розподілу класів незалежно від порогів прийняття рішень [8] .

У задачах регресії основними метриками є середньоквадратична помилка (Mean Squared Error, MSE) та середня абсолютна помилка (Mean Absolute Error, MAE). MSE є особливо корисною, коли важливо враховувати великі відхилення, адже вона підсилює їхній вплив через квадратування. Це дозволяє швидко виявити значні похибки у прогнозах моделі. MAE, у свою чергу, фокусується на середньому відхиленні між передбаченнями та реальними значеннями, менш чутлива до великих похибок і часто використовується для оцінки загальної точності прогнозу. Ще одним важливим критерієм для регресії є коефіцієнт

детермінації (R^2), який показує, наскільки добре модель пояснює варіацію залежної змінної. Значення R^2, близьке до 1, свідчить про високу якість моделі.

У кластеризації метрики відрізняються, оскільки вона не має заздалегідь визначених міток класів. Основною метрикою є силуетний коефіцієнт (Silhouette Score), який оцінює, наскільки добре кожен об'єкт належить до свого кластеру порівняно з іншими кластерами. Чим ближче це значення до 1, тим краще. Також важливими є середня відстань до центру кластера, яка показує щільність кластерів, та Adjusted Rand Index (ARI), що оцінює якість кластеризації, якщо реальні мітки все ж відомі.

Обґрунтування вибору критеріїв оцінки ефективності базується на специфіці задачі. Для класифікації важливо не лише визначити точність, але й враховувати баланс між Precision і Recall, щоб рекомендації були точними і водночас повними. У регресії метрики MSE і MAE допомагають оцінити похибки, а R^2 — зрозуміти, наскільки модель відповідає реальним даним. У кластеризації силуетний коефіцієнт є ключовим для визначення якості сегментації учнів [9].

Таким чином, правильний вибір критеріїв оцінки ефективності дозволяє визначити, які методи є найбільш придатними для задачі формування навчально-професійної траєкторії старшокласників. Це, своєю чергою, забезпечує створення ефективної системи рекомендацій, яка буде корисною та надійною для використання у профорієнтації.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ

3.1 Вибір інструментарію для реалізації методів машинного навчання

Реалізація методів машинного навчання в рамках цього дослідження є складним і багатоступеневим процесом, який вимагає використання сучасних інструментів для аналізу, обробки даних та побудови моделей. Правильний вибір мови програмування, бібліотек і середовища розробки має вирішальне значення, адже це впливає як на ефективність роботи, так і на зручність подальшого вдосконалення алгоритмів. У цьому розділі нами буде детально розглянуто мови програмування, їхні переваги та недоліки, огляд бібліотек для машинного навчання та обґрунтування вибору інструментарію для реалізації поставлених завдань.

Python є найпопулярнішою мовою програмування для задач машинного навчання. Його популярність пояснюється простим синтаксисом, широкими можливостями та розвиненою екосистемою бібліотек [12]. Python підтримує всі етапи реалізації алгоритмів, від попередньої обробки даних до побудови складних моделей. Головними перевагами Python є:

− простота синтаксису. Python є ідеальною мовою для студентів і дослідників, оскільки дозволяє швидко реалізовувати алгоритми без глибоких знань програмування;

− екосистема бібліотек. Python має велику кількість бібліотек, таких як Scikit-learn, TensorFlow, Pandas і NumPy, які охоплюють усі аспекти роботи з даними;

− широка спільнота. Завдяки активній спільноті користувачів Python має багато ресурсів, таких як документація, форуми та навчальні матеріали, які полегшують вивчення мови та вирішення технічних проблем.

R є мовою, розробленою для статистичного аналізу та візуалізації даних. Ця мова часто використовується у наукових дослідженнях, зокрема у задачах, де потрібна складна статистична обробка даних. Головними перевагами R є:

− Потужні статистичні інструменти. Мова R має багатий набір пакетів для аналізу даних, таких як caret, mlr і tidyverse;

− Розвинена візуалізація. Вбудовані інструменти для побудови графіків та діаграм дозволяють створювати інформативні візуалізації даних. Однак R має обмеження у продуктивності та менш гнучка у створенні масштабованих систем, ніж Python.

Java залишається популярною мовою програмування для розробки масштабованих програм, зокрема систем машинного навчання. Основними перевагами Java є:

− продуктивність. Завдяки компіляції у байт-код Java забезпечує високу швидкість виконання алгоритмів;

− інтеграція. Java легко інтегрується з великими корпоративними системами. Проте складний синтаксис Java та обмежений вибір бібліотек для машинного навчання (наприклад, Weka або DL4J) роблять її менш популярною для швидкого прототипування [13].

C++ є мовою для високопродуктивних обчислень. Вона забезпечує максимальну швидкість виконання, що є важливим у задачах реального часу. Проте складний синтаксис і висока складність реалізації алгоритмів роблять C++ менш зручною для задач, що вимагають швидкого тестування та розробки.

Julia є відносно новою мовою, яка поєднує високу продуктивність із простотою. Вона спеціалізується на наукових обчисленнях і підтримує машинне навчання за допомогою бібліотек, таких як MLJ.jl. Основні переваги Julia:

− швидке виконання завдяки JIT-компіляції;

− інтуїтивний синтаксис. Однак невелика спільнота і обмежена кількість навчальних матеріалів зменшують її привабливість для початківців.

Виходячи з аналізу мов програмування, на основі отриманих характеристик та інструментів які пропонує кожна мова програмування, нами

було обрано Python як основну мову розробки моделей. Далі нами буде розглянуто існуючі бібліотеки, які пропонують різноманітні реалізації алгоритмів штучного інтелекту.

Scikit-learn є однією з найпотужніших бібліотек для реалізації класичних алгоритмів машинного навчання. Вона надає інструменти для класифікації, регресії, кластеризації, а також для попередньої обробки даних [14]. У нашому дослідженні Scikit-learn використовується для:

− класифікації: алгоритми, такі як DecisionTreeClassifier і SVC, дозволяють прогнозувати категорії для учнів;

− регресії: LinearRegression і Ridge підходять для оцінки числових значень, таких як потенційна успішність у певній професії;

− кластеризації: алгоритми, такі як KMeans і DBSCAN, допомагають групувати учнів за схожими характеристиками.

Pandas, бібліотека, що забезпечує роботу з табличними даними, а NumPy дозволяє швидко виконувати математичні операції. Ці бібліотеки використовуються для очищення, нормалізації та аналізу даних. Наприклад, Pandas дозволяє видаляти пропущені значення, а NumPy допомагає масштабувати дані до однакового діапазону.

Візуалізація даних є важливою частиною дослідження, тому бібліотека Matplotlib, що дозволяє створювати базові графіки, а Seaborn яка надає можливості для побудови теплових карт і кореляційних діаграм, будуть використані нами для виявлення закономірностей у даних.

На основі огляду інструментів було прийнято рішення використовувати Python у поєднанні з бібліотеками Scikit-learn, Pandas, NumPy, Matplotlib і Seaborn. Цей вибір зумовлений:

− простотою інтеграції між бібліотеками;

− широким спектром доступних алгоритмів для класифікації, регресії та кластеризації;

− гнучкістю у роботі з даними, зокрема у їхній підготовці, аналізі та візуалізації.

3.2 Підготовка даних для дослідження

Ефективність роботи алгоритмів машинного навчання значною мірою залежить від якості вхідних даних. Тому важливим етапом нашого дослідження є ретельна підготовка даних, яка включає очищення, нормалізацію, створення додаткових метрик і розділення даних на набори для навчання, тестування та валідації.

Для моделей класифікації, нами було видалено стовбчики, що зберігають інформацію про ім’я, прізвище та електронну адресу, оскільки вони не приймають участі в побудові моделей та прогнозуванні результатів. Також, на етапі очищення даних, нами було перетворено категоріальні дані, такі як стать, наявність роботи, рівень логічного мислення, рівень творчих здібностей, участь у позакласних активностях та емоційний інтелект в числовий формат, лістинг коду зображено на рисунку 3.1:

```
label_encoder = LabelEncoder()
df['gender'] = label_encoder.fit_transform(df['gender'])
df['part_time_job'] = label_encoder.fit_transform(df['part_time_job'])
df['logical_reasoning'] = label_encoder.fit_transform(df['logical_reasoning'])
df['creativity'] = label_encoder.fit_transform(df['creativity'])
df['emotional_intelligence'] = label_encoder.fit_transform(df['emotional_intelligence'])
df['extracurricular_activities'] = df['extracurricular_activities'].apply(lambda x: 1 if x else 0)
df['career_aspiration_encoded'] = label_encoder.fit_transform(df['career_aspiration'])
```

Рисунок 3.1 – Переведення категоріальних ознак в числові

Нормалізація, лістинг коду якої зображено на рисунку 3.2, виконувалася для оцінок із предметів, оскільки їх значення знаходяться в діапазоні від 2 до 12, а інші ознаки мають різні діапазони.

```
score_columns = ['math_score', 'history_score', 'physics_score', 'chemistry_score',
                 'biology_score', 'english_score', 'geography_score']
scaler = StandardScaler()
df[score_columns] = scaler.fit_transform(df[score_columns])
```

Рисунок 3.2 – Нормалізація числових даних

Це дозволило привести всі числові ознаки до однакового масштабу, що є важливим для алгоритмів машинного навчання, чутливих до масштабів даних.

Додаткові метрики були створені для підвищення точності моделей. Вони відображають рівень взаємодії студента, його нахили до технічних, гуманітарних чи медичних професій та розраховані за формулами математичної моделі. Лістинг коду створення метрик зображено на рисунку 3.3.

```
def calculate_engagement_index(part_time_job, extracurricular_activities, weekly_self_study_hours):
    job_weight = 0.3 if part_time_job == 1 else 0
    activities_weight = 0.4 * extracurricular_activities
    study_weight = 0.3 * weekly_self_study_hours
    return job_weight + activities_weight + study_weight

# Функции для расчета профессиональных индексов
def calculate_tech_index(math_score, physics_score, interest_science_tech):
    return 0.5 * math_score + 0.3 * physics_score + 0.2 * interest_science_tech

def calculate_humanities_index(history_score, english_score, interest_art_humanities):
    return 0.4 * history_score + 0.4 * english_score + 0.2 * interest_art_humanities

def calculate_medical_index(biology_score, chemistry_score, weekly_self_study_hours):
    return 0.4 * biology_score + 0.3 * chemistry_score + 0.3 * weekly_self_study_hours

# Применение функций для добавления новых метрик в DataFrame
df['engagement_index'] = df.apply(lambda row: calculate_engagement_index(
    row['part_time_job'], row['extracurricular_activities'], row['weekly_self_study_hours']), axis=1)

df['tech_index'] = df.apply(lambda row: calculate_tech_index(
    row['math_score'], row['physics_score'], row['interest_science_tech']), axis=1)

df['humanities_index'] = df.apply(lambda row: calculate_humanities_index(
    row['history_score'], row['english_score'], row['interest_art_humanities']), axis=1)

df['medical_index'] = df.apply(lambda row: calculate_medical_index(
    row['biology_score'], row['chemistry_score'], row['weekly_self_study_hours']), axis=1)
```

Рисунок 3.3 – Алгоритм створення додаткових метрик

Наостанок, для підготовки даних нами було проведено розділення початкових даних на 3 вибірки, а саме: навчальну, тестову та валідаційну. Лістинг коду розподілу даних відображено на рисунку 3.4.

```
from sklearn.model_selection import train_test_split

X = df.drop(columns=['career_aspiration','career_aspiration_encoded'])
y = df['career_aspiration_encoded']

x_train,x_test,y_train,y_test=train_test_split(X_resampled,y_resampled,test_size=0.2,random_state=0 )
```

Рисунок 3.4 – Алгоритм розподілу даних на вибірки

Для покращення точності моделей регресії та уникнення їх перенавчання, нами було створено додатковий параметр який призначений врегулювати однотимність данних та зменшити шанси на перенавчання. Реалізацію створення параметру відображено на рисунку 3.5.

```
df['average_score'] = df[score_columns].mean(axis=1)
df['success_probability'] = (
    0.4 * df['engagement_index'] +
    0.6 * df['average_score']
)
```

Рисунок 3.5 – Додактовий параметр регресії

Для проведення обєктивного дослідження кластеризації, нам необхідно буде отримати оптимальну кількості кластерів на основі графіку який ми отримуємо після використання методу ліктя, програнму реалізацію якого зображено на рисунку 3.6.

```
inertia = []
range_n_clusters = range(2, 11)
for n_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(clustering_data)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(8, 6))
plt.plot(range_n_clusters, inertia, marker='o', linestyle='--')
plt.title('Метод ліктя для визначення оптимальної кількості кластерів')
plt.xlabel('Кількість кластерів')
plt.ylabel('Inertia')
plt.grid()
plt.show()
```

Рисунок 3.6 – Навчання моделі методом KMeans

Таким чином, ми забезпечили якісне навчання моделей класифікації, регресії та кластеризації, їх валідацію і мінімізували ризики винекнення перевидобутку.

