

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НУ «Запорізька Політехніка»

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК І ТЕХНОЛОГІЙ

(повне найменування інституту, факультету)

КАФЕДРА «СИСТЕМНИЙ АНАЛІЗ ТА ОБЧИСЛЮВАЛЬНА МАТЕМАТИКА»

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

Бакалавра

(рівень вищої освіти (освітньо-кваліфікаційний рівень))

на тему Аналіз закладів вищої освіти країн Азії

Виконав: студент __ 4 _ курсу, групи _КНТ-810

спеціальності (напряму підготовки)

124 Системний аналіз

(код і назва напряму підготовки, спеціальності)

БОБИРЬ Г.Р

(прізвище та ініціали)

Керівник РЯБЕНКО А.Є

(прізвище та ініціали)

Рецензент ЛОЗОВСЬКА Л.І

(прізвище та ініціали)

м.Запоріжжя

2024 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук та технологій

Кафедра _____ «Системний аналіз та обчислювальна
математика» _____

Ступінь вищої освіти _____ Бакалавр _____

Спеціальність 124 Системний аналіз
(код і найменування)

Освітня програма (спеціалізація) : «Інтелектуальні технології та прийняття
рішень в складних системах»

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

Еліна ТЕРЕЩЕНКО

«_____» _____ 2024 року

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

БОБИРЯ Георгія Романовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проекту (роботи): Аналіз закладів вищої освіти країн Азії
керівник проекту (роботи) _____ к.ф -м.н., доц. РЯБЕНКО Антон
Євгенович

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від «12» квітня 2024 року №139

2. Строк подання студентом проекту (роботи) «18» червня 2024 року

3. Вихідні дані до проекту (роботи): Правила збирання та перетворення
визначальних параметрів. Правила використання мов програмування. Стандарти
ДСТУ. Положення про дипломні роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити): вступ, аналітичний огляд освітньої системи країн Азії, математичне
та технічне забезпечення для задачі аналізу закладів вищої освіти,
конструювання, верифікація та аналіз програмної компоненти, висновки

5. Перелік обов'язкового графічного матеріалу:

1) Алгоритм роботи методу СДС;

2) Алгоритм процесу тренування моделі;



- 3) Алгоритм процесу прогнозування;
4) Діаграми процесу авторизації користувача і навчання та додавання моделі в систему.
5) Діаграма послідовності генерації даних та прогнозування.
6) Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1	РЯБЕНКО А.Є., к.ф.-м.н., доц.	19.02.2024	12.06.2024
2	РЯБЕНКО А.Є., к.ф.-м.н., доц.	19.02.2024	12.06.2024
Нормоконтроль	ШИРОКОРАД Д.В., к.ф. - м.н., доц.	13.06.2024	14.06.2024

7) Дата видачі завдання «19» лютого 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Сформулювати мету та основні завдання дипломної роботи	19.02.2024 – 29.02.2024	
2	Опрацювати літературу та існуючі дослідження за темою роботи	01.03.2024 – 08.04.2024	
3	Розробка програмної реалізації для вирішення задачі	09.04.2024 – 08.05.2024	
4	Розрахунки та аналіз даних	09.05.2024 – 22.05.2024	
5	Оформлення пояснювальної записки	23.05.2024 – 03.06.2024	
6	Попередній захист дипломної роботи та отримання рецензій.	04.06.2024 – 17.06.2024	
7	Захист дипломної роботи.	18.06.2024	

Студент(ка)

_____ Георгій БОБИРЬ
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Антон РЯБЕНКО
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

ПЗ: 86 с., 40 рис., 9 табл., 33 джерела.

Об'єкт дослідження: процеси аналізу та оцінки якості освіти в закладах вищої освіти країн Азії.

Мета дослідження – дослідити вплив різних індикаторів на світові рейтинги університетів та розробити математичну модель для прогнозування цих рейтингів, використовуючи алгоритм стохастичного двокоординатного сходження.

Метод дослідження – аналітичний огляд існуючих даних, математичне моделювання, застосування алгоритмів машинного навчання, програмне забезпечення для обробки даних та аналізу результатів.

Проблематика дослідження зумовлена потребою підвищення якості освітніх послуг у відповідь на глобалізаційні виклики та зростаючу конкуренцію серед вищих навчальних закладів. Інноваційність, інтернаціоналізація освіти та інтеграція новітніх технологій стають ключовими факторами успіху університетів на міжнародному рівні.

На основі аналізу були розроблені методики для оцінювання та прогнозування ефективності освітніх програм, що дозволяє університетам адаптуватися до змінних умов та оптимізувати свої стратегії розвитку. Результати дослідження мають практичне значення для формування управлінських рішень у сфері освіти та можуть бути використані для планування поліпшення освітніх послуг.

ОСВІТНІ СИСТЕМИ, АНАЛІЗ ОСВІТИ, ВИЩА ОСВІТА, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, СТОХАСТИЧНЕ ДВОКООРДИНАТНЕ СХОДЖЕННЯ, ML.NET, МАШИННЕ НАВЧАННЯ, ЯКІСТЬ ОСВІТИ.



ЗМІСТ

ЗАДАННЯ.....	2
РЕФЕРАТ.....	4
ВСТУП.....	7
1 АНАЛІТИЧНИЙ ОГЛЯД ОСВІТНЬОЇ СИСТЕМИ КРАЇН АЗІЇ.....	10
1.1 Загальна характеристика освітніх систем Азії.....	10
1.2 Інтернаціоналізація освіти у країнах Азії.....	14
1.3 Виклики та перспективи розвитку освіти в регіоні.....	19
1.4 Аналіз ключових показників якості освіти.....	22
2 МАТЕМАТИЧНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАДАЧІ АНАЛІЗУ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ.....	25
2.1 Формалізація задачі.....	25
2.2 Аналіз та вибір методу машинного навчання.....	26
2.3 Розробка математичної моделі.....	35
2.4 Вибір технологій для реалізації системи.....	37
2.5 Вибір бібліотеки машинного навчання для аналізу закладів вищої освіти країн Азії.....	41
2.6 Опис алгоритмів навчання та прогнозування моделі.....	43
3 КОНСТРУЮВАННЯ, ВЕРИФІКАЦІЯ ТА АНАЛІЗ ПРОГРАМНОЇ КОМПОНЕНТИ.....	48
3.1 Проектування та розробка схеми бази даних.....	48
3.2 Проектування діаграм основних бізнес-процесів.....	51
3.3 Імплементация програмних компонентів.....	55
3.4 Реалізація алгоритму машинного навчання в системі.....	61
3.5 Валідація функціональності компонентів системи.....	74
3.6 Проведення навчання та вивчення результатів компонентного тестування моделі.....	78
3.7 Кореляційний аналіз отриманих даних.....	82
3.8 Оцінка отриманих результатів.....	98

ВИСНОВКИ 100

ПЕРЕЛІК ПОСИЛАНЬ 102

ДОДАТОК А. Програмні коди 106

ДОДАТОК Б. Скрип створення бази даних..... 121

ДОДАТОК В. Перелік графічного матеріалу 125

ВСТУП

В умовах глобалізації та інтенсифікації міжнародної конкуренції, розвиток вищої освіти набуває особливого значення як для окремих країн, так і для регіонів в цілому. Зокрема, заклади вищої освіти країн Азії привертають увагу своїм стрімким розвитком та зростанням якості освітніх послуг, що робить їх конкурентоспроможними на міжнародному рівні. Це, у свою чергу, спонукає до детального аналізу як внутрішніх механізмів забезпечення якості освіти, так і зовнішніх факторів, що впливають на ефективність та привабливість азіатських університетів для міжнародної академічної спільноти.

Актуальність даної теми безсумнівна, адже освітня сфера є ключовим фактором соціально-економічного розвитку та інноваційного прогресу суспільства. Вища освіта формує фундамент для розвитку наукових досліджень, технологічних інновацій та кадрового потенціалу нації. З огляду на швидкі темпи економічного розвитку країн Азії, аналіз їхніх закладів вищої освіти набуває особливої ваги для розуміння сучасних трендів та викликів у глобальному освітньому просторі.

Розвиток та інтеграція інформаційних технологій в освітній процес, зокрема використання алгоритмів машинного навчання для аналізу та прогнозування якості освіти, є одним з ключових аспектів сучасної академічної діяльності. Використання алгоритмів машинного навчання для аналізу історичних даних і прогнозування майбутньої ефективності університетів відкриває нові можливості для оптимізації навчальних програм, підвищення якості викладання та ефективного управління науковими дослідженнями.

Вивчення досвіду азіатських країн у цій сфері має важливе значення не лише для академічних кіл, а й для розробки державних освітніх політик, забезпечення міжнародної інтеграції та співпраці в освітній сфері. Таким чином, дослідження стану та перспектив розвитку вищої освіти в Азії є актуальним та

необхідним для визначення стратегій розвитку освітнього сектору в глобальному контексті.

Обґрунтування актуальності теми базується на критичному аналізі існуючих досліджень, порівняльному аналізі освітніх систем різних країн, а також на аналізі впливу глобалізаційних процесів на розвиток вищої освіти. Враховуючи велику соціальну та економічну значимість цієї проблематики, дослідження також має на меті ідентифікацію ефективних моделей управління та стратегій інноваційного розвитку, що були впроваджені в закладах вищої освіти Азії. Такий підхід дозволяє виявити ключові фактори успіху, які можуть бути адаптовані та впроваджені в інших країнах та регіонах, зокрема в Україні, для підвищення якості та конкурентоспроможності національної вищої освіти.

Мета дослідження – дослідити вплив різних індикаторів на світові рейтинги університетів та розробити математичну модель для прогнозування цих рейтингів, використовуючи алгоритм стохастичного двокоординатного сходження. Система спрямована на підвищення ефективності освітніх програм, оптимізацію управління науковими дослідженнями та забезпечення адаптації освітніх установ до сучасних викликів.

Для досягнення поставленої мети необхідно вирішити наступні основні задачі:

- провести аналітичний огляд освітньої системи країн Азії, що включає загальну характеристику освітніх систем регіону, аналіз процесів інтернаціоналізації освіти, виявлення ключових викликів та перспектив розвитку, а також аналіз ключових показників якості освіти;
- розробити математичне та технічне забезпечення для аналізу закладів вищої освіти, включаючи формалізацію задачі аналізу, вибір та аналіз методу машинного навчання, розробку математичної моделі, а також вибір технологій та бібліотеки машинного навчання для реалізації системи прогнозування;
- сконструювати, верифікувати та проаналізувати програмну компоненту системи, що передбачає проектування та розробку схеми бази даних,

імплементацию програмних компонентів, реалізацію алгоритму машинного навчання в системі, а також валідацію функціональності компонентів системи та вивчення результатів компонентного тестування;

– оцінити отримані результати, здійснити аналіз ефективності розробленої системи прогнозування на основі отриманих даних про майбутню якість освіти в закладах вищої освіти країн Азії, враховуючи здатність системи адаптуватися до змінних умов освітнього середовища.

Наукова новизна отриманих результатів полягає у розробці та успішному застосуванні комплексного підходу до прогнозування майбутньої якості освіти в закладах вищої освіти країн Азії з використанням методів машинного навчання, зокрема, алгоритму стохастичного двокоординатного сходження (Stochastic Dual Coordinate Ascent). Вперше було створено модель, яка інтегрує широкий спектр історичних даних, включаючи якість викладацького складу, наукові публікації, цитування, та інші важливі індикатори, для аналізу та оцінки майбутньої ефективності університетів. Цей підхід дозволив не лише аналізувати поточний стан освітніх закладів, але й прогнозувати їх розвиток у майбутньому, що є значним внеском у сферу управління якістю вищої освіти.

Практичне значення отриманих результатів виражається у створенні системи для прогнозування майбутньої якості освіти, яка може бути використана закладами вищої освіти для вдосконалення своїх освітніх програм та наукових досліджень. Система сприяє підвищенню конкурентоспроможності університетів за рахунок своєчасної адаптації до змінних умов і викликів освітнього середовища, оптимізації ресурсного забезпечення та покращення якості навчання. Застосування розробленої системи дозволяє освітнім закладам не лише реагувати на поточні тенденції, але й активно впливати на формування майбутнього освітнього простору, враховуючи прогнозовані зміни та виклики.

1 АНАЛІТИЧНИЙ ОГЛЯД ОСВІТНЬОЇ СИСТЕМИ КРАЇН АЗІЇ

1.1 Загальна характеристика освітніх систем Азії

Освітні системи Азії є результатом унікального поєднання історичних, культурних та соціально-економічних факторів, які відіграли вирішальну роль у їх формуванні. Історія освіти в таких країнах, як Японія, Китай та Індія, сягає далеко в минуле, де освітні практики тісно переплетені з філософськими традиціями та релігійними вченнями. Наприклад, у Китаї конфуціанство з його акцентом на навчанні та освіченості як шляху до доброчесності мало значний вплив на розвиток освітньої системи. У Японії велике значення приділялось формуванню моральних якостей особистості, а в Індії традиційна освіта базувалась на ведичних текстах, що наголошували на духовному розвитку.

Ці глибоко вкорінені традиції сформували ставлення до освіти, яке вважається не лише засобом отримання професійних знань, але й інструментом гармонійного розвитку особистості [1]. Повага до вчителя, яка вважається ключовим елементом успішного навчального процесу, є однією з найбільш характерних рис освітньої культури цих країн. Учитель не просто передає знання, але й виступає як моральний авторитет, приклад для наслідування.

Також, історичний розвиток освіти в азійських країнах відбувався в умовах соціально-економічних змін, що зумовили потребу в адаптації освітніх систем до сучасних викликів. Це призвело до формування освітніх структур, здатних поєднувати традиційні цінності з необхідністю вирішення актуальних проблем сучасного світу, таких як глобалізація, технологічний розвиток, забезпечення якості освіти та її доступності.

Особливу роль в освіті цих країн відіграє інтеграція з міжнародною освітньою спільнотою, що сприяє обміну знаннями та досвідом, але водночас вимагає збереження національної ідентичності та культурних традицій. В цьому контексті інтеграція з міжнародною освітньою спільнотою відкриває нові можливості для азійських університетів і шкіл не тільки у сфері академічного

обміну, але й у співпраці в наукових дослідженнях та інноваційних проектах. Це сприяє вдосконаленню освітніх програм, адаптації курсів до світових стандартів, впровадженню передових технологій та методик навчання. Однак, паралельно з відкритістю до світових освітніх практик, азійські країни прагнуть зберегти свою унікальність, інтегруючи національні та культурні аспекти в освітній процес. Таке поєднання глобалізації та локалізації дозволяє формувати у студентів не тільки глибокі професійні знання, а й розуміння власної культурної спадщини, що є важливим для розвитку міцної національної ідентичності.

Така інтеграція також спонукає азійські країни до реформування освітньої політики, з метою не лише поліпшення якості освіти, але й забезпечення її відповідності міжнародним стандартам [2]. Це, у свою чергу, призводить до зростання конкурентоспроможності азійських освітніх установ на світовій арені, приваблення іноземних студентів та науковців, а також до розширення можливостей для своїх громадян за кордоном.

Азійські країни активно розвивають міжнародне співробітництво в освітній сфері, вступаючи в численні міжурядові та недержавні освітні ініціативи, укладаючи двосторонні угоди про обмін студентами та науковцями, сприяючи реалізації спільних науково-дослідних проектів. Таке зростання міжнародної інтеграції сприяє не лише підвищенню якості освіти та досліджень, але й забезпечує обмін культурними цінностями та розширює горизонти як для викладачів, так і для студентів.