3.3 Реалізація алгоритмів класифікації

У процесі дослідження ми реалізували алгоритми класифікації для прогнозування майбутньої професії старшокласників на основі їхніх академічних оцінок, навичок, інтересів та додаткових метрик. Реалізація класифікаційних моделей включала три основні етапи: створення та навчання моделей, оптимізацію параметрів і валідацію точності.

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier

classifiers = [
    LogisticRegression(),
    RandomForestClassifier(),
    SVC(),
    DecisionTreeClassifier(),
    GaussianNB(),
    GradientBoostingClassifier(),
    KNeighborsClassifier()
]

for classifier in classifiers:
    classifier.fit(X_train_scaled, y_train)
    y_pred = classifier.predict(X_test_scaled)

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Classifier: {classifier.__class__.__name__}")
    print(f"Accuracy: {accuracy:.4f}")

    print("Classification Report:")
    print(classification_report(y_test, y_pred))

    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))
    print("="*50)
```

Рисунок 3.7 – Алгоритм порівняння навчаємих моделей

В першу чергу нами було реалізовано алгоритм порівняння навчаємих моделей за різними класифікаційними алгоритмами, які пропонує нам бібліотека sklearn. Порівняння різних алгоритмів класифікації дозволило нам оцінити сильні та слабкі сторони кожного методу й вибрати найкращий метод для реалізації задачі прогнозування професій старшокласників. Реалізація алгоритму зображена на рисунку 3.7.

Для покращення роботи моделі ми провели оптимізацію гіперпараметрів, використовуючи метод GridSearchCV. Це дозволило автоматично перевірити різні комбінації параметрів, таких як кількість дерев, максимальна глибина дерев і кількість ознак для розділення вузлів. У результаті ми змогли визначити найкращі параметри, які забезпечили баланс між точністю моделі та швидкістю навчання. Такий підхід дозволив значно підвищити продуктивність моделі, роблячи її більш точною та ефективною для прогнозування професій старшокласників. Лістинг коду для оптимізації моделі відображено на рисунку 3.8.

```
from sklearn.model_selection import GridSearchCV

param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

grid_search = GridSearchCV(RandomForestClassifier(), param_grid, cv=3, scoring='accuracy', n_jobs=-1, verbose=2)
grid_search.fit(x_train, y_train)

best_model = grid_search.best_estimator_
print("Лучшие параметры для RandomForestClassifier:", grid_search.best_params_)
print("Точность лучшей модели на тестовых данных:", accuracy_score(y_test, best_model.predict(x_test)))
```

Рисунок 3.8 – Алгоритм оптимізація гіперпараметрів моделі

Для оцінки ефективності моделі ми провели валідацію на тестовій вибірці, яка становила 20% від усіх даних. Ми використали метрики точності, звіт класифікації та матрицю сплутування для аналізу результатів. Це дозволило визначити, як добре модель класифікує кожен клас, виявити потенційні слабкі сторони та оцінити здатність моделі узагальнювати дані. Завдяки валідації ми впевнилися, що модель працює стабільно й ефективно на нових, невідомих даних. Лістинг коду для оптимізації моделі відображено на рисунку 3.9.

```
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_test_pred_rf))
print("\nClassification Report:")
print(classification_report(y_test, y_test_pred_rf))

predicted_labels = rf_model.predict(x_test)
predicted_series = pd.Series(predicted_labels)

actual_counts = y_test.value_counts()
predicted_counts = predicted_series.value_counts()

plt.figure(figsize=(10, 5))

actual_counts.plot(kind='bar', color='blue', width=0.4, position=1, label='Actual Labels')
predicted_counts.plot(kind='bar', color='red', width=0.4, position=0, label='Predicted Labels')

plt.xlabel('Labels')
plt.ylabel('Count')
plt.title('Comparison of Actual and Predicted Labels')
plt.legend()
plt.show()
```

Рисунок 3.9 – Алгоритм оцінки ефективності моделі

Порівняння різних алгоритмів класифікації дозволило нам зробити обґрунтований вибір оптимальної моделі. RandomForestClassifier показав найкращі результати, завдяки чому його було обрано для подальшої роботи. Однак інші моделі, такі як Gradient Boosting, також залишаються перспективними для можливих удосконалень у майбутньому.

```
models = {
    "LinearRegression": LinearRegression(),
    "Ridge": Ridge(alpha=1.0),
    "Lasso": Lasso(alpha=0.1),
    "RandomForestRegressor": RandomForestRegressor(n_estimators=100, random_state=42),
    "GradientBoostingRegressor": GradientBoostingRegressor(n_estimators=100, random_state=42)
}


best_model = None
best_r2 = -np.inf

for name, model in models.items():
    scores = cross_val_score(model, X_train_scaled, y_train, cv=5, scoring='r2')
    print(f"Модель: {name}")
    print(f"Середній R² на крос-валідації: {scores.mean():.4f}")
    print(f"R² на кожному фолді: {scores}")
    print("=" * 50)

    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    r2 = r2_score(y_test, y_pred)
    print(f"R² на тестовому наборі: {r2:.4f}")
    print("-" * 50)

    if r2 > best_r2:
        best_r2 = r2
        best_model = model

print(f"Найкраща модель: {best_model.__class__.__name__} з R² = {best_r2:.4f}")
```

Рисунок 3.10 – Алгоритм порівняння моделей регресії

У процесі роботи, для дослідження методів регресії, нами було реалізовано алгоритм порівняння різних моделей регресії, які пропонує бібліотека sklearn. Метою реалізації було обрати найкращий метод для задачі прогнозування середнього бала та оцінки ймовірності успішності старшокласників у вибраній професійній траєкторії.регресії. Алгоритм порівняння зображено на рисунку 3.10.

Реалізація дослідження механізмів кластеризації, вимагала від нас розробки алгоритмів фільтрації, навчання та оцінки кожного алгоритму. На основі графіку який ми отримали згідно результатів методу ліктя, ми перейши до реалізації навчання моделі KMeans з вказаною кількістю кластерів. Програмна реалізація алгоритму кластеризації за допомогою методу KMeans зображено на рисунку 3.11.

```
optimal_clusters = 4

kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
df['kmeans_cluster'] = kmeans.fit_predict(clustering_data)


plt.figure(figsize=(8, 6))
plt.scatter(clustering_data[:, 0], clustering_data[:, 1], c=df['kmeans_cluster'], cmap='viridis')
plt.title('Кластери KMeans')
plt.xlabel('Engagement Index')
plt.ylabel('Tech Index')
plt.colorbar(label='Cluster')
plt.show()

silhouette_avg_kmeans = silhouette_score(clustering_data, df['kmeans_cluster'])
print(f"Середній силуетний бал для KMeans: {silhouette_avg_kmeans:.4f}")
```

Рисунок 3.11 – Навчання моделі методом KMeans

Для обєктивізації дослідження, нами також було реалізовано моделі кластеризації на основі методів DBScan та на основі власних формул розрахунку параметру кластеризації. В якості власного параметру нами була розрахована формула з вагами яка включала в себе індекс спрямованості учня гуманітарними чи технічними напрямками. Реалізація методу DBScan та кластеризації на основі власного параметру зображено на рисунку 3.12 та рисунку 3.13 відповідно.

```
dbscan = DBSCAN(eps=1.5, min_samples=5)
df['dbscan_cluster'] = dbscan.fit_predict(clustering_data)

plt.figure(figsize=(8, 6))
plt.scatter(clustering_data[:, 0], clustering_data[:, 1], c=df['dbscan_cluster'], cmap='plasma')
plt.title('Кластери DBSCAN')
plt.xlabel('Engagement Index')
plt.ylabel('Tech Index')
plt.colorbar(label='Cluster')
plt.show()
```

Рисунок 3.12 – Реалізація методу DBScan

```
df['custom_score'] = (
    0.5 * df['tech_index'] +
    0.3 * df['humanities_index'] +
    0.2 * df['medical_index']
)

df['custom_cluster'] = pd.cut(
    df['custom_score'],
    bins=[0, 0.33, 0.66, 1.0],
    labels=['Low Interest', 'Medium Interest', 'High Interest']
)


plt.figure(figsize=(8, 6))
plt.scatter(clustering_data[:, 0], clustering_data[:, 1], c=df['custom_cluster'].cat.codes, cmap='coolwarm')
plt.title('Кластери на основі Custom Score')
plt.xlabel('Engagement Index')
plt.ylabel('Tech Index')
plt.colorbar(label='Custom Cluster')
plt.show()
```

Рисунок 3.13 – Реалізація методу DBScan

Отже, нами було реалізовано основні методи машиного навчання для дослідження їх точності та актуальності використання в сфері професійної спрямованості старшокласників.

3.4 Створення API

На етапі розробки програмного забезпечення для автоматизації профорієнтації одним із ключових завдань стало створення Application Programming Interface(API). API — це технологія, яка забезпечує зв’язок між компонентами системи, дозволяючи різним частинам програмного забезпечення взаємодіяти між собою. В нашому випадку, API виступає посередником між веб-

інтерфейсом, за допомогою якого користувач вводить дані, і моделями машинного навчання, які аналізують ці дані та формують результат.

API необхідний, щоб забезпечити зручну й стандартизовану передачу даних між веб-інтерфейсом та моделями. Це дозволяє автоматизувати роботу системи: дані про учня обробляються моделями, які прогнозують найбільш імовірні професійні напрями, і повертаються у вигляді зрозумілої відповіді. Такий підхід значно спрощує процес профорієнтації та робить його більш ефективним [16].

Для реалізації API було обрано технологію **Flask**, яка є легким і потужним інструментом для створення веб-додатків. Flask дозволяє швидко створювати функціональні прототипи та підтримує стандарти REST API, що робить його ідеальним вибором для нашого проєкту.

До основних функції розробленої API можна віднести наступні:

− ініціалізація програми: створення основного додатку Flask і визначення маршрутів для обробки запитів;

− завантаження моделі: на початку роботи API завантажуються попередньо навчені моделі машинного навчання та мапінги даних, необхідні для роботи;

− обробка даних: дані, отримані з веб-інтерфейсу, проходять процес валідації, мапінгу категоріальних значень та обчислення додаткових метрик, таких як індекс залученості або технічний індекс;

− передбачення: оброблені дані передаються у модель, яка формує результат прогнозування;

− формування відповіді: API повертає веб-інтерфейсу результат у форматі JSON, що включає топ-3 професії та їх імовірності.

Програмна реалізації основних програмних алгоритмів класифікації в API відображено на рисунку 3.14

```python
app = Flask(__name__)

with open('RandomForestClassifier_model.pkl', 'rb') as model_file:
    rf_model = pickle.load(model_file)
with open('career_aspirations_mapping.pkl', 'rb') as mapping_file:
    career_mapping = pickle.load(mapping_file)

@app.route('/explore', methods=['POST'])
def explore():
    try:

        input_data = request.get_json()
        user_df = pd.DataFrame([input_data])

        user_df['engagement_index'] = user_df['weekly_self_study_hours'] * 0.3 + \
                                      user_df['extracurricular_activities'] * 0.4 + \
                                      user_df['part_time_job'] * 0.3
        probabilities = rf_model.predict_proba(user_df)

        top_indices = probabilities[0].argsort()[-3:][::-1]
        result = [{"career": career_mapping[idx], "probability": probabilities[0][idx]} for idx in top_indices]

        return jsonify({"predictions": result})
    except Exception as e:
        return jsonify({"error": str(e)}), 400

if __name__ == '__main__':
    app.run(debug=True)
```

Рисунок 3.14 – Програмна реалізація класифікації в API

3.5 Керівництво для користувача по роботі з API

Для забезпечення ефективної взаємодії з розробленим API надається керівництво, яке пояснює основні етапи його використання, підключення до системи та інтеграції з іншими компонентами.

API розгорнуто за допомогою Flask і доступне через HTTP-запити. Щоб підключитися до API, необхідно мати:

а) адресу сервера, на якому працює API (наприклад, http://localhost:5000/explore під час локального тестування);

б) інструмент для надсилання запитів, наприклад:

1) postman — графічний інтерфейс для тестування API;
2) cURL — інструмент командного рядка;
3) власний програмний клієнт, написаний на будь-якій мові програмування.

API приймає дані у форматі JSON. Приклад структури даних відображено в наступному лістингу коду:

```
{
  "gender": "male",
  "part_time_job": "yes",
  "absence_days": 5,
  "extracurricular_activities": "no",
  "weekly_self_study_hours": 8,
  "math_score": 10,
  "history_score": 8,
  "physics_score": 9,
  "chemistry_score": 7,
  "biology_score": 6,
  "english_score": 9,
  "geography_score": 7,
  "logical_reasoning": "medium",
  "creativity": "high",
  "emotional_intelligence": "low",
  "interest_science_tech": 0.8,
  "interest_art_humanities": 0.6,
  "interest_programming_it": 0.9
}
```

Перед використанням, необхідно впевнитися, що всі залежності (наприклад, Flask, Pandas, Scikit-learn) встановлені.Також необхідно протестувати API локально, перш ніж розгортати його на сервері.

Після надсилання запиту API:

а) приймає JSON-дані та обробляє їх, перетворюючи категоріальні значення в числові;

б) обчислює додаткові метрики (наприклад, engagement_index та інші);

в) передає оброблені дані в модель машинного навчання;

г) отримує результат прогнозу, формує відповідь та повертає її у форматі JSON.

4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ ТА АНАЛІЗ ЕФЕКТИВНОСТІ

4.1 Методика проведення дослідження

Розробка методики дослідження для формування навчально-професійної траєкторії старшокласників є важливим етапом, який забезпечує структурований підхід до вирішення поставлених задач. У межах даної роботи були визначені чіткі етапи, які охоплюють процес збору даних, їхню обробку, моделювання, розрахунки та аналіз результатів. Цей комплексний підхід дозволяє отримати надійні, точні та релевантні результати, які можуть бути застосовані для створення рекомендацій учням.

Процес починається зі збору даних, який включає отримання інформації про учнів, зокрема їхні академічні досягнення, психологічні характеристики, інтереси та кар'єрні уподобання. Джерелами таких даних є шкільні записи, анкети, результати тестувань і опитувальники. Зібрані дані охоплюють різні аспекти, які можуть впливати на вибір професійного напряму, наприклад, оцінки з ключових предметів, участь у конкурсах, рівень емоційного інтелекту та особистісні інтереси. Цей етап має на меті забезпечити повноту та різноманітність даних для подальшого аналізу.

Після збору даних виконується їх попередня обробка, яка є критично важливою для створення якісної вибірки. На цьому етапі проводиться очищення даних: виявляються й усуваються пропущені або аномальні значення. Наприклад, якщо в анкетах учнів відсутні оцінки з певного предмета, ці пропуски заповнюються середніми значеннями для відповідної категорії або спеціальними маркерами, які можуть бути оброблені моделями машинного навчання. Аномалії, такі як недосяжно високі оцінки, ідентифікуються та коригуються. Для забезпечення однакового масштабу характеристик дані проходять процес нормалізації, під час якого показники, наприклад, оцінки чи результати тестів, приводяться до єдиної шкали. Окрім цього, забезпечується балансування

вибірки, особливо для задач класифікації, щоб уникнути упередженості моделі до більш представлених категорій.

Наступний етап дослідження полягає у побудові моделей машинного навчання. У межах даної роботи обрано три основні напрямки: класифікація, регресія та кластеризація. Для кожного з цих напрямів було визначено конкретні алгоритми, які відповідають специфіці поставлених завдань. Наприклад, для класифікації використовувалися дерева рішень і метод опорних векторів, які дозволяють прогнозувати професійний напрямок, відповідний здібностям учнів. У задачах регресії застосовувалися лінійна та рідж-регресія для прогнозування ймовірності успіху учня в певній галузі. Кластеризація виконувалася за допомогою методу K-середніх, який допомагає групувати учнів за схожими характеристиками.

Особлива увага приділялася процесу тестування моделей. Для цього вибірка даних була розділена на три частини: навчальну, валідаційну та тестову. Це дозволило перевірити, наскільки моделі здатні працювати на нових даних, що не використовувалися під час навчання. Оцінка якості моделей здійснювалася за допомогою таких метрик, як точність (Accuracy), F1-міра та середньоквадратична помилка (MSE). Ці показники дозволили об’єктивно порівняти різні алгоритми та визначити найефективніші.

Після навчання моделей проводився аналіз отриманих результатів. Цей етап включав побудову графіків і візуалізацій, які демонстрували залежності між академічними досягненнями, психологічними характеристиками та професійними рекомендаціями. Наприклад, виявлялися закономірності між високими оцінками з математики й технічними спеціальностями або між розвиненим емоційним інтелектом і гуманітарними професіями. Крім того, були розроблені рекомендації, які базуються на результатах роботи моделей і можуть бути використані як інструмент для допомоги учням.

Усі ці етапи інтегруються в єдину методику, яка забезпечує послідовність і системність дослідження. Розроблені методики дозволяють аналізувати різноманітні характеристики учнів, будувати моделі з високою точністю та

формулювати рекомендації, які відповідають індивідуальним потребам кожного учня. Це робить дослідження не лише науково значущим, але й практично корисним, оскільки отримані результати можуть бути впроваджені в системи профорієнтації для автоматизації цього процесу.

4.2 Формування залежностей між параметрами об’єкту дослідження професійної області

Під час дослідження було проаналізовано, як різні параметри, що описують старшокласників, впливають на їхню професійну спрямованість. Для цього ми зосередилися на встановленні залежностей між академічними показниками, особистісними характеристиками, соціальною активністю, а також рівнем зацікавленості у певних сферах діяльності. Аналіз цих зв’язків дозволив створити основу для моделювання та прогнозування професійної траєкторії.

Для дослідження були відібрані параметри, які є найважливішими для аналізу професійної орієнтації старшокласників. Академічні показники, такі як оцінки з математики, фізики, історії та інших предметів, використовувалися для оцінки здібностей у конкретних професійних галузях. Особистісні характеристики, такі як логічне мислення, творчість і емоційний інтелект, були включені для врахування індивідуальних особливостей студентів. Рівень соціальної активності, включаючи участь у позакласних заходах і наявність підробітку, використовувався для визначення загальної залученості студента у навчальний процес. Ці параметри обрані через їхній прямий вплив на професійні успіхи та адаптивність у різних сферах діяльності.

Для встановлення залежностей між параметрами об’єкту дослідження ми використовували різні методи. Кореляційний аналіз допоміг визначити, які параметри мають сильний взаємозв’язок, наприклад, між оцінками з технічних предметів і інтересом до науки. Регресійний аналіз дозволив побудувати точні функціональні залежності для прогнозування середнього бала чи ймовірності успіху в обраній професії. Крім цього, ми розробили власні метрики, такі як

tech_index, humanities_index і medical_index, які враховують не лише оцінки, але й рівень інтересу студента до певних галузей.

Аналіз показав, що високі оцінки з математики та фізики найбільш значущі для технічних професій, таких як інженерія чи програмування. Історія та англійська мова, навпаки, мають більший вплив на гуманітарні професії. Особистісні характеристики, наприклад логічне мислення, значно впливають на професійну траєкторію у технічній галузі, тоді як творчість є важливішою для гуманітарних напрямків. Крім того, було встановлено, що студенти з високим рівнем соціальної активності, зокрема участю у позакласних заходах, демонструють кращі загальні результати.

Обрані параметри ґрунтуються на численних дослідженнях, які підкреслюють зв’язок між академічною успішністю, інтересами та професійними результатами. Поєднання оцінок, особистісних характеристик та соціальної активності дозволяє створити комплексний профіль кожного студента, що робить прогнозування більш точним і релевантним. Це дозволило врахувати як об’єктивні, так і суб’єктивні фактори, які впливають на вибір професійної траєкторії.

Таким чином, сформовані залежності стали базою для побудови моделей класифікації та регресії, які забезпечують точне прогнозування і допомагають студентам зробити обґрунтований вибір майбутньої професії.

4.3 Метрики оцінювання результатів дослідження

Для оцінки якості моделей, розроблених у межах цього дослідження, було використано декілька метрик. Їх вибір був зумовлений специфікою задач, які вирішувались у роботі: класифікація для визначення професійної траєкторії та регресія для прогнозування числових показників, таких як середній бал і ймовірність успішності. Метрики дозволили всебічно оцінити якість моделей, враховуючи точність, стабільність і прогнозну спроможність.

Для задач класифікації основними метриками оцінки стали:

− точність (Accuracy): Ця метрика використовується для оцінки загальної кількості правильних передбачень моделі, що дозволяє зрозуміти її ефективність на повному наборі даних;

− f1-міра: Важлива для оцінки якості роботи моделі у випадках, коли дані є незбалансованими. Вона враховує як точність передбачень, так і їх повноту, забезпечуючи більш об’єктивну оцінку;

− матриця плутанини (Confusion Matrix): Використовувалася для детального аналізу помилок моделі, дозволяючи зрозуміти, які класи найчастіше плутаються між собою.

Для задач прогнозування числових значень були обрані наступні метрики:

− середньоквадратична помилка (MSE): Дозволила оцінити середню величину відхилення між реальними та передбаченими значеннями. Вона була корисною для виявлення значних похибок у прогнозах;

− r^2 (коефіцієнт детермінації): Використовувався для оцінки того, наскільки модель здатна пояснити варіації у залежній змінній. Ця метрика стала ключовим показником точності для регресійних моделей;

− середня абсолютна помилка (MAE): Показала середню похибку прогнозів моделі у зрозумілому та інтуїтивному форматі, що було корисним для практичної інтерпретації результатів.

Метрики були обрані з урахуванням особливостей задач дослідження. Для класифікації ми враховували необхідність точної оцінки якості передбачень у випадках із незбалансованими класами, що зробило F1-міру та матрицю плутанини важливими інструментами. У регресійних задачах ключовою була можливість оцінити як середню точність прогнозів, так і стабільність моделі на тестових даних, для чого R^2 став найкращим вибором. Середньоквадратична та абсолютна помилки дозволили більш глибоко зрозуміти похибки моделі й порівняти їх між різними підходами.

Таким чином, використані метрики забезпечили всебічну оцінку моделей, дозволивши вибрати найбільш ефективні підходи для вирішення задач дослідження.

4.4 Формулювання отриманих результатів дослідження

У ході виконання дослідження було реалізовано та проаналізовано три основні підходи машинного навчання: класифікацію, регресію та кластеризацію. Кожен із методів оцінено за його придатністю для розв'язання задач профорієнтації старшокласників. Основна мета полягала в тому, щоб оцінити сильні та слабкі сторони кожного підходу, а також визначити найбільш ефективні моделі для практичного застосування.

Моделі класифікації були спрямовані на визначення професійної траєкторії учнів на основі їхніх академічних успіхів, інтересів та інших параметрів. Зокрема, було протестовано моделі LogisticRegression, RandomForestClassifier, SVC, DecisionTreeClassifier, GaussianNB, GradientBoostingClassifier, та KNeighborsClassifier. Детальні результати дослідження методів класифікації представлено в таблиці 4.1.

Таблиця 4.1 – Результати класифікації за обраними метриками

Модель	Accuracy	Precision	Recall	F1-Score
LogisticRegression	74.72%	0.75	0.75	0.75
RandomForestClassifier	77.43%	0.78	0.77	0.77
SVC	76.31%	0.77	0.76	0.76
DecisionTreeClassifier	75.01%	0.75	0.75	0.75
GradientBoostingClassifier	75.43%	0.76	0.75	0.75
KNeighborsClassifier	74.84%	0.76	0.75	0.75
GaussianNB	74.02%	0.75	0.74	0.74

Також, на основі отриманих даних, нами було сформовано стовбчасту діаграму, яка наглядно відображає точність кожної моделі, згідно параметру accuracy. Діаграму зображено на рисунку 4.1.

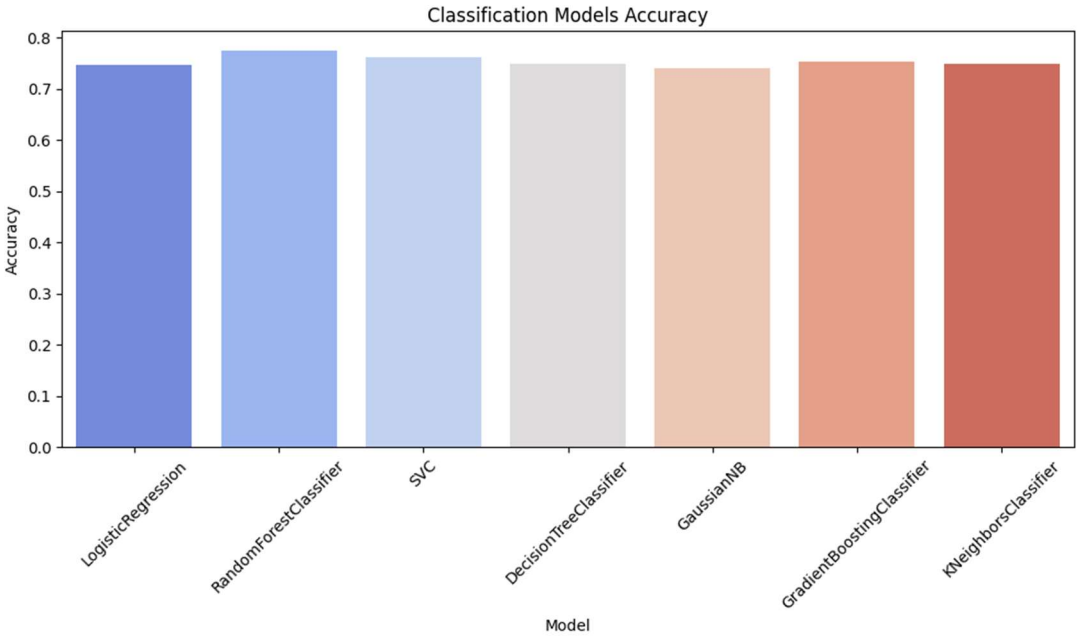

Рисунок 4.1 – Діаграма точності алгоритмів класифікації

Виходячи з результатів дослідження методів класифікації, було сформовано наступні висновки:

− randomForestClassifier досягла найвищої точності (Accuracy = 77.43%) серед усіх моделей. Цей алгоритм показав стабільні результати за метриками F1-міри, точності та повноти;

− svc і GradientBoostingClassifier також продемонстрували високі результати, але потребували більше часу на навчання;

− класифікація дозволила автоматизувати визначення потенційної професійної області, однак точність для менш представлених класів залишалася нижчою. Це свідчить про необхідність додаткового балансування даних.

Алгоритми регресії були використані для прогнозування числових параметрів, зокрема середнього академічного балу та ймовірності успішності в обраній професії. Були протестовані такі моделі, як LinearRegression, Ridge, Lasso, RandomForestRegressor, GradientBoostingRegressor. Детальні результати дослідження методів регресії представлено в таблиці 4.1.

Таблиця 4.2 – Результати регресії за обраними метриками