Освітні інституції Азії також приділяють велику увагу розвитку цифрових технологій у навчанні, впроваджуючи інноваційні освітні платформи, дистанційне навчання та використання штучного інтелекту для персоналізації освітнього процесу. Це дозволяє не тільки зробити освіту більш доступною для широкого кола людей, але й забезпечує високий рівень індивідуалізації навчання, відповідаючи особистим освітнім потребам кожного студента.

Структури освітніх систем країн Азії демонструють цікаве поєднання універсальних освітніх принципів і унікальних національних підходів, які відображають культурне, історичне та соціально-економічне розмаїття регіону.

Характерною особливістю є система поділу на рівні освіти, що зазвичай включає початкову, середню та вищу освіту, доповнену можливостями для післядипломної та довічної освіти.



Рисунок 1.1 – Структура освітніх систем країн Азії

Початкова і середня освіта в більшості країн Азії є обов'язковою, що забезпечує високий рівень грамотності населення. В країнах, таких як Південна Корея та Японія, особлива увага приділяється якості початкової та середньої освіти, яка розглядається як фундамент для подальшого навчання та професійного розвитку [3]. Високий рівень конкуренції за вступ до престижних університетів спонукає студентів до інтенсивного навчання з ранніх років.

У сфері вищої освіти спостерігається значна різноманітність форм навчання, від традиційних університетів, що пропонують широкий спектр гуманітарних, соціальних та природничих наук, до спеціалізованих коледжів та технічних інститутів, що зосереджуються на підготовці фахівців у певних галузях. Наприклад, Індія відома своїми інженерними школами, такими як Індійські технологічні інститути (ІТІ), які користуються світовим визнанням.

Післядипломна освіта в Азії також демонструє значний ріст, відображаючи глобальні тенденції в посиленні наукових досліджень та розвитку академічних кадрів. Багато університетів активно розвивають магістерські та докторські

програми, співпрацюючи з міжнародними науковими установами та промисловістю для підвищення якості та актуальності наукових досліджень.

Окремої уваги заслуговують ініціативи та реформи в освіті, що реалізуються в різних країнах Азії. Ці заходи часто спрямовані на підвищення доступності та якості освіти, адаптацію освітніх систем до потреб сучасного ринку праці та викликів глобалізації. Наприклад, у Китаї велика увага приділяється розвитку професійної освіти та навчання з метою підготовки кваліфікованих робітників, необхідних для підтримки швидкого економічного росту країни. У той же час, Сінгапур зосереджений на створенні інноваційної освітньої системи, що стимулює критичне мислення та креативність, враховуючи потреби освітньої економіки.

Багато країн Азії також активно інвестують у технології в освіті, реалізуючи програми електронного навчання та використовуючи інтернет-ресурси для розширення доступу до якісної освіти, особливо в віддалених та сільських регіонах. Це не тільки сприяє зменшенню розриву у рівні освіти між містами та сільською місцевістю, але й відкриває нові можливості для довічного навчання та саморозвитку.

Іншим важливим аспектом реформ є зусилля щодо міжнародної інтеграції освітніх систем. Університети Азії посилюють співпрацю з закладами освіти інших країн, розширюють програми обміну для студентів та викладачів, а також залучають міжнародних експертів для підвищення якості наукових досліджень та освітніх програм [4]. Це не тільки підвищує престиж азійських університетів на міжнародному рівні, але й сприяє культурному обміну та розумінню між народами.

Таким чином, освітні системи країн Азії продовжують розвиватися, відповідаючи на виклики сучасності та водночас зберігаючи унікальні національні традиції. Їх досвід показує, як можливо поєднати глобальні тенденції та локальні цінності у створенні ефективних та інноваційних освітніх моделей, спрямованих на формування гармонійно розвиненої особистості, здатної до

життєвого самовдосконалення та ефективної взаємодії в мультикультурному світі.

1.2 Інтернаціоналізація освіти у країнах Азії

Азійські країни протягом останнього десятиліття завойовують провідні позиції на глобальному ринку освітніх послуг. Для підтвердження зростаючого рівня вищої освіти в азійських країнах наводяться дані рейтингів вищих навчальних закладів, у яких університети Китаю та Японії займають все вищі позиції. Відзначається помітне підвищення якісного рівня вищої освіти, пропонованої у Сінгапурі, Південній Кореї, Тайвані, Гонконзі та інших країнах. Зміни у системах вищої освіти азійських країн відбуваються в напрямку глобалізації та інтернаціоналізації. Тим часом фахівці говорять про необхідність «азійського шляху» розвитку вищої освіти.

Університети країн Азії в основних аспектах своєї діяльності змушені адаптувати англосаксонську модель з метою досягнення конкурентоздатної позиції на світовому ринку освітніх послуг. Хо Ка Мок зауважує, що інтернаціоналізацію не слід розглядати лише як відповідь на американські або англосаксонські стандарти. При цьому він визнає більш високий рівень академічних спільнот у США та країнах Європи. За думкою Хо Ка Мока, слід критично підходити до питання про те, що необхідно запозичувати з Європи та США і в якій мірі інтегрувати ці запозичення у освітні системи азійських країн [5].

Азійські університети все більше уваги приділяють міжнародному бенчмаркінгу, що означає вивчення та запозичення позитивного досвіду зарубіжних, переважно європейських країн у сфері освіти. До цього їх підштовхує бажання зайняти провідні позиції у світових рейтингах університетів, підвищити свій престиж у очах світової громадськості,

забезпечити притік іноземних абітурієнтів тощо. При цьому враховується той факт, що критерії для потрапляння в рейтинги визначені англосаксонськими традиціями та практикою.

Якісний розвиток вищої освіти в країнах Азії багато хто пов'язує з його інтернаціоналізацією. Перші спроби інтернаціоналізації вищої освіти в країнах Азії були відзначені в другій половині XIX століття. Вони пов'язані зі стажуванням та більш глибокою підготовкою азіатських студентів і викладачів у зарубіжних країнах. Університети Азії починають адаптувати західну модель вищої освіти, яка продемонструвала свою ефективність в Німеччині, Франції, Великій Британії, а також у США. Флагманами інтернаціоналізації освіти в Азії цього періоду можна вважати Китай і Японію. Частково це пов'язано з тим, що Японія та Китай не були колоніями європейських країн, як, наприклад, Індонезія, Малайзія, Філіппіни. У країнах-колоніях, як неодноразово відзначав Ф. Альтбах, майже не приділялося уваги викладанню мови, а також загалом освіті та науці, що в подальшому вплинуло на рівень освіти в цих країнах [6].

З середини XX століття і до 1980-х років характер інтернаціоналізації вищої освіти визначався Холодною війною, а країни були умовно поділені на дві групи. Одна група адаптувала радянську модель, а інша - американську модель вищої освіти. Кардинальні зміни почали відбуватися в 1990-х роках, що було пов'язано з інтенсивною економічною глобалізацією, розвитком ІТ-технологій, а також впровадженням у громадські сфери ринково орієнтованих механізмів. Прагнення до інтернаціоналізації освіти в азійських країнах пояснюється економічними факторами, а також зростанням конкуренції в глобальному середовищі. Спектр заходів, підпорядкованих інтернаціоналізації, став розширюватися: від мобільності студентів і викладачів до інтернаціоналізації навчальних планів.

Інтенсивність заходів щодо інтернаціоналізації вищої освіти в країнах Азії залежить від освітньої політики окремих країн. Засновуючись на досвіді азійських країн, виділяють три напрямки транснаціональної освіти:

- регульоване державою (Китай, Малайзія, Південна Корея);

- ринково орієнтоване (Гонконг);
- транзитивне, що характеризується переходом від регульованого державою до ринково орієнтованого (Японія, Тайвань) [7].

У Тайвані серед провідних форм міжнародної освіти виокремлюють навчання за кордоном, програми подвійних дипломів та онлайн-навчання. У Сінгапурі розвиток отримав «внутрішню» дистанційну освіту, збільшилася кількість філіалів закордонних університетів.

Серед азійських країн найбільш показовими є зміни в системі вищої освіти Китаю. Дослідники вищої освіти в країнах Азії розглядають Китай як загальний образ тих змін, які відбуваються в освіті всіх азійських країн.

Вища освіта в Китаї протягом багатьох років була ізольована від світового ринку освітніх послуг, а сама модель вищої освіти в середині ХХ століття була під впливом моделі вищої освіти, що існувала в СРСР. До 1988 року освіта була безкоштовною, але існуюча модель не відповідала вимогам економічного розвитку країни. З 1992 року китайські університети отримали можливість приймати на навчання чверть студентів за плату.

У 2000-х роках вища освіта в Китаї стала більш відкритою і звертали увагу на всесвітні тенденції розвитку вищої освіти. Вступ КНР до СОТ призвів до значних змін у китайському суспільстві, а також у системі вищої освіти, що стало предметом значного дослідницького інтересу. Велика відкритість країни, зміни в економіці та освіті зробили освітній ринок Китаю привабливим для закордонних освітніх закладів.

У зв'язку з вступом до СОТ Китай змушений орієнтуватися на міжнародні стандарти в галузі освіти. Вищі навчальні заклади в Китаї почали дотримуватися вимог інтернаціоналізації вищої освіти, конкурувати за приваблення іноземних студентів. Якщо у 1990 році у Китаї за різними програмами навчалося приблизно 7 тис. іноземних громадян, то у 2000 році їх уже було 85829, чверть з яких навчалася за програмами, що передбачають отримання ступеня. Також стала помітна інша тенденція. Якщо у 1990-х роках навчання іноземних студентів фінансувалося за рахунок стипендій та грантів, то останні роки 90 % іноземних

студентів навчаються в Китаї за власний рахунок. Понад половина іноземних громадян, що перебувають в Китаї, вивчають китайську мову. Другим за популярністю напрямком серед іноземних студентів є медицина. 60% іноземців вивчають традиційну медицину, а 40% - китайську.

За думкою фахівців, багаті культурні та історичні традиції Китаю дозволять поступово збільшувати кількість приваблених на навчання іноземних студентів. Значним стимулом для приваблення іноземних студентів є щорічні стипендії на безкоштовне навчання в університетах Китаю. Визначальним для китайської вищої освіти став 2003 рік, коли було прийнято рішення дозволити закордонним освітнім установам створювати на території Китаю спільні освітні заклади з китайською стороною. Вже до 2004 року налічувалося 700 освітніх програм, пропонованих китайськими навчальними закладами спільно з закордонними партнерами.

Інша особливість розвитку вищої освіти в Китаї пов'язана з зростанням кількості китайських громадян, які виїжджають на навчання в закордонні університети. Значно зросли доходи значної частини китайських сімей дозволили батькам оплачувати навчання дітей як у китайських вузах, так і за кордоном. У 2007 році на навчанні в закордонних університетах перебувало 144 тис. студентів. За прогнозами Міністерства освіти Китаю, починаючи з 2010 року, за кордоном щорічно будуть навчатися 200 тис. осіб, а з 2020 року - вже 300 тис. Станом на 2006 рік, університети Японії становили 66% від загальної кількості іноземних студентів, Австралії - 22%, Канади - 20%, США і Великої Британії - приблизно 16%. У Великій Британії представництво китайських студентів є найбільшим серед усіх іноземних студентів [8].

Паралельно з традиційними формами навчання, китайські студенти отримують освіту за дистанційними програмами, що пропонуються закордонними університетами, у віртуальних університетах, філіалах та інших подібних установах. Найбільших успіхів на китайському ринку освітніх послуг досягли Університет штату Міссурі (США) та Стенфордський центр професійного розвитку (США).

Значну роль у розвитку університетської освіти в Китаї відіграла державна підтримка за допомогою так званих проектів 211 і 985. Згідно з проектом 211 у 1993 році було обрано 100 провідних університетів Китаю, які отримали пріоритетне фінансування. Проект 985, запропонований у 1998 році, передбачав більше суттєвого фінансування 43 найкращих університетів Китаю з метою підняття їх до «світового рівня».

Додатковим стимулом до розвитку системи вищої освіти в Китаї стало дозвіл на створення навчальних закладів спільно з закордонними партнерами та формування приватного сектору вищої освіти. Результатом цієї ініціативи стала діяльність на території Китаю університетів Ліверпуля та Ноттінгема (Великобританія), а також заснування за ініціативою уряду Данії Китайсько-Датського університету та інших.

У останні роки в Китаї активно розвиваються приватні університети. Р. Хейно та Дж. Лін приводять приклад першого в Китаї приватного університету «Yellow River University of S&T», заснованого у 1994 році [9].

Спостережені зміни у системі вищої освіти в КНР призвели до змін у ціннісних засадах академічної культури Китаю. Тому, більш глибокого дослідження заслуговує поточний стан академічної культури Китаю та зміни, що відбуваються в ній, порівняльний аналіз академічної культури азійських та західних університетів.

Ф. Альтбах зауважує, що академічна культура в країнах Азії ґрунтується на принципі академічних досягнень, на свободі науково-дослідної ініціативи та на принципі конкуренції, які доповнюються елементами співпраці та мобільності. Багато аспектів академічного життя в Китаї все ще залежать від особистих зв'язків, наприклад, кар'єрного зростання, розподілу фінансових ресурсів тощо. Другою характерною рисою китайського академічного середовища є те, що професорсько-викладацький склад, як правило, поповнюється своїми випускниками. В результаті молоді викладачі при прийомі на роботу в університет не проходять відповідної оцінки на відкритому ринку праці.

Для адаптації китайської академічної культури до вимог сучасності потрібні реформи, спрямовані на стимулювання інновацій та розвиток відкритого середовища для науково-дослідницької діяльності. Це може включати зміцнення механізмів відбору та оцінки наукових кадрів, розвиток прозорих критеріїв кар'єрного росту, підтримку міждисциплінарного співробітництва та створення стимулів для молодих науковців.

Крім того, для сприяння інтернаціоналізації китайського вищого навчання необхідно активізувати обмін студентами та викладачами з іноземних країн, розвивати міжнародні програми та ініціативи співпраці з провідними університетами світу. Це допоможе залучити нові ідеї та підходи, сприяти культурному розмаїттю та розвитку міжкультурного розуміння, а також підвищити міжнародний престиж китайської освіти.

1.3 Виклики та перспективи розвитку освіти в регіоні

Виклики та перспективи розвитку освіти в азійському регіоні відображають широкий спектр соціальних, економічних і технологічних тенденцій, що впливають на глобальний освітній ландшафт. Освітні системи країн Азії стикаються з рядом викликів, які вимагають адаптивних рішень та інноваційного підходу для забезпечення якості освіти та її доступності. Водночас, в цих викликах криються значні перспективи для розвитку та трансформації освітніх практик.

Освітні системи Азії знаходяться на перехресті глобальних та локальних викликів, що формують їхній розвиток у XXI столітті (рис. 1.2). З одного боку, ці системи повинні відповідати на швидкі зміни в технологіях, демографії та соціально-економічних умовах, що безпосередньо впливають на потреби та очікування суспільства від освіти [10]. З іншого боку, вони мають унікальну можливість використати ці виклики як каталізатори для інновацій та

покращення, впроваджуючи нові підходи та методи навчання, які б зробили освіту більш доступною, якісною та відповідною до потреб майбутнього.



Рисунок 1.2 – Основні проблеми та виклики розвитку освіти в Азії

Ключовими аспектами, що визначають сучасну динаміку розвитку освіти в азійському регіоні, є:

- технологічні зміни. Швидкий розвиток технологій вимагає від освітніх систем швидкої адаптації та інтеграції нових засобів навчання. Проблема полягає в забезпеченні викладачів та студентів необхідними навичками для ефективної роботи в цифровому середовищі;

- демографічні зміни. У деяких країнах Азії, таких як Японія та Китай, спостерігається старіння населення, що ставить під загрозу здатність освітніх систем підтримувати необхідний рівень робочої сили. Водночас, в інших регіонах, наприклад у Південній Азії, високий рівень молодіжного населення створює потребу в розширенні доступу до якісної освіти;

- соціально-економічні розриви. Нерівність у доступі до освіти між міськими та сільськими регіонами, а також між різними соціальними групами, залишається значною проблемою. Це вимагає розробки і реалізації цілеспрямованих програм для забезпечення рівного доступу до освітніх ресурсів;

– якість освіти [11]. Забезпечення високої якості освіти, включаючи навчальні програми, кваліфікацію викладачів, та інфраструктуру, є постійним викликом. Особливо це стосується потреби в модернізації освітніх програм для відповідності сучасним професійним та життєвим вимогам.