Модель	Середній R^2	R^2 на тестовому наборі
LinearRegression	1.0000	1.0000
Ridge	1.0000	1.0000
Lasso	0.6946	0.6934
RandomForestRegressor	0.9527	0.9571
GradientBoostingRegressor	0.9761	0.9782

Для наглядного відображення точності прогнозування алгоритмів регресії, нами було сформовано точковий графік, який зображено на рисунку 4.2.

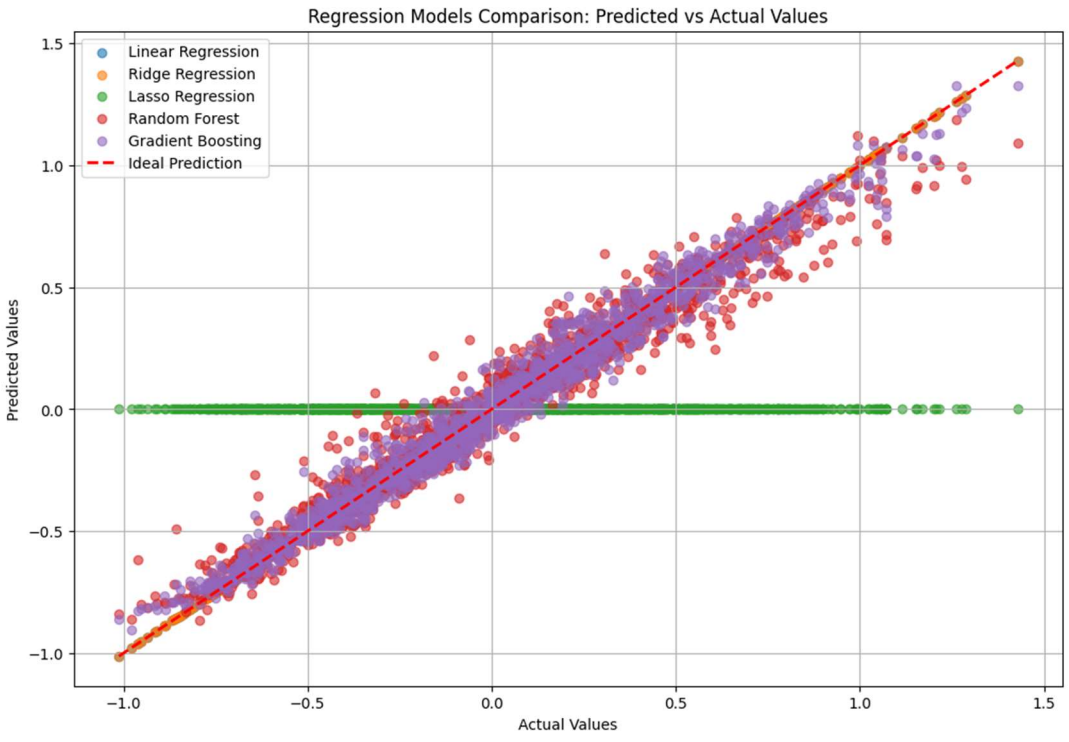

Рисунок 4.2 – Графік точності алгоритмів регресії

Виходячи з результатів дослідження методів регресії, було сформовано наступні висновки:

− моделі LinearRegression та Ridge показали ідеальні результати (R^2 = 1.0000), однак це може свідчити про перенавчання, оскільки тестовий набір не відображає складність реальних даних;

− gradientBoostingRegressor досяг збалансованих показників із $R^2 = 0.9782$, що робить його оптимальним вибором для регресійних задач;

− lasso продемонстрував середні результати ($R^2 = 0.6934$), що вказує на його обмежену ефективність у задачах із високою залежністю параметрів.

Кластеризація була застосована для поділу старшокласників на групи залежно від їхніх інтересів, рівня активності та академічних досягнень. Для цього були реалізовані підходи KMeans, DBSCAN та власна формула кластеризації. Детальні результати дослідження методів кластеризації представлено в таблиці 4.3.

Таблиця 4.3 – Результати регресії за обраними метриками

Метод	Кількість кластерів	Розподіл кластерів
KMeans	4	[2457, 2212, 1745, 1586]
DBSCAN	1	Усі об’єкти віднесено до одного кластеру
Custom Clustering	0	Всі значення відсутні

Для наглядного відображення розподілу кластеризації, нами було сформовано точковий графік, який зображено на рисунку 4.3.

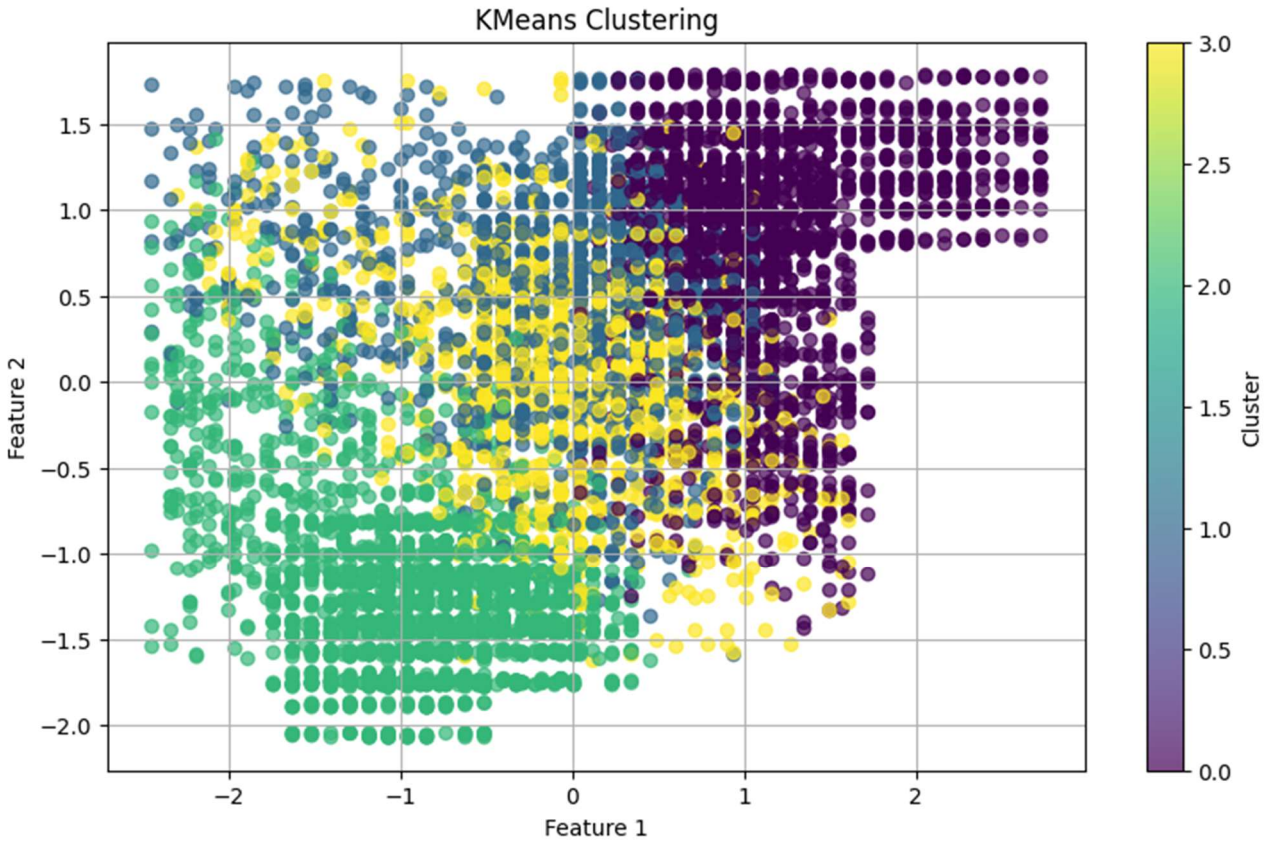

Рисунок 4.3 – Графік точності алгоритму кластеризації

Виходячи з результатів дослідження методів кластеризації, було сформовано наступні висновки:

− KMeans успішно розподілив дані на чотири чітко виражені групи (кластерів), що дозволяє виокремлювати типові профілі старшокласників;

− DBSCAN відніс усі об'єкти до одного кластеру, що свідчить про недостатню варіативність даних або некоректно підібрані гіперпараметри;

− власна формула кластеризації не змогла створити значущих кластерів, що вказує на необхідність її вдосконалення.

4.5 Інтерпритація отриманих результатів з точки зору наукової та технічної цінності

У результаті проведеного дослідження було оцінено ефективність трьох основних напрямків машинного навчання — класифікації, регресії та кластеризації — для задач профорієнтації старшокласників. Кожен метод

продемонстрував свої переваги та недоліки, що дозволило визначити їхню придатність для конкретних цілей.

Класифікаційні моделі, зокрема RandomForestClassifier, показали високу точність у прогнозуванні найбільш відповідної професійної траєкторії для старшокласників. Це свідчить про те, що класифікація є ефективним підходом для вирішення задач, де важливо чітко визначити категорії або класи, до яких належить об'єкт.

Регресія виявилася корисною для прогнозування числових показників, таких як середній академічний бал та ймовірність успішності в професійній діяльності. Найкращі результати показала модель GradientBoostingRegressor, яка досягла високого рівня точності та стабільності. Це дозволяє використовувати регресію для оцінки успіху та ефективності різних навчальних траєкторій.

Кластеризація допомогла виявити структуру та групи серед старшокласників на основі їхніх інтересів, рівня залученості та академічних показників. Метод KMeans показав найкращі результати, створивши чіткі та логічні групи. Водночас інші підходи, такі як DBSCAN та кастомна формула, мали обмежену ефективність через недостатню варіативність даних або некоректно підібрані параметри.

Результати дослідження базуються на сучасних методах машинного навчання та аналізу даних. Для забезпечення достовірності було проведено:

− крос-валідацію моделей, дозволило уникнути перенавчання та оцінити узагальнюючу здатність моделей на різних наборах даних. Зокрема, найкращі результати були отримані для GradientBoostingClassifier у класифікації та GradientBoostingRegressor у регресії, що підтверджує точність і надійність обраних підходів;

− аналіз кореляцій та вибір релевантних ознак для видалення надмірно корельованих ознак, що забезпечило високу якість даних і точність моделей;

− репрезентативність даних, що охоплювали різні параметри старшокласників, такі як: академічні оцінки, позашкільні активності, психологічні індекси, що дозволило створити багатовимірні моделі.

Наукова новизна нашої роботи полягає у застосуванні інтегрованого підходу для аналізу багатовимірних даних старшокласників, який об'єднує класифікацію, регресію та кластеризацію. Це дозволяє вирішувати задачі різного рівня складності та отримувати як точкові, так і узагальнені результати.

Одним із ключових внесків є розробка специфічних метрик, таких як engagement_index, tech_index, humanities_index та medical_index, які дозволяють оцінити навчальні та професійні якості учнів. Ці метрики можуть бути адаптовані для інших сфер, наприклад, у медичній чи соціальній аналітиці.

Також важливо відзначити, що запропоновані математичні моделі та алгоритми обробки даних базуються на сучасних методах машинного навчання, що забезпечує їхню релевантність у контексті сучасних наукових досліджень.

Практична значущість нашої роботи проявляється у можливості реального використання розроблених моделей для підтримки освітніх і кар'єрних рішень:

− освітня галузь: Запропоновані моделі можуть використовуватися в школах для визначення сильних і слабких сторін учнів та допомогати в формуванні персоналізованих навчальних траєкторій;

− кар’єрне консультування: Система може допомагти виявляти найбільш підходящі професійні напрями для учнів на основі аналізу їхніх академічних успіхів, інтересів і позашкільної активності;

− соціально-економічна сфера: Використання таких систем може зменшити випадки професійної невідповідності, сприяти економії ресурсів на перепідготовку та оптимізувати використання інтелектуального потенціалу.

Економічна доцільність роботи полягає в її здатності автоматизувати процеси профорієнтації та аналізу даних, зменшити витрати на індивідуальні консультації та оптимізувати ресурси освітніх установ. Масштабованість запропонованих підходів дозволяє адаптувати їх до різних умов і регіонів, розширюючи можливості використання.

Отже, наше дослідження демонструє як наукову, так і практичну цінність застосування методів машинного навчання для аналізу та підтримки прийняття рішень у профорієнтації. Розроблені моделі показують високі результати за

основними метриками, що підтверджує їхню ефективність. У майбутньому такі підходи можуть бути вдосконалені та масштабовані для використання у різних галузях. Це дослідження може бути першим кроком уперед у вирішенні актуальних задач освіти та соціального розвитку.

5 АНАЛІЗ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ІННОВАЦІЇ

5.1 Розрахунок собівартості програмної інновації

Для виконання розрахунку були використані початкові дані, представлені в таблиці 5.1

Таблиця 5.1 — Початкові дані для визначення собівартості

Найменування початкових даних	Показник	Джерело отримання
Трудомісткість складання програми, днів	40	Фактичні витрати часу (2 місяці по 20 робочих днів)
Місячна ставка розробника, грн	15000	Ринковий рівень зарплат розробників
Кількість годин в місяці, год	160	Кількість робочих годин на місяць
Додаткова зарплата (%)	15	Додаткові витрати (премії, бонуси)
Відрахування до соціальних фондів (%)	22	Законодавчі норми
Загальновиробничі витрати (%)	80	Оренда, обладнання, підтримка
ПДВ (%)	20	Податок на додану вартість

Робимо розрахунок на 1 місяць(20 днів):

Стаття 1. Комплектуючі вироби – 1 компакт-диск :

$$З_к = \sum Ц_к * n_к, \quad (5.1)$$

де $З_к$ – витрати на комплектуючі вироби, грн.;

$Ц_к$ – ціна за 1 одиницю комплектуючих виробів, грн.;

$n_к$ – кількість комплектуючих виробів по кожному типорозмірі, шт

Визначимо витрати на комплектуючі вироби за формулою 5.2:

(5.2)

$$З_к = \sum 7{,}60 * 1 = 7{,}60 \text{ грн.}$$

Витрати на комплектуючі виробі склали 7,60 грн.

Стаття 2. Витрати на електроенергію розраховуємо за формулою 5.3:

$$B_E = P_E \sum Wi * t_{\text{шт } i}, \tag{5.3}$$

де P_E – ціна за 1 кВт-год, грн.;

$t_{\text{шт } i}$ – кількість годин в місяці;

Wi – середня потужність, що споживається.

Визначимо витрати на електроенергію за формулою 5.4:

$$B_E = 4{,}32 * \sum 0{,}250 * 160 = 172{,}8 \text{ грн} \tag{5.4}$$

Витрати на електроенергію дорівнюють 172,8 грн.

Стаття 3. Основна заробітна плата визначається за формулою 5.5:

$$З_{\text{осн}} = l_{\text{год}} * T_{\text{год}}, \tag{5.5}$$

де $l_{\text{год}}$ – годинна тарифна ставка програміста, грн.;

$T_{\text{год}}$ – кількість годин у місяці.

Визначаємо годинну тарифну ставку укладача ПЗ за формулою 5.6:

$$З_{\text{осн}} = 93{,}750 * 160 = 15000 \tag{5.6}$$

Стаття 4. Додаткова заробітна плата визначається за формулою 5.7:

(5.7)

$$З_{дод} = \frac{З_{осн} * Д\%}{100},$$

де Д – відсоток додаткової заробітної плати.

Визначимо додаткову заробітну плату за формулою 5.8:

$$З_{дод} = \frac{15000 * 15}{100} = 2250 \text{ грн} \quad (5.8)$$

Додаткова заробітна плата дорівнює 2250,00 грн.

Стаття 5. Відрахування в соціальні фонди знайдемо за формулою 5.9:

$$З_{соц} = \frac{(З_{осн} + З_{доп}) * С\%}{100}, \quad (5.9)$$

де С% — відсоток відрахувань у соціальні фонди.

Визначимо відрахування в соціальні фонди за формулою 5.10:

$$З_{соц} = \frac{(15000 + 2250) * 22}{100} = 3795{,}0 \text{ грн} \quad (5.10)$$

Відрахування в соціальні фонди дорівнює 3795,0 грн;

Стаття 6. Загальновиробничі витрати визначаються за формулою 5.11:

$$З_{заг} = \frac{З_{осн} * Н_1\%}{100}, \quad (5.11)$$

де $Н_1\%$ — відсоток загальновиробничих витрат.

Визначимо загальновиробничі витрати за формулою 5.12:

$$З_{заг} = \frac{15000 * 80}{100} = 12000 \text{ грн} \quad (5.12)$$

Загальновиробничі витрати дорівнюють 12000,00 грн.

Визначимо виробничу собівартість склавши всі попередні розрахунки (формула 5.13):

$$S_{пп} = 172{,}8 + 7{,}60 + 15000 + 2250 + 3795{,}0 + 12000 = 33225{,}4 \text{ грн} \quad (5.13)$$

Виробнича собівартість дорівнює 33225,4 грн.

В таблиці 5.2 наведено планову калькуляцію виробничої собівартості, ціни підприємства й ціни для замовника на виконання розробки програми.

Таблиця 5.2 – Планова калькуляція

Статті калькуляції	Сума, грн.
Стаття 1 Комплектуючи вироби	7,60
Стаття 2. Витрати на електроенергію:	172,8
Стаття 3 Основна заробітна плата	15000
Стаття 4 Додаткова заробітна плата	2250
Стаття 5 Відрахування в соціальні фонди	3795,0
Стаття 6 Загальновиробничі витрати	12000
Виробнича собівартість, 1 місяць	33225,4
Собівартість ПЗ	66450,8

Робота по розробці ПЗ проводилась протягом двох місяців, таким чином розрахунок показав, що собівартість АС складає 66450,8 грн

5.2 Розрахунок ефективності впровадження програмної інновації

Розрахунок ефективності впровадження алгоритмів профорієнтації для старшокласників є комплексним завданням, яке враховує як технічні, так і соціально-економічні аспекти. В основу аналізу покладено оцінку ризиків, пов'язаних із впровадженням розробленої системи, а також її потенційну економічну та практичну доцільність.

Одним з основних ризиків може бути недостатня точності роботи системи класифікації, в результаті чого можливі втрати через неправильну профорієнтацію учнів. Витрати такого ризику визначаються за формулою 5.14:

$$В_к = n_{нп} * C_в, \quad (5.14)$$

де $В_к$ – витрати від недостатньої точності класифікації, грн;

$n_{нп}$ –учні із неправильними рекомендаціями, кількість;

$C_в$ – середня вартість додаткових консультацій та матеріалів, грн.

Наприклад, при 340 учнях, які отримали неправилий напрямок та вартості додаткової консультації в 500 грн, витрати становлять (формула 5.15):

$$В_к = 340 * 500 = 170\,000 \text{ грн}, \quad (5.15)$$

Другий ризик полягає в тому, що низький рівень регресії може викликати похибки в оцінці успішності учні. Він розраховується за формулою 5.16:

$$В_р = Δ_{усп} * n_{уч} * C_{пер}, \quad (5.16)$$

де $В_р$ – витрати від низького рівня регресії, грн;

$Δ_{усп}$ –середнє відхилення прогнозу успішності, %;

$n_{уч}$ – загальна кількість учнів, кількість.

$C_{пер}$ – витрати на перенавчання одного учня, грн;

Наприклад, при 1700 учнях, відхиленні в 1% успішності прогнозу регресії та вартості в 300 грн на перенавчання одного учня, ми отримуємо наступні витрати, які розраховуються за формулою 5.17:

$$\mathrm{B_р} = 0.1 * 1700 * 300 = 51\,000 \text{ грн}, \quad (5.17)$$

Правильно налаштована модель та система рекомендацій, знижує ризик помилкових рекомендацій та призводить до економії часу для проведеня профорієнтаційних консультацій на одного учня з 2 годин до 30 хвилин. Розрахунок економії часу для проведеня профорієнтаційної консультації розраховується за формулою 5.18:

$$\mathrm{E_{час}} = (T_{\text{б}} - T_{\text{а}}) * n * \mathrm{C_г}, \quad (5.18)$$

де $\mathrm{E_{час}}$ – економія часу на проведення профорієнтації, грн;

$T_{\text{б}}$– час витрачений без автоматизації, год;

$T_{\text{а}}$– час витрачений з автоматизацією, год;

n – загальна кількість учнів, кількість.