Освітня сфера в азійських країнах перебуває на порозі значних змін, що викликані глобалізацією, технологічним прогресом, та зростаючими соціально-економічними вимогами. Водночас із викликами, ці зміни відкривають широкі можливості для реформування та подальшого розвитку освітніх систем. Азійські країни активно шукають шляхи до інновацій та адаптації своїх освітніх моделей з метою підготовки кваліфікованих спеціалістів, здатних відповідати вимогам сучасного світу. Це вимагає комплексного підходу, який охоплює впровадження новітніх технологій у навчання, акцент на STEM освіту, забезпечення високої якості навчальних програм, та розширення міжнародної співпраці (рис. 1.3).



Рисунок 1.3 – Діаграма розподілу перспектив розвитку освіти в Азії

Перспективи розвитку освіти в Азії охоплюють наступні ключові аспекти:

- інтеграція ІКТ у навчання, що передбачає широке використання цифрових технологій та інтернет-ресурсів для підвищення доступності та ефективності освітнього процесу;
- орієнтація на STEM та міждисциплінарні програми, спрямовані на підготовку фахівців, які зможуть вирішувати складні інноваційні задачі;

- підвищення якості освіти, що включає оновлення навчальних програм, покращення кваліфікації викладачів, та модернізацію освітньої інфраструктури;
- розширення міжнародної співпраці у сфері освіти, що дозволяє інтегрувати кращі світові практики, забезпечує обмін досвідом та знаннями між країнами [12-13].

Ці перспективи відображають стратегічні напрямки розвитку освіти в азійському регіоні, спрямовані на формування сучасної, інноваційної та конкурентоспроможної освітньої системи.

1.4 Аналіз ключових показників якості освіти

В епоху глобалізації та стрімких змін, роль освіти як драйвера інновацій та соціального прогресу є незаперечною. Значення глибокого розуміння ключових аспектів якості освіти стає критичним не лише для окремих націй, а й для цілих регіонів, особливо таких як Азія, де освітні системи проходять через швидкі трансформації, викликані економічними та технологічними інноваціями. Вивчення ключових індикаторів якості освіти дозволяє оцінити не тільки ефективність освітніх програм, але й їхню здатність адаптуватися до потреб сучасного розвитку, визначаючи їхній вплив на соціально-економічне зростання країн та регіону загалом.

Враховуючи акцент на значущості аналітичного підходу до вивчення освітніх систем, наступним етапом є візуальне представлення отриманих даних [14]. Діаграма розподілу, яка представлена на рис. 1.4, надає змогу оглядово оцінити ключові показники якості освіти в Азії. Це забезпечує не тільки можливість оцінки кожного індикатора в ізоляції, але й сприяє розумінню їх взаємного впливу та вкладу в комплексний розвиток освітніх систем регіону.



Рисунок 1.4 – Показники якості освіти

До ключових показників якості освіти відносяться:

- зайнятість випусників. Рівень зайнятості випусників свідчить про здатність освітньої системи підготувати спеціалістів, затребуваних на ринку праці. Високий показник зайнятості випусників відображає ефективність освіти у формуванні практичних навичок та компетенцій;
- якість викладацького складу. Кваліфікація та досвід викладачів мають безпосередній вплив на якість освіти. У країнах Азії значна увага приділяється підвищенню кваліфікації викладачів та залученню висококласних науковців для роботи у вищих навчальних закладах;
- публікації. Кількість та якість наукових публікацій відображає науковий потенціал країни та її внесок у світову науку. Висока активність у цій сфері свідчить про розвинену науково-дослідницьку базу та інтеграцію країни у глобальний науковий простір;
- вплив. Вплив освітньої системи можна оцінити через її внесок у розвиток суспільства, культури, економіки, а також через міжнародне визнання. Азійські освітні системи прагнуть розширювати свій вплив, зокрема через міжнародні партнерства та обмінні програми;

– цитування. Кількість цитувань наукових робіт свідчить про їх значимість та вплив на наукову спільноту. Високий показник цитування свідчить про якість наукових досліджень та активну участь університетів в міжнародному науковому діалозі. У країнах Азії, де існує сильна науково-дослідницька база, показники цитування часто вищі, що свідчить про внесок цих країн у світову науку;

– широкий вплив. Оцінка широкого впливу освітньої системи включає аналіз її внеску у соціальний, економічний та культурний розвиток як на національному, так і на міжнародному рівні. Азійські країни, активно інвестуючи в освіту, прагнуть не тільки підвищити свій внутрішній потенціал, але й відігравати ключову роль у глобальних процесах;

– патенти. Кількість патентів, отриманих університетами та науково-дослідницькими інститутами, є індикатором інноваційної активності та практичного застосування наукових досліджень. У азійському регіоні багато країн демонструють високий рівень інновацій, що веде до зростання кількості патентів. Це не тільки сприяє економічному розвитку, але й підкреслює роль освітніх інституцій у просуванні наукових і технологічних інновацій [15].

Враховуючи викладені ключові показники якості освіти, такі як якість освітніх програм, рівень зайнятості випускників, кваліфікацію викладачів, наукові публікації, вплив, цитованість, широкий вплив та інноваційні патенти, стає зрозуміло, що ці фактори відіграють критичну роль у формуванні ефективної освітньої системи. Методи машинного навчання, у свою чергу, відкривають нові можливості для аналізу цих показників та прогнозування якості освіти. Застосування алгоритмів машинного навчання дозволяє обробляти великі обсяги даних про освітні заклади, їхні програми, викладачів та студентів, виявляючи закономірності та тенденції, які не завжди очевидні при традиційних методах аналізу. Це сприяє розробці обґрунтованих рекомендацій для підвищення ефективності освітнього процесу та адаптації освітніх програм до змінних умов сучасного світу.

2 МАТЕМАТИЧНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАДАЧІ АНАЛІЗУ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ

2.1 Формалізація задачі

Задача полягає у розробці та імplementації системи для аналізу та прогнозування майбутньої ефективності університетів країн Азії, заснованої на комплексному оцінюванні за такими критеріями:

- якість освіти. Вимірювання цього параметру передбачає аналіз успішності студентів, рівня акредитації програм, результатів незалежних оцінювань та опитувань серед студентів. Цей критерій також включає оцінювання якості навчального процесу та інфраструктури для забезпечення ефективного навчального середовища;

- рівень зайнятості випускників. Аналізується відсоток випускників, які змогли знайти роботу за спеціальністю протягом певного періоду після закінчення навчання. Цей індикатор свідчить про реальну затребуваність випускників на ринку праці та ефективність освітньої програми;

- якість викладацького складу. Враховуються кваліфікація та досвід викладачів, їхні наукові досягнення, участь у міжнародних наукових проектах, а також відгуки студентів про якість викладання. Цей критерій є ключовим для забезпечення високого рівня освітніх послуг;

- кількість наукових публікацій. Оцінюється активність університету в науковій сфері, кількість публікацій в рецензованих наукових журналах, участь у наукових конференціях та симпозіумах. Цей показник вказує на науковий потенціал закладу та його внесок у розвиток науки;

- рівень впливу та цитування наукових робіт. Аналізується, наскільки широко цитуються наукові роботи викладачів та дослідників університету в міжнародних наукових джерелах. Цей індикатор дозволяє оцінити внесок університету в світову наукову спільноту;

- широкий вплив університету. Включає оцінку впливу університету на регіональний та міжнародний розвиток, його участь у міжнародних освітніх та наукових програмах, партнерства з іншими університетами, індустрією та громадськістю;

- кількість патентів. Вимірюється кількість патентів та корисних моделей, отриманих викладачами та дослідниками університету, що свідчить про інноваційний потенціал та практичне застосування досліджень.