$C_{\text{г}}$ – ставка консультатна для проведення профорієнтації, грн/год;

Наприклад, для кількості учнв в 1700 осіб, часу проведення консультації без автоматизації в 2 години, а з системою автоматизації в 30 хвилин та ставкою консультанта в 150 грн на годину, ми отримаємо наступну фінансову економію часу (формула 5.19

$$\mathrm{E_{час}} = (2 - 0.5) * 1700 * 150 = 382\,500\text{грн}, \quad (5.18)$$

Загальний економічний ефект обчислюється за формулою 5.19:

$$Е_{заг} = Е_{час} + (В_к - В_р), \quad (5.19)$$

Підставляючи значення (формула 5.20):

$$Е_{заг} = 382\,500 - (170\,000 + 51\,000) = 82\,500 грн. \quad (5.20)$$

Термін окупності визначається за формулою:

$$Т = \frac{С}{E_{заг}/Ч},$$

де С – собівартість інтеграції архітектури, грн;

Ч – період, протягом якого отримується економічний ефект, місяців.

При собівартості 66450,8 грн. і річному періоді (12 місяців) термін окупності дорівнює:

$$Т = \frac{66450{,}8}{82500 / 12} = \frac{66450{,}8}{6875} \approx 9{,}66 \text{ місяців}$$

Отримані результати демонструють, що впровадження автоматизованої системи профорієнтації є економічно вигідним рішенням. Зниження часових витрат на проведення консультацій, значна економія коштів за рахунок оптимізації процесу та мінімізація ризиків, пов’язаних із людськими помилками, забезпечують швидку окупність інвестицій у розробку системи. Розрахунки показують, що економічний ефект суттєво перевищує витрати, а система здатна окупитися за декілька місяців. Це свідчить про високий потенціал практичної цінності та перспективність впровадження системи у навчальних закладах.

ВИСНОВКИ

У ході виконання роботи було розроблено комплексну систему автоматизації профорієнтації старшокласників із використанням методів машинного навчання, класифікації, регресії та кластеризації. Проведений аналіз результатів показав, що найстабільнішою та найбільш придатною для практичної реалізації є класифікація, тоді як регресія та кластеризація наразі не забезпечують достатньої точності та адаптивності для повноцінного застосування у профорієнтаційній діяльності.

Методи класифікації, зокрема логістична регресія, RandomForest та SVC, продемонстрували високу точність і стабільність у прогнозуванні професійної спрямованості учнів. Середній рівень точності класифікаційних моделей становив 75–78%, що дозволяє з достатнім ступенем впевненості робити висновки про професійні інтереси учнів. Ці методи є придатними для практичного застосування, оскільки забезпечують високу швидкість обробки даних і прийняття рішень.

Хоча регресійні моделі показали високі значення R^2 (до 1.0 для LinearRegression), їхня застосовність у контексті профорієнтації є обмеженою. Причиною цього є залежність від якісного числового представлення даних, яке не завжди коректно відображає складність професійних інтересів учнів. Крім того, модель перенавчалася на доступних даних, що створює ризик низької узагальнювальної здатності в реальних умовах.

Методи кластеризації, такі як KMeans та DBSCAN, показали неоднозначні результати. Хоча KMeans дозволяє виявити певні групи учнів за схожими характеристиками, цей підхід не забезпечує достатньої інтерпретованості отриманих кластерів у контексті профорієнтації. DBSCAN показав низьку варіативність (усі дані належать до одного кластеру), що свідчить про його низьку ефективність для аналізу профорієнтаційних даних. Custom Score кластеризація не виявила значущих результатів через складність обробки багатовимірних профільних даних.

Результати дослідження показали, що інтеграція автоматизованої системи профорієнтації є технічно можливою, але потребує значних ресурсів для подальшої оптимізації. Зокрема, класифікаційні моделі можна впроваджувати вже на поточному етапі, тоді як регресійні та кластеризаційні підходи потребують додаткових досліджень і доопрацювань.

В ході дослідження було виявлено наступні обмеження розроблювальної системи:

− наявний набір даних має обмежений обсяг і різноманітність, що знижує узагальнювальну здатність моделей;

− складність профорієнтації. Інтереси учнів є багатовимірними та динамічними, що ускладнює їх точне моделювання за допомогою регресії та кластеризації;

− інтерпретація результатів. Методи кластеризації не завжди забезпечують зрозумілу та корисну інформацію для кінцевих користувачів.

В ході дослідження було виявлено наступні перспективи розроблювальної системи:

− збір і аналіз більшого обсягу даних, включаючи різноманітні соціально-економічні та психологічні параметри учнів;

− використання складніших алгоритмів (наприклад, нейронних мереж) для обробки багатовимірних даних;

− розробка адаптивної системи, яка дозволить враховувати індивідуальні особливості учнів у реальному часі;

− необхідність додаткових ресурсів.

Для повноцінного масштабування та впровадження розробленої системи автоматизації профорієнтації необхідно врахувати ряд обмежень і забезпечити додаткові ресурси. Хоча система демонструє перспективність, її ефективність у реальних умовах вимагає подальших досліджень і оптимізації. Підвищення точності та адаптивності моделей потребує часу для доопрацювання алгоритмів, а також фінансових інвестицій для розширення інфраструктури, збору та аналізу великих обсягів даних. Крім того, необхідна участь фахівців з освітніх

технологій, психології та програмування для адаптації системи до потреб освітніх установ та учнів.

Незважаючи на наявні обмеження, результати дослідження свідчать про значний потенціал системи. Подальші зусилля, спрямовані на розширення бази даних, удосконалення алгоритмів і забезпечення інтерпретації результатів для кінцевих користувачів, здатні перетворити її на ефективний інструмент підтримки молоді у виборі професії. Таким чином, інвестування часу, ресурсів та експертної підтримки дозволить реалізувати систему на практиці, зробивши її вагомим внеском у сучасну освіту та профорієнтацію.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Holland, J. L. Making Vocational Choices: A Theory of Vocational Personalities and Work Environments. – Psychological Assessment Resources, 1997. – 305 p.

2. Brown, D., & Lent, R. W. Career Development and Counseling: Putting Theory and Research to Work. – Wiley, 2012. – 544 p.

3. Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach. – Pearson, 2020. – 1152 p.

4. Shalev-Shwartz, S., & Ben-David, S. Understanding Machine Learning: From Theory to Algorithms. – Cambridge University Press, 2014. – 411 p.

5. Підготовка старшокласників до вступу до ЗВО соціально-психологічний вимір [Електронний ресурс] / Режим доступу: https://dspace.udpu.edu.ua/bitstream/123456789/15323/1/Підготовка%20старшокласників%20до%20вступу%20до%20ЗВО%20соціально-психологічний%20вимір.pdf

6. Математическая модель. [Електронний ресурс] / Режим доступу: http://www.orenipk.ru/kp/distant_vk/docs/2_1_1/inf/inf_mat_mod.html

7. Заболотний С. В. Основи аналізу даних у сучасних інформаційних системах. – Київ: Вид-во "Кондор", 2017. – 198 с.

8. Шахновський С. М. Метричні методи оцінки ефективності алгоритмів. – Харків: Вид-во "Ранок", 2019. – 230 с.

9. Чеберячко О. Є., Савченко В. Г. Оцінювання ефективності програмних рішень у машинному навчанні. – Збірник наукових праць НТУУ "КПІ", 2021. – Вип. 12, с. 45–60.

10. Ігнатенко М. Ю. Штучний інтелект: сучасні підходи та застосування в освіті. – Львів: ЛНУ ім. Івана Франка, 2019. – 312 с.

11. Тимченко О. А. Методи аналізу даних у задачах профорієнтації. – Освіта і кар’єра, 2020. – Вип. 3, с. 33–49.

12. Ryabtsev, Alexander. 8 Reasons Why Python is Good for Artificial Intelligence and Machine Learning. Software Development Blog & IT Tech Insights | Django Stars. – [Електронний ресурс]. – Режим доступу: https://djangostars.com/blog/8-reasons-why-python-is-good-for-ai-and-ml/. – Дата звернення: 9 жовтня 2023. Архівовано: 18 жовтня 2023.

13. Oracle. Programming language Java 14 is out with these 16 major feature improvements. – ZDnet. – [Електронний ресурс]. – Режим доступу: https://www.zdnet.com/article/programming-language-java-14/. – Архів оригіналу: 19 березня 2020.

14. Сергієнко І. В. Інтелектуальний аналіз даних з використанням Python та його бібліотек. – Харків: Вид-во "Основа", 2021. – 300 с.

15. Сергієнко І. В. Веб-сервіси та API: основи інтеграції у системах аналізу даних. – Київ: Наукова думка, 2019. – 215 с.

16. Grolinger, K., & Capretz, M. A. M. Data-as-a-Service: Integrating APIs in Big Data Ecosystems. – Future Generation Computer Systems, 2021. – Vol. 120, pp. 285–299.

17. Микитюк П. П. Інноваційний менеджмент: Навчальний посібник. – Тернопіль: Економічна думка, 2006. – 295 с.

18. Йохна М. А., Стадник В. В. Економіка і організація інноваційної діяльності. – Навч. посібник. – Київ, 2005.

ДОДАТОК А

Код програми

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler

import warnings
# Ignore all warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('/content/student_data_ukr.csv')


df.tail()
df.drop(columns=['first_name','last_name','email'], axis=1,
inplace=True)
df.head()
# Шаг 1: Кодирование категориальных признаков
label_encoder = LabelEncoder()
df['gender'] = label_encoder.fit_transform(df['gender'])
df['part_time_job'] =
label_encoder.fit_transform(df['part_time_job'])
df['logical_reasoning'] =
label_encoder.fit_transform(df['logical_reasoning'])
df['creativity'] = label_encoder.fit_transform(df['creativity'])
df['emotional_intelligence'] =
label_encoder.fit_transform(df['emotional_intelligence'])
df['extracurricular_activities'] =
df['extracurricular_activities'].apply(lambda x: 1 if x else 0)
df['career_aspiration_encoded'] =
label_encoder.fit_transform(df['career_aspiration'])
# Шаг 2: Нормализация оценок
```

```
score_columns = ['math_score', 'history_score', 'physics_score',
'chemistry_score',
                 'biology_score', 'english_score',
'geography_score']
scaler = StandardScaler()
df[score_columns] = scaler.fit_transform(df[score_columns])


# Шаг 3: Добавление total_score и average_score
df['total_score'] = df[score_columns].sum(axis=1)
df['average_score'] = df['total_score'] / len(score_columns)


# Просмотр результата
df.head()


# Шаг 5. Расчёт дополнительных метрик (индексов)


# Функция для расчета индекса вовлеченности
def calculate_engagement_index(part_time_job,
extracurricular_activities, weekly_self_study_hours):
    job_weight = 0.3 if part_time_job == 1 else 0
    activities_weight = 0.4 * extracurricular_activities
    study_weight = 0.3 * weekly_self_study_hours
    return job_weight + activities_weight + study_weight


# Функции для расчета профессиональных индексов
def calculate_tech_index(math_score, physics_score,
interest_science_tech):
    return 0.5 * math_score + 0.3 * physics_score + 0.2 *
interest_science_tech


def calculate_humanities_index(history_score, english_score,
interest_art_humanities):
    return 0.4 * history_score + 0.4 * english_score + 0.2 *
interest_art_humanities


def calculate_medical_index(biology_score, chemistry_score,
weekly_self_study_hours):
```

```
    return 0.4 * biology_score + 0.3 * chemistry_score + 0.3 *
weekly_self_study_hours


# Применение функций для добавления новых метрик в DataFrame

df['engagement_index'] = df.apply(lambda row:
calculate_engagement_index(

    row['part_time_job'], row['extracurricular_activities'],
row['weekly_self_study_hours']), axis=1)


df['tech_index'] = df.apply(lambda row: calculate_tech_index(

    row['math_score'], row['physics_score'],
row['interest_science_tech']), axis=1)


df['humanities_index'] = df.apply(lambda row:
calculate_humanities_index(

    row['history_score'], row['english_score'],
row['interest_art_humanities']), axis=1)


df['medical_index'] = df.apply(lambda row:
calculate_medical_index(

    row['biology_score'], row['chemistry_score'],
row['weekly_self_study_hours']), axis=1)


# Просмотр результата

df.head()


 # Calculate the correlation matrix for selected numerical columns

correlation_matrix = df[['gender', 'career_aspiration_encoded',
'weekly_self_study_hours', 'extracurricular_activities',
'absence_days', 'part_time_job', 'engagement_index', 'tech_index',
'humanities_index', 'medical_index']].corr()


# Create a heatmap to visualize the correlation matrix

plt.figure(figsize=(10, 8))

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f", linewidths=0.5)

plt.title('Correlation Matrix')

plt.show()
```

```
X =
df.drop(columns=['career_aspiration','career_aspiration_encoded'])

y = df['career_aspiration_encoded']

from imblearn.over_sampling import SMOTE


# Create SMOTE object


smote = SMOTE(random_state=42)


# Resample the dataset using SMOTE

X_resampled, y_resampled = smote.fit_resample(X, y)

X_resampled.shape

y_resampled.shape

#split data

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(X_resampled,y_resam
pled,test_size=0.2,random_state=0 )

x_train.shape

y_train.shape


x_test.shape

y_test.shape


x_train.describe()



from sklearn.preprocessing import StandardScaler


# Initialize the StandardScaler

scaler = StandardScaler()


# Fit the scaler to the training data and transform both training
and testing data

X_train_scaled = scaler.fit_transform(x_train)

X_test_scaled = scaler.transform(x_test)
```