Використання цих критеріїв дозволяє створити комплексну модель аналізу та прогнозування, засновану на об'єктивних та вимірюваних індикаторах. Така модель сприятиме оптимізації процесів управління якістю освіти та сприятиме подальшому розвитку вищих навчальних закладів в країнах Азії.

Результатом аналізу є прогнозування загального балу ефективності кожного університету, що використовується для визначення його місця у світових та регіональних рейтингах. Це дозволить адміністрації університетів та органам управління освітою краще розуміти поточний стан та перспективи розвитку вищих навчальних закладів, а також приймати обґрунтовані рішення щодо покращення якості освіти.

2.2 Аналіз та вибір методу машинного навчання

Розробка системи аналізу та прогнозування ефективності університетів країн Азії базується на виборі алгоритму, здатного ефективно вирішувати задачі класифікації та регресії, при цьому забезпечуючи високу точність прогнозування та оптимізацію обчислювальних ресурсів. Вибір методу машинного навчання є фундаментальним кроком, який визначає не лише стратегію обробки даних, але й загальну ефективність майбутньої системи.

Важливо враховувати, що ідеальний алгоритм повинен бути не тільки високопродуктивним у короткостроковій перспективі, але й масштабним та

адаптивним до змінних умов обробки великих наборів даних. Отже, процес вибору методу машинного навчання охоплює глибокий аналіз доступних алгоритмів з урахуванням специфіки задачі, їх потенційних переваг та обмежень.

У дослідженні особлива увага приділяється таким методам машинного навчання, як стохастичне двокоординатне сходження (СДС), градієнтний спуск (ГС), та метод опорних векторів (ОВ). Кожен з цих методів має свої особливості, які можуть бути використані для ефективної реалізації задачі аналізу та прогнозування в контексті університетів. Подальші кроки аналізу зосереджені на детальному розгляді кожного з цих методів з метою визначення найбільш оптимального для вирішення поставленої задачі.

Метод стохастичного двокоординатного сходження є одним з передових алгоритмів в області машинного навчання, призначеним для ефективної оптимізації в задачах, де необхідно працювати з великими наборами даних і високою розмірністю простору ознак. В основі методу лежить ідея стохастичного вибору підмножини координат (або ознак) для оновлення на кожному кроці оптимізаційного процесу, що дозволяє значно знизити обчислювальну складність і покращити швидкість збіжності алгоритму.

СДС знайшов широке застосування в багатьох сферах, де присутні великі обсяги даних і потреба в їх швидкому аналізі. Це включає, але не обмежується, такими галузями як фінанси, електронна комерція, біоінформатика, соціальні мережі, і, звісно, освіта. В освітній сфері метод може бути використаний для аналізу та прогнозування ефективності університетів, оцінки якості освітніх програм, прогнозування успішності студентів, та інших завдань, пов'язаних з обробкою та аналізом великих даних [16].

У контексті аналізу закладів вищої освіти країн Азії, СДС може бути застосований для оптимізації великих датасетів, які включають різноманітні параметри: якість викладацького складу, наукові публікації, цитування, рівень зайнятості випускників, інноваційність досліджень тощо. Завдяки стохастичному підходу до вибору координат для оновлення, СДС дозволяє

здійснювати глибокий аналіз таких даних, ефективно виявляючи ключові фактори, які впливають на ефективність університетів.

Одним із потенційних сценаріїв застосування СДС в цій області є прогнозування рейтингів університетів на основі історичних даних. Метод може використовуватись для оптимізації регресійних моделей, які оцінюють вплив різних індикаторів на загальну оцінку ефективності закладу. Такий підхід дозволяє не лише аналізувати поточний стан освітніх установ, але й робити обґрунтовані прогнози щодо їх розвитку в майбутньому [17].

На рис. 2.1 представлено алгоритм роботи методу СДС.

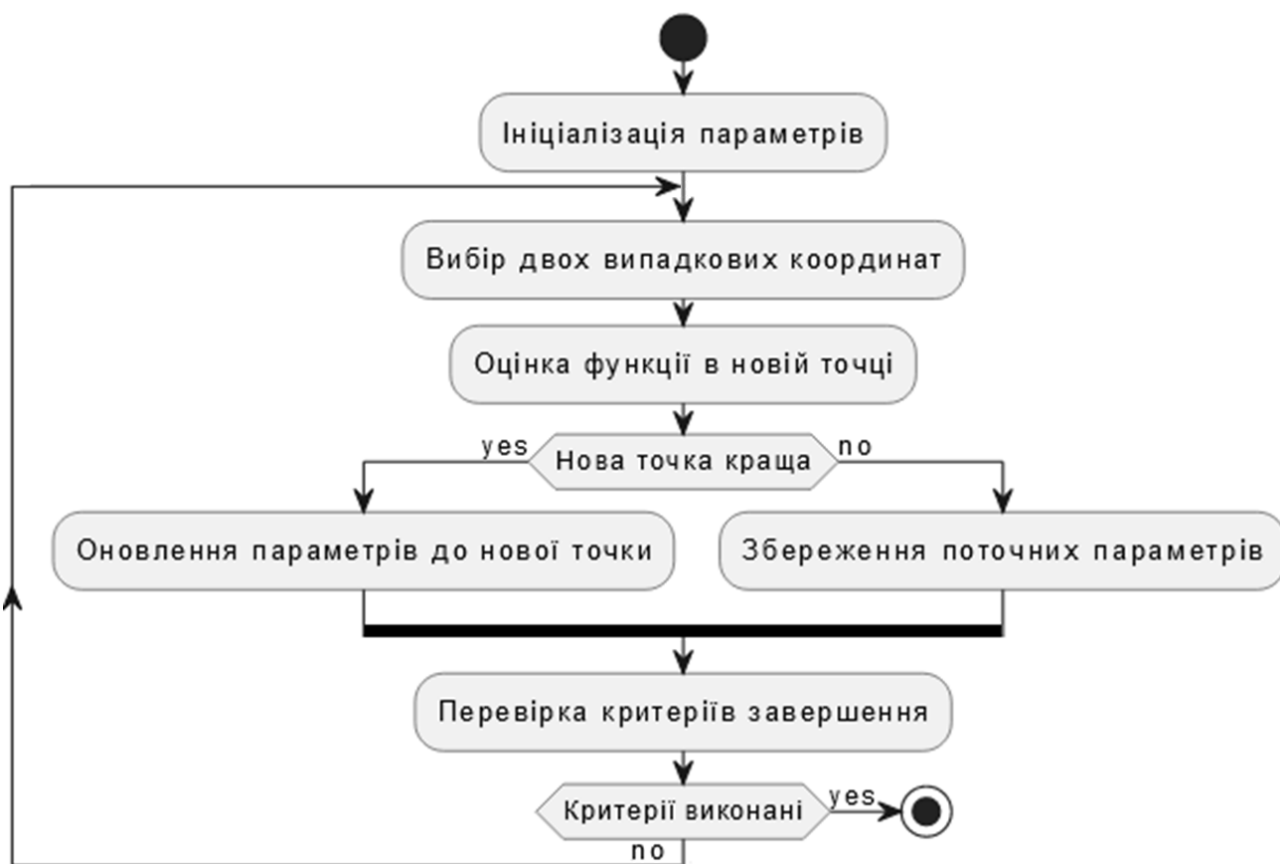


Рисунок 2.1 – Алгоритм роботи методу СДС

Метод СДС є ітераційним процесом оптимізації, який розпочинається з ініціалізації початкових параметрів, таких як початкові координати точки, значення змінних, та інші необхідні параметри. Далі, з усього набору можливих координат випадковим чином вибираються дві для подальшого пошуку. В новій точці, що відповідає обраним координатам, обчислюється значення цільової функції для оцінки ефективності вибору. Залежно від результату, параметри

методу оновлюються до нових значень або залишаються без змін. Після цього проводиться перевірка на виконання критеріїв завершення процесу, як-от досягнення заданої точності або максимальної кількості ітерацій. Якщо умови для завершення не виконані, алгоритм повторює ітерацію з оновленими параметрами. Процес триває до тих пір, поки не буде знайдено рішення, що відповідає заданим критеріям, після чого робота алгоритму завершується.