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier

# Define a list of classifiers
classifiers = [
    LogisticRegression(),
    RandomForestClassifier(),
    SVC(),
    DecisionTreeClassifier(),
    GaussianNB(),
    GradientBoostingClassifier(),
    KNeighborsClassifier()  # Add KNN to the list of classifiers
]

# Iterate over each classifier
for classifier in classifiers:
    # Train the classifier
    classifier.fit(X_train_scaled, y_train)

    # Make predictions
    y_pred = classifier.predict(X_test_scaled)

    # Calculate accuracy
    accuracy = accuracy_score(y_test, y_pred)

    # Print the accuracy
```

```
    print(f"Classifier: {classifier.__class__.__name__}")
    print(f"Accuracy: {accuracy:.4f}")

    # Print the classification report
    print("Classification Report:")
    print(classification_report(y_test, y_pred))

    # Print the confusion matrix
    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))
    print("="*50)

# Initialize the RandomForestClassifier model
rf_model = RandomForestClassifier()

# Train the model on the training set
rf_model.fit(x_train, y_train)

# Make predictions on the testing set
y_test_pred_rf = rf_model.predict(x_test)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_test_pred_rf))

# Display confusion matrix and classification report for testing
set
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_test_pred_rf))

print("\nClassification Report:")
print(classification_report(y_test, y_test_pred_rf))

#test 1
prediction1 = rf_model.predict(x_test.iloc[[1]])
```

```
print("Predicted Label :",prediction1)
print("Actual Label:", y_test.iloc[1])

#test 2
prediction2 = rf_model.predict(x_test.iloc[[12]])
print("Predicted Label :",prediction2)
print("Actual Label:", y_test.iloc[12])
# Predict labels using the random forest model
predicted_labels = rf_model.predict(x_test)

# Convert the NumPy array of predicted labels to a pandas Series
predicted_series = pd.Series(predicted_labels)

# Get value counts of actual labels
actual_counts = y_test.value_counts()

# Get value counts of predicted labels
predicted_counts = predicted_series.value_counts()

# Create a bar plot
plt.figure(figsize=(10, 5))

# Plot actual labels
actual_counts.plot(kind='bar', color='blue', width=0.4,
position=1, label='Actual Labels')

# Plot predicted labels
predicted_counts.plot(kind='bar', color='red', width=0.4,
position=0, label='Predicted Labels')

plt.xlabel('Labels')
plt.ylabel('Count')
plt.title('Comparison of Actual and Predicted Labels')
plt.legend()
```

```
plt.show()


import pickle


# Save the trained SVC model
with open('RandomForestClassifierr_model.pkl', 'wb') as file:
    pickle.dump(rf_model, file)


print("RandomForestClassifier model saved successfully.")
# Load the saved SVC model
with open('RandomForestClassifierr_model.pkl', 'rb') as file:
    loaded_rf_model = pickle.load(file)


print("RandomForestClassifier model loaded successfully.")
len(df['career_aspiration'].unique())


# Define the mapping of encoded labels to career aspirations
career_aspirations_mapping =
dict(zip(df['career_aspiration_encoded'],
df['career_aspiration']))
import pandas as pd
from sklearn.preprocessing import LabelEncoder


# Инициализация LabelEncoder для кодирования категориальных данных
label_encoder = LabelEncoder()


# Маппинг значений с английского на украинский
gender_map = {"male": 0, "female": 1}
part_time_job_map = {"yes": 1, "no": 0}  # Прямое преобразование в
числовые значения
extracurricular_map = {"yes": 1, "no": 0}  # Прямое преобразование
в числовые значения
level_map = {"low": 0, "medium": 1, "high": 2}  # Прямое
преобразование в числовые значения
career_map = {"Engineer": "Інженер", "Doctor": "Лікар", "Lawyer":
"Юрист", "Artist": "Артист"}
```

```
# Признаки, использовавшиеся при обучении модели
feature_names = [
    'gender', 'part_time_job', 'absence_days',
'extracurricular_activities',
    'weekly_self_study_hours', 'math_score', 'history_score',
'physics_score',
    'chemistry_score', 'biology_score', 'english_score',
'geography_score',
    'logical_reasoning', 'creativity', 'emotional_intelligence',
    'interest_science_tech', 'interest_art_humanities',
'interest_programming_it',
    'total_score', 'average_score', 'engagement_index',
'tech_index',
    'humanities_index', 'medical_index'
]

# Словарь для ввода данных пользователя
user_input = {}

# Получение пользовательского ввода для каждого признака
for feature in feature_names:
    if feature == 'gender':
        gender = input("Enter gender (male/female): ").lower()
        value = gender_map.get(gender, gender)  # Преобразуем в
украинский

    elif feature == 'part_time_job':
        job = input(f"Does the student have a part-time job?
(yes/no): ").lower()
        value = part_time_job_map.get(job, job)  # Преобразуем в
1/0

    elif feature == 'extracurricular_activities':
        activity = input(f"Does the student participate in
extracurricular activities? (yes/no): ").lower()
```

```
        value = extracurricular_map.get(activity, activity)  # Преобразуем в 1/0


    elif feature in ['math_score', 'history_score', 'physics_score', 'chemistry_score', 'biology_score', 'english_score', 'geography_score']:
        score = float(input(f"Enter the score for {feature} (0-12): "))
        value = score


    elif feature in ['interest_science_tech', 'interest_art_humanities', 'interest_programming_it']:
        value = int(input(f"Enter interest level for {feature} (0 or 1): "))


    elif feature in ['weekly_self_study_hours', 'absence_days']:
        value = float(input(f"Enter the value for {feature}: "))


    elif feature in ['creativity', 'emotional_intelligence', 'logical_reasoning']:
        level = input(f"Enter level for {feature} (low, medium, high): ").lower()
        value = level_map.get(level, level)  # Преобразуем в 0/1/2


    else:
        # Пропускаем ввод для вычисляемых признаков
        continue


    user_input[feature] = value


# Преобразуем ввод пользователя в DataFrame
user_df = pd.DataFrame(user_input, index=[0])


# Рассчёт дополнительных метрик
if 'total_score' not in user_input:
    user_df['total_score'] = user_df[['math_score', 'history_score', 'physics_score',
```

```
                                        'chemistry_score',
'biology_score',
                                        'english_score',
'geography_score']].sum(axis=1)


if 'average_score' not in user_input:
    user_df['average_score'] = user_df['total_score'] / 7


# Рассчитываем дополнительные индексы
if 'engagement_index' not in user_input:
    def calculate_engagement_index(part_time_job,
extracurricular_activities, weekly_self_study_hours):
        return 0.3 * part_time_job + 0.4 *
extracurricular_activities + 0.3 * weekly_self_study_hours
    user_df['engagement_index'] = calculate_engagement_index(
        int(user_df['part_time_job'].iloc[0]),
int(user_df['extracurricular_activities'].iloc[0]),
user_df['weekly_self_study_hours'].iloc[0]
    )


if 'tech_index' not in user_input:
    def calculate_tech_index(math_score, physics_score,
interest_science_tech):
        return 0.5 * math_score + 0.3 * physics_score + 0.2 *
interest_science_tech
    user_df['tech_index'] = calculate_tech_index(
        user_df['math_score'].iloc[0],
user_df['physics_score'].iloc[0],
user_df['interest_science_tech'].iloc[0]
    )


if 'humanities_index' not in user_input:
    def calculate_humanities_index(history_score, english_score,
interest_art_humanities):
        return 0.4 * history_score + 0.4 * english_score + 0.2 *
interest_art_humanities
    user_df['humanities_index'] = calculate_humanities_index(
```

```
        user_df['history_score'].iloc[0],
user_df['english_score'].iloc[0],
user_df['interest_art_humanities'].iloc[0]
    )


if 'medical_index' not in user_input:
    def calculate_medical_index(biology_score, chemistry_score,
weekly_self_study_hours):
        return 0.4 * biology_score + 0.3 * chemistry_score + 0.3 *
weekly_self_study_hours
    user_df['medical_index'] = calculate_medical_index(
        user_df['biology_score'].iloc[0],
user_df['chemistry_score'].iloc[0],
user_df['weekly_self_study_hours'].iloc[0]
    )


# Упорядочиваем DataFrame в соответствии с порядком,
использованным при обучении модели
user_df = user_df[feature_names]


# Предсказание
predictions = rf_model.predict(user_df)


# Преобразование кодов предсказанных значений в текстовое
представление профессии
predicted_career_aspirations =
[career_aspirations_mapping[prediction] for prediction in
predictions]


# Вывод результатов
total_score = user_df['total_score'].values[0]
average_score = user_df['average_score'].values[0]
print("\nTotal Score:", total_score)
print("Average Score:", average_score)
print("\nPrediction:", predictions)
print("Predicted Career Aspirations:",
predicted_career_aspirations)
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split,
cross_val_score
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder, StandardScaler
import seaborn as sns
import matplotlib.pyplot as plt
import pickle


# === 1. Завантаження даних ===
df = pd.read_csv('/content/all_students_data_shuffled.csv')


# Видаляємо нерелевантні колонки
df.drop(columns=['first_name', 'last_name', 'email'],
inplace=True)


# === 2. Підготовка даних ===


# 2.1 Кодування категоріальних змінних
label_encoder = LabelEncoder()
df['gender'] = label_encoder.fit_transform(df['gender'])
df['part_time_job'] =
label_encoder.fit_transform(df['part_time_job'])
df['logical_reasoning'] =
label_encoder.fit_transform(df['logical_reasoning'])
df['creativity'] = label_encoder.fit_transform(df['creativity'])
df['emotional_intelligence'] =
label_encoder.fit_transform(df['emotional_intelligence'])
df['extracurricular_activities'] =
df['extracurricular_activities'].apply(lambda x: 1 if x else 0)


# 2.2 Нормалізація числових змінних
```

```
score_columns = ['math_score', 'history_score', 'physics_score',
'chemistry_score',
                 'biology_score', 'english_score',
'geography_score']
scaler = StandardScaler()
df[score_columns] = scaler.fit_transform(df[score_columns])


# 2.3 Розрахунок додаткових метрик
def calculate_engagement_index(part_time_job,
extracurricular_activities, weekly_self_study_hours):
    return 0.3 * part_time_job + 0.4 * extracurricular_activities
+ 0.3 * weekly_self_study_hours


def calculate_tech_index(math_score, physics_score,
interest_science_tech):
    return 0.5 * math_score + 0.3 * physics_score + 0.2 *
interest_science_tech


def calculate_humanities_index(history_score, english_score,
interest_art_humanities):
    return 0.4 * history_score + 0.4 * english_score + 0.2 *
interest_art_humanities


def calculate_medical_index(biology_score, chemistry_score,
weekly_self_study_hours):
    return 0.4 * biology_score + 0.3 * chemistry_score + 0.3 *
weekly_self_study_hours


df['engagement_index'] = df.apply(lambda row:
calculate_engagement_index(
    row['part_time_job'], row['extracurricular_activities'],
row['weekly_self_study_hours']), axis=1)


df['tech_index'] = df.apply(lambda row: calculate_tech_index(
    row['math_score'], row['physics_score'],
row['interest_science_tech']), axis=1)


df['humanities_index'] = df.apply(lambda row:
calculate_humanities_index(
```

```
    row['history_score'], row['english_score'],
row['interest_art_humanities']), axis=1)


df['medical_index'] = df.apply(lambda row:
calculate_medical_index(
    row['biology_score'], row['chemistry_score'],
row['weekly_self_study_hours']), axis=1)


df['average_score'] = df[score_columns].mean(axis=1)
df['success_probability'] = (
    0.4 * df['engagement_index'] +
    0.6 * df['average_score']
)


# Перетворення текстових даних на числові (якщо ще не зроблено)
df['career_aspiration'] =
label_encoder.fit_transform(df['career_aspiration'])


# Фільтрація тільки числових стовпців
numerical_columns = df.select_dtypes(include=['float64',
'int64']).columns


# Обчислення кореляційної матриці
correlation_matrix = df[numerical_columns].corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f")
plt.title('Матриця кореляцій')
plt.show()


# Видалення надмірно корельованих ознак
X = df[['math_score', 'history_score', 'physics_score',
'chemistry_score',
        'biology_score', 'english_score', 'geography_score',
        'engagement_index', 'tech_index', 'humanities_index',
'medical_index']]
y = df['average_score']
```

```
# === 4. Розділення даних ===
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)


# Масштабування
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)


assert 'average_score' not in X.columns, "Цільова змінна входить
до X!"
models = {
    "LinearRegression": LinearRegression(),
    "Ridge": Ridge(alpha=1.0),
    "Lasso": Lasso(alpha=0.1),
    "RandomForestRegressor":
RandomForestRegressor(n_estimators=100, random_state=42),
    "GradientBoostingRegressor":
GradientBoostingRegressor(n_estimators=100, random_state=42)
}


best_model = None
best_r2 = -np.inf


for name, model in models.items():
    scores = cross_val_score(model, X_train_scaled, y_train, cv=5,
scoring='r2')
    print(f"Модель: {name}")
    print(f"Середній R² на крос-валідації: {scores.mean():.4f}")
    print(f"R² на кожному фолді: {scores}")
    print("=" * 50)


    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
```

```
    r2 = r2_score(y_test, y_pred)
    print(f"R² на тестовому наборі: {r2:.4f}")
    print("-" * 50)

    if r2 > best_r2:
        best_r2 = r2
        best_model = model

print(f"Найкраща модель: {best_model.__class__.__name__} з R² = {best_r2:.4f}")
import matplotlib.pyplot as plt

def regression_comparison_scatter_plot(y_test, models, model_names, X_test_scaled):
    """
    Створює точковий графік порівняння реальних і прогнозованих значень для кількох моделей регресії.

    Parameters:
        y_test: array-like, справжні значення.
        models: list, список моделей регресії.
        model_names: list, список назв моделей.
        X_test_scaled: array-like, масштабовані тестові дані.
    """
    plt.figure(figsize=(12, 8))

    for model, name in zip(models, model_names):
        y_pred = model.predict(X_test_scaled)
        plt.scatter(y_test, y_pred, alpha=0.6, label=name)

    # Лінія ідеального прогнозу
    plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], 'r--', linewidth=2, label='Ideal Prediction')
```

```
    plt.title('Regression Models Comparison: Predicted vs Actual
Values')
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')
    plt.legend()
    plt.grid(True)
    plt.show()


# Приклад використання:
# Список моделей (які ви вже навчили)
regression_models = [LinearRegression(), Ridge(), Lasso(),
RandomForestRegressor(), GradientBoostingRegressor()]
regression_model_names = ["Linear Regression", "Ridge Regression",
"Lasso Regression", "Random Forest", "Gradient Boosting"]


# Навчання моделей (якщо ще не зроблено)
for model in regression_models:
    model.fit(X_train_scaled, y_train)


# Побудова графіка
regression_comparison_scatter_plot(y_test, regression_models,
regression_model_names, X_test_scaled)
# Збереження найкращої моделі
with open('best_regression_model.pkl', 'wb') as file:
    pickle.dump(best_model, file)