Метод градієнтного спуску є фундаментальним алгоритмом оптимізації, який широко застосовується в машинному навчанні для мінімізації функції втрат. Він полягає в ітеративному оновленні параметрів моделі в напрямку, протилежному градієнту функції втрат по цим параметрам, що дозволяє поступово знаходити мінімум цієї функції [18].

Метод градієнтного спуску знайшов своє застосування у великій кількості задач оптимізації, від простих лінійних регресій до складних нейронних мереж. Він використовується в таких областях як фінанси, маркетинг, розпізнавання образів, автоматичний переклад, та інших, де потрібно ефективно мінімізувати помилку прогнозування.

Для аналізу та прогнозування ефективності закладів вищої освіти, метод градієнтного спуску може бути застосований для оптимізації моделей, які оцінюють різноманітні аспекти діяльності університетів. Це можуть бути моделі, що прогнозують якість освіти на основі таких параметрів, як наукові публікації, кваліфікація викладацького складу, інфраструктура, залучення студентів до наукової діяльності, і так далі.

Застосування ГС дозволяє налаштувати ваги цих параметрів таким чином, щоб мінімізувати розбіжності між прогнозованими та реальними показниками ефективності закладів. Це може включати аналіз історичних даних для виявлення тенденцій, які впливають на рейтинги університетів, та ідентифікацію ключових факторів, що сприяють підвищенню якості освіти [19].

На рис. 2.2 представлено діаграму діяльності для алгоритму роботи методу градієнтного спуску.

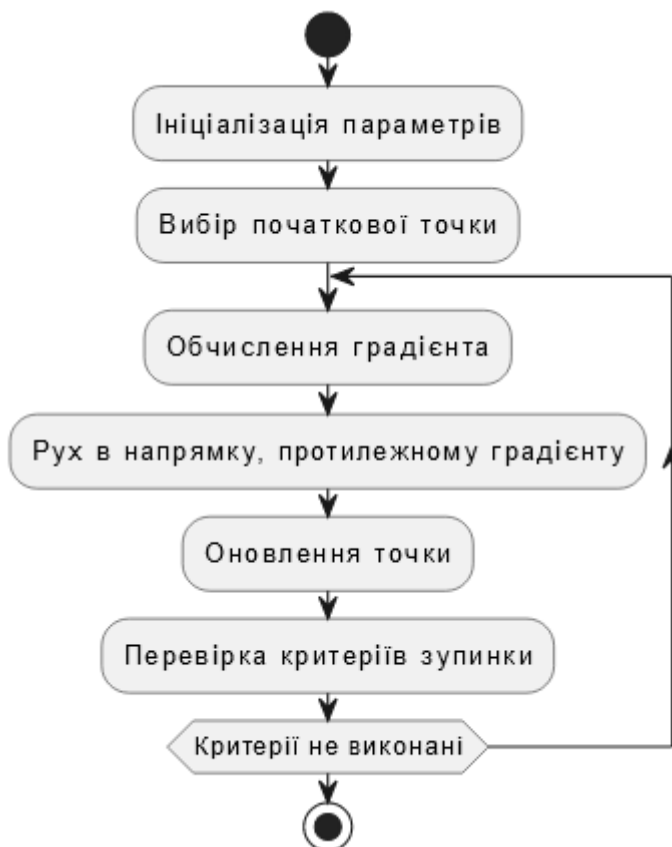


Рисунок 2.2 – Алгоритм роботи методу ГС

Метод градієнтного спуску починається з встановлення необхідних параметрів, включаючи швидкість навчання та визначення стартової точки у просторі параметрів, щоб розпочати пошук мінімуму цільової функції. На кожному етапі процесу обчислюється градієнт цільової функції в поточній точці, який вказує напрямок найшвидшого зростання функції. Щоб знайти мінімум, потрібно рухатися в протилежному напрямку до градієнту. Оновлення поточної точки здійснюється шляхом корекції у напрямку, протилежному градієнту, з метою мінімізації значення цільової функції. Процес продовжується до того моменту, коли будуть виконані критерії зупинки, такі як мінімальне змінення значення функції чи параметрів між ітераціями або досягнення максимальної кількості ітерацій. Завершивши, метод вважає поточну точку за наближене рішення мінімуму цільової функції. Цей підхід робить метод градієнтного спуску цінним інструментом для розв'язання задач оптимізації, особливо у сферах машинного навчання та глибокого навчання, де ключовим є знаходження локальних мінімумів функції [20].

Метод опорних векторів є потужним інструментом машинного навчання, заснованим на принципах статистичного навчання, для розв'язання задач класифікації та регресії. ОВ шукає гіперплощину або набір гіперплощин у багатовимірному просторі, які можуть бути використані для класифікації або регресії. Однією з ключових особливостей ОВ є здатність максимізувати маржу між класами даних, що дозволяє досягати високої точності класифікації.

ОВ широко застосовується у багатьох областях, включаючи розпізнавання образів, біоінформатика, обробку природної мови, та інші, де потрібні потужні інструменти для аналізу складних наборів даних. Метод опорних векторів ефективно працює навіть тоді, коли кількість ознак значно перевищує кількість спостережень, що робить його ідеально підходящим для задач, де даних відносно мало, але вони багатовимірні [21].

У контексті аналізу закладів вищої освіти країн Азії, ОВ може бути використаний для різноманітних завдань, від класифікації університетів за рівнем якості освіти до прогнозування успішності студентів або ефективності наукових досліджень. Завдяки здатності ефективно обробляти великі обсяги багатовимірні даних, ОВ ідеально підходить для аналізу комплексних наборів даних, які зазвичай асоціюються з освітніми установами.

Наприклад, ОВ може бути використаний для розробки моделей, здатних розрізняти університети за рівнем дослідницької активності, базуючись на кількості публікацій, цитувань, наукових грантів, тощо. Також, метод можна застосувати для виявлення закладів, які показують виняткові результати в конкретних галузях науки або освіти, виходячи зі структурованих даних, таких як рейтинги, оцінки студентів та викладачів, і фінансові показники.

Окрім того, ОВ дозволяє ідентифікувати ключові фактори, що сприяють високій якості освітніх програм, аналізуючи різноманітні ознаки та їх вплив на загальний успіх університету. Це робить метод незамінним інструментом для розробки стратегій розвитку та покращення освітніх закладів.

Рис. 2.2 відображає діаграму діяльності для алгоритму роботи методу опорних векторів.



Рисунок 2.3 – Алгоритм роботи методу ОВ

Алгоритм роботи методу ОВ починається з вибору типу ядра, який найкраще підходить для перетворення вхідного простору даних, щоб забезпечити кращу сепарабельність класів. Вибір ядра (лінійне, поліноміальне, радіальне базове функції тощо) залежить від особливостей даних, які потрібно класифікувати. Далі ініціалізуються параметри моделі ОВ, включно з параметром регуляризації та параметрами ядра, що є критичними для налаштування моделі [22].

Процес продовжується побудовою оптимізаційної моделі, яка мінімізує помилку класифікації та максимізує відстань між опорними векторами, що є найближчими точками різних класів. Ця відстань максимізується для знаходження гіперплощини, що найкраще розділяє класи у перетвореному просторі. Оптимізація параметрів моделі відбувається з метою мінімізації помилок на тренувальному наборі даних.

Після налаштування моделі вона використовується для класифікації нових даних, що дозволяє ефективно розділити їх на класи. Метод ОВ вважається ефективним інструментом у машинному навчанні для бінарної класифікації,

особливо завдяки його спроможності працювати з великими просторами ознак та забезпечувати високу точність на різноманітних типах даних [23].