# Збереження найкращої моделі
with open('best_avg_model.pkl', 'wb') as file:
    pickle.dump(best_model_avg, file)


# Завантаження моделі
with open('best_avg_model.pkl', 'rb') as file:
    loaded_avg_model = pickle.load(file)


print("\nНайкраща модель завантажена успішно.\n")
```

```
# === 6. Завантаження та тестування ===
with open('best_regression_model.pkl', 'rb') as file:
    loaded_model = pickle.load(file)


def get_user_input():
    user_data = {
        'math_score': float(input("Введіть оцінку з математики (0-
12): ")),
        'history_score': float(input("Введіть оцінку з історії (0-
12): ")),
        'physics_score': float(input("Введіть оцінку з фізики (0-
12): ")),
        'chemistry_score': float(input("Введіть оцінку з хімії (0-
12): ")),
        'biology_score': float(input("Введіть оцінку з біології
(0-12): ")),
        'english_score': float(input("Введіть оцінку з англійської
(0-12): ")),
        'geography_score': float(input("Введіть оцінку з географії
(0-12): ")),
        'engagement_index': float(input("Введіть індекс
залученості (0-1): ")),
        'tech_index': float(input("Введіть технічний індекс (0-1):
")),
        'humanities_index': float(input("Введіть гуманітарний
індекс (0-1): ")),
        'medical_index': float(input("Введіть медичний індекс (0-
1): "))
    }
    return pd.DataFrame(user_data, index=[0])


user_input = get_user_input()
user_input_scaled = scaler.transform(user_input)
predicted_avg_score = loaded_model.predict(user_input_scaled)
print(f"\nПрогнозований середній бал:
{predicted_avg_score[0]:.2f}")
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans, DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import LabelEncoder

# === 1. Підготовка даних ===
# Завантаження даних
df = pd.read_csv('/content/all_students_data_shuffled.csv')
df.head()

# Шаг 1: Кодирование категориальных признаков
label_encoder = LabelEncoder()
df['gender'] = label_encoder.fit_transform(df['gender'])
df['part_time_job'] =
label_encoder.fit_transform(df['part_time_job'])
df['logical_reasoning'] =
label_encoder.fit_transform(df['logical_reasoning'])
df['creativity'] = label_encoder.fit_transform(df['creativity'])
df['emotional_intelligence'] =
label_encoder.fit_transform(df['emotional_intelligence'])
df['extracurricular_activities'] =
df['extracurricular_activities'].apply(lambda x: 1 if x else 0)
df['career_aspiration_encoded'] =
label_encoder.fit_transform(df['career_aspiration'])
# Нормалізація interest_* до діапазону [0, 1]
df['interest_science_tech'] = df['interest_science_tech'] / 10
df['interest_art_humanities'] = df['interest_art_humanities'] / 10
df['interest_programming_it'] = df['interest_programming_it'] / 10

# Розрахунок додаткових метрик
def calculate_engagement_index(part_time_job,
extracurricular_activities, weekly_self_study_hours):
    return 0.3 * part_time_job + 0.4 * extracurricular_activities
+ 0.3 * weekly_self_study_hours
```

```
def calculate_tech_index(math_score, physics_score,
interest_science_tech):
    return 0.5 * math_score + 0.3 * physics_score + 0.2 *
interest_science_tech


def calculate_humanities_index(history_score, english_score,
interest_art_humanities):
    return 0.4 * history_score + 0.4 * english_score + 0.2 *
interest_art_humanities


def calculate_medical_index(biology_score, chemistry_score,
weekly_self_study_hours):
    return 0.4 * biology_score + 0.3 * chemistry_score + 0.3 *
weekly_self_study_hours


# Додавання стовпців, якщо вони не існують
if 'engagement_index' not in df.columns:
    df['engagement_index'] = df.apply(lambda row:
calculate_engagement_index(
        row['part_time_job'], row['extracurricular_activities'],
row['weekly_self_study_hours']), axis=1)


if 'tech_index' not in df.columns:
    df['tech_index'] = df.apply(lambda row: calculate_tech_index(
        row['math_score'], row['physics_score'],
row['interest_science_tech']), axis=1)


if 'humanities_index' not in df.columns:
    df['humanities_index'] = df.apply(lambda row:
calculate_humanities_index(
        row['history_score'], row['english_score'],
row['interest_art_humanities']), axis=1)


if 'medical_index' not in df.columns:
    df['medical_index'] = df.apply(lambda row:
calculate_medical_index(
        row['biology_score'], row['chemistry_score'],
row['weekly_self_study_hours']), axis=1)
```

```
# Фільтрація числових ознак для кластеризації


features_for_clustering = ['engagement_index', 'tech_index',
'humanities_index', 'medical_index',
                           'math_score', 'logical_reasoning',
'weekly_self_study_hours']
scaler = StandardScaler()
clustering_data =
scaler.fit_transform(df[features_for_clustering])



# === 2. Стандартна кластеризація ===
## 2.1 KMeans


inertia = []
range_n_clusters = range(2, 11)
for n_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(clustering_data)
    inertia.append(kmeans.inertia_)


plt.figure(figsize=(8, 6))
plt.plot(range_n_clusters, inertia, marker='o', linestyle='--')
plt.title('Метод ліктя для визначення оптимальної кількості
кластерів')
plt.xlabel('Кількість кластерів')
plt.ylabel('Inertia')
plt.grid()
plt.show()


optimal_clusters = 4


kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
df['kmeans_cluster'] = kmeans.fit_predict(clustering_data)
```

```
plt.figure(figsize=(8, 6))

plt.scatter(clustering_data[:, 0], clustering_data[:, 1],
c=df['kmeans_cluster'], cmap='viridis')

plt.title('Кластери KMeans')

plt.xlabel('Engagement Index')

plt.ylabel('Tech Index')

plt.colorbar(label='Cluster')

plt.show()


silhouette_avg_kmeans = silhouette_score(clustering_data,
df['kmeans_cluster'])

print(f"Середній силуетний бал для KMeans:
{silhouette_avg_kmeans:.4f}")


## 2.2 DBSCAN

dbscan = DBSCAN(eps=1.5, min_samples=5)

df['dbscan_cluster'] = dbscan.fit_predict(clustering_data)


plt.figure(figsize=(8, 6))

plt.scatter(clustering_data[:, 0], clustering_data[:, 1],
c=df['dbscan_cluster'], cmap='plasma')

plt.title('Кластери DBSCAN')

plt.xlabel('Engagement Index')

plt.ylabel('Tech Index')

plt.colorbar(label='Cluster')

plt.show()



# === 3. Власна кластеризація ===


df['custom_score'] = (

    0.5 * df['tech_index'] +

    0.3 * df['humanities_index'] +

    0.2 * df['medical_index']
```

```
)

df['custom_cluster'] = pd.cut(
    df['custom_score'],
    bins=[0, 0.33, 0.66, 1.0],
    labels=['Low Interest', 'Medium Interest', 'High Interest']
)

plt.figure(figsize=(8, 6))
plt.scatter(clustering_data[:, 0], clustering_data[:, 1],
c=df['custom_cluster'].cat.codes, cmap='coolwarm')
plt.title('Кластери на основі Custom Score')
plt.xlabel('Engagement Index')
plt.ylabel('Tech Index')
plt.colorbar(label='Custom Cluster')
plt.show()


# === 4. Збереження та аналіз кластерів ===
print("Розподіл кластерів KMeans:")
print(df['kmeans_cluster'].value_counts())

print("\nРозподіл кластерів DBSCAN:")
print(df['dbscan_cluster'].value_counts())

print("\nРозподіл кластерів за Custom Score:")
print(df['custom_cluster'].value_counts())

# Візуалізація кластерів
def clustering_scatter_plot(clustering_data, cluster_labels,
title="Clustering Visualization"):
    plt.figure(figsize=(10, 6))
    plt.scatter(clustering_data[:, 0], clustering_data[:, 1],
c=cluster_labels, cmap='viridis', alpha=0.7)
    plt.colorbar(label='Cluster')
```

```
    plt.title(title)
    plt.xlabel('Feature 1')
    plt.ylabel('Feature 2')
    plt.grid(True)
    plt.show()
from flask import Flask, request, jsonify
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import pickle

# Инициализация Flask приложения
app = Flask(__name__)

# Загрузка обученной модели
with open('RandomForestClassifierr_model.pkl', 'rb') as file:
    rf_model = pickle.load(file)

# Загрузка маппинга профессий
with open('career_aspirations_mapping.pkl', 'rb') as file:
    career_aspirations_mapping = pickle.load(file)

# Карта для категориальных значений
gender_map = {"male": 0, "female": 1}
part_time_job_map = {"yes": 1, "no": 0}
extracurricular_map = {"yes": 1, "no": 0}
level_map = {"low": 0, "medium": 1, "high": 2}

# Определение используемых признаков
feature_names = [
    'gender', 'part_time_job', 'absence_days',
'extracurricular_activities',
    'weekly_self_study_hours', 'math_score', 'history_score',
'physics_score',
```

```
    'chemistry_score', 'biology_score', 'english_score',
'geography_score',
    'logical_reasoning', 'creativity', 'emotional_intelligence',
    'interest_science_tech', 'interest_art_humanities',
'interest_programming_it',
    'total_score', 'average_score', 'engagement_index',
'tech_index',
    'humanities_index', 'medical_index'
]


# Дополнительные функции расчета индексов
def calculate_engagement_index(part_time_job,
extracurricular_activities, weekly_self_study_hours):
    return 0.3 * part_time_job + 0.4 * extracurricular_activities
+ 0.3 * weekly_self_study_hours


def calculate_tech_index(math_score, physics_score,
interest_science_tech):
    return 0.5 * math_score + 0.3 * physics_score + 0.2 *
interest_science_tech


def calculate_humanities_index(history_score, english_score,
interest_art_humanities):
    return 0.4 * history_score + 0.4 * english_score + 0.2 *
interest_art_humanities


def calculate_medical_index(biology_score, chemistry_score,
weekly_self_study_hours):
    return 0.4 * biology_score + 0.3 * chemistry_score + 0.3 *
weekly_self_study_hours


# API для обработки запроса
@app.route('/explore', methods=['POST'])
def explore():
    try:
        # Получение JSON данных из запроса
```

```
        input_data = request.get_json()


        # Создание DataFrame из JSON
        user_df = pd.DataFrame([input_data])


        # Маппинг категориальных значений
        user_df['gender'] = user_df['gender'].map(gender_map)
        user_df['part_time_job'] =
user_df['part_time_job'].map(part_time_job_map)
        user_df['extracurricular_activities'] =
user_df['extracurricular_activities'].map(extracurricular_map)
        user_df['logical_reasoning'] =
user_df['logical_reasoning'].map(level_map)
        user_df['creativity'] =
user_df['creativity'].map(level_map)
        user_df['emotional_intelligence'] =
user_df['emotional_intelligence'].map(level_map)


        # Рассчет дополнительных метрик
        user_df['total_score'] = user_df[['math_score',
'history_score', 'physics_score',
                                          'chemistry_score',
'biology_score',
                                          'english_score',
'geography_score']].sum(axis=1)
        user_df['average_score'] = user_df['total_score'] / 7
        user_df['engagement_index'] = user_df.apply(lambda row:
calculate_engagement_index(
            row['part_time_job'],
row['extracurricular_activities'],
row['weekly_self_study_hours']), axis=1)
        user_df['tech_index'] = user_df.apply(lambda row:
calculate_tech_index(
            row['math_score'], row['physics_score'],
row['interest_science_tech']), axis=1)
        user_df['humanities_index'] = user_df.apply(lambda row:
calculate_humanities_index(
```

```
            row['history_score'], row['english_score'],
row['interest_art_humanities']), axis=1)
        user_df['medical_index'] = user_df.apply(lambda row:
calculate_medical_index(
            row['biology_score'], row['chemistry_score'],
row['weekly_self_study_hours']), axis=1)

        # Упорядочивание колонок в соответствии с моделью
        user_df = user_df[feature_names]

        # Предсказание
        probabilities = rf_model.predict_proba(user_df)

        # Топ-3 предсказания
        top_n = 3
        top_indices = probabilities[0].argsort()[-top_n:][::-1]
        top_career_predictions = [{"career":
career_aspirations_mapping[idx], "probability":
probabilities[0][idx]}
                                  for idx in top_indices]

        return jsonify({"predictions": top_career_predictions})

    except Exception as e:
        return jsonify({"error": str(e)}), 400

if __name__ == '__main__':
    app.run(debug=True)



app = Flask(__name__)

with open('RandomForestClassifier_model.pkl', 'rb') as model_file:
```

```
    rf_model = pickle.load(model_file)
with open('career_aspirations_mapping.pkl', 'rb') as mapping_file:
    career_mapping = pickle.load(mapping_file)

@app.route('/explore', methods=['POST'])
def explore():
    try:

        input_data = request.get_json()
        user_df = pd.DataFrame([input_data])

        user_df['engagement_index'] =
user_df['weekly_self_study_hours'] * 0.3 + \
                                      user_df['extracurricular_act
ivities'] * 0.4 + \
                                      user_df['part_time_job'] *
0.3
        probabilities = rf_model.predict_proba(user_df)

        top_indices = probabilities[0].argsort()[-3:][::-1]
        result = [{"career": career_mapping[idx], "probability":
probabilities[0][idx]} for idx in top_indices]

        return jsonify({"predictions": result})
    except Exception as e:
        return jsonify({"error": str(e)}), 400

if __name__ == '__main__':
    app.run(debug=True)
```