У процесі вибору методу машинного навчання для вирішення поставленої задачі було побудовано табл. 2.1, яка відображає аналіз та порівняння методів СДС, ГС та ОВ. Цей аналіз базується на ключових характеристиках, які важливі для успішної реалізації та ефективності методу в контексті задачі аналізу закладів вищої освіти країн Азії.

Таблиця 2.1 – Порівняльний аналіз методів машинного навчання

Характеристика	СДС	ГС	ОВ
Специфіка вибору напрямку пошуку	Випадковий вибір двох координат	Гradientний (у напрямку найшвидшого зниження функції)	Залежить від вибору ядра та оптимізації гіперплощини
Здатність до паралелізації	Висока (можливість паралельного оновлення різних координат)	Середня (паралелізм можливий у векторних обчисленнях)	Низька (складно паралелізувати процес навчання)
Чутливість до локальних мінімумів	Низька (завдяки випадковості вибору координат)	Висока (може «застрягти» у локальних мінімумах)	Залежить від типу ядра та налаштувань
Вимоги до обчислювальних ресурсів	Низькі (не потрібно обчислювати градієнт для всього набору даних)	Високі (потрібно обчислювати градієнт на кожному кроці)	Високі (особливо при великій кількості ознак і даних)
Складність реалізації	Середня (потребує реалізації механізму випадкового вибору)	Низька (метод добре документований і легко реалізований)	Висока (потребує розуміння ядерних методів)
Потреба в налаштуванні гіперпараметрів	Низька (основний гіперпараметр - крок сходження)	Висока (швидкість навчання, кількість ітерацій)	Висока (параметри регуляризації, тип ядра)
Універсальність застосування	Середня (ефективний для великої кількості задач)	Висока (застосовний до багатьох задач оптимізації)	Середня (підходить для задач класифікації)

Вибір методу СДС для аналізу закладів вищої освіти країн Азії базується на ряді його унікальних переваг, які виявляються особливо цінними для вирішення цієї конкретної задачі. Перш за все, ефективність СДС у великих датасетах з високою розмірністю простору ознак робить його ідеальним для обробки складних і об'ємних даних освітнього сектора. Враховуючи значний обсяг інформації, яка має бути проаналізована, здатність СДС ефективно оптимізувати без потреби в обчисленні градієнта для всього набору даних на кожному кроці значно знижує обчислювальні витрати.

Окрім того, випадковий вибір координат у СДС зменшує його чутливість до локальних мінімумів, що забезпечує більш глобальний пошук оптимального рішення. Ця властивість є особливо важливою в контексті освітніх даних, де ландшафт може містити численні локальні оптимуми через різноманіття факторів, які впливають на ефективність навчальних закладів. Висока здатність СДС до паралелізації також сприяє його вибору, оскільки це дозволяє значно прискорити процес аналізу, роблячи його більш ефективним і масштабованим у порівнянні з іншими методами, які мають нижчу здатність до паралелізації.

Також, низькі вимоги СДС до обчислювальних ресурсів і потреби в налаштуванні гіперпараметрів роблять його привабливим з практичної точки зору, зважаючи на обмежені ресурси та необхідність в ефективному управлінні часом і технічними можливостями при великомасштабних дослідженнях. Таким чином, враховуючи специфіку задачі аналізу ефективності закладів вищої освіти, СДС виокремлюється як найбільш оптимальний метод, що забезпечує високу ефективність обробки даних при відносно низьких витратах ресурсів і часу.

2.3 Розробка математичної моделі

Для стохастичного двокоординатного сходження математична модель може бути описана на основі задачі оптимізації, де ми намагаємося мінімізувати цільову функцію, представлену як суму втрат L по всім навчальним прикладам, додавши до цього регуляризаційний термін для контролю за складністю моделі. Ця оптимізація здійснюється шляхом ітеративного вибору і оновлення двох координат (ознак) вектору параметрів w моделі, що значно знижує обчислювальне навантаження порівняно з іншими методами.

Математично модель може бути виражена як:

$$f(x) = w^T x + b, \quad (2.1)$$

де: $f(x)$ – прогнозоване значення моделі;

x – вектор вхідних ознак;

w – вектор ваг ознак;

b – зміщення.

Цільова функція $J(w)$, яка підлягає мінімізації, може включати суму втрат L і регуляризаційний термін $\lambda R(w)$, де L вимірює розбіжності між фактичними та прогнозованими значеннями, а $\lambda R(w)$ контролює за складністю моделі, запобігаючи перенавчанню:

$$J(w) = L(w) + \lambda R(w) \quad (2.2)$$

Специфіка СДС полягає у виборі напрямку оновлення. На кожному кроці алгоритм вибирає дві координати ваг w_i і w_j , і оновлює їх, рухаючись у напрямку, який зменшує цільову функцію $J(w)$. Вибір координат здійснюється випадково,

що допомагає уникнути застрягання в локальних мінімумах і забезпечує більшу різноманітність пошукових напрямків.

Втрати L можуть бути різними залежно від задачі, наприклад, квадратична втрата для регресії або логістична втрата для класифікації. Регуляризаційний термін часто вибирається як L_1 (ласо регуляризація), що сприяє розрідженості моделі, або L_2 (регуляризація Ріджа), що запобігає надмірному зростанню ваг.

Оптимізація моделі за допомогою СДС включає вибір гіперпараметрів, таких як швидкість навчання та кількість ітерацій, які впливають на процес навчання. Навчання моделі продовжується до тих пір, поки не буде досягнуто заданого критерію зупинки, наприклад, коли зміна $J(w)$ стає нижчою за певний поріг або коли кількість ітерацій перевищує заданий ліміт.

Для аналізу закладів вищої освіти спочатку необхідно визначити вхідні та вихідні параметри на основі заданих даних, після чого формалізувати процес навчання і оцінки моделі, включаючи функцію втрат і методику оцінки якості прогнозування.

Математично, набір вхідних даних X моделі, отриманих з коду, може бути представлений як матриця, де кожен рядок відповідає одному університету, а кожен стовпець – одній з характеристик:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}, \quad (2.3)$$

де m – кількість університетів, а n – кількість ознак, що розглядаються для кожного університету (наприклад, якість освіти, зайнятість випускників тощо).

Вихідний параметр y , який є вектором прогнозованих балів ефективності університетів, можна представити як:

$$y = [y_1, y_2, \dots, y_m]^T, \quad (2.4)$$

де кожен y_i відповідає прогнозованому загальному балу ефективності i -го університету.

Процес навчання моделі в контексті СДС полягає у мінімізації функції втрат $L(y, \hat{y})$, яка оцінює різницю між істинними значеннями ефективності y і прогнозованими значеннями \hat{y} :

$$L(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2, \quad (2.5)$$

Метод СДС оптимізує цільову функцію через ітеративний вибір і оновлення пари координат ваг моделі, тим самим мінімізуючи $L(y, \hat{y})$.

Для оцінки якості прогнозування моделі використовуються такі метрики, як коефіцієнт детермінації R^2 і середня абсолютна помилка (MAE), які визначаються на основі різниці між істинними та прогнозованими значеннями.

2.4 Вибір технологій для реалізації системи

У процесі розробки системи для аналізу та прогнозування ефективності закладів вищої освіти країн Азії, ключовим аспектом є вибір відповідних технологій, які б забезпечили не тільки високу продуктивність та точність роботи системи, але й гнучкість, масштабованість та зручність подальшого супроводу. Вибір технологічного стеку має ґрунтуватися на декількох фундаментальних критеріях, серед яких: потреби проекту, здатність технологій задовольнити ці потреби, вартість впровадження та підтримки.

Першим та одним із найважливіших рішень є вибір мови програмування. Для розробки системи велике значення мають такі аспекти, як наявність бібліотек та інструментів для машинного навчання, обробки та аналізу даних.

C# є потужною, типізованою мовою програмування, розробленою Microsoft як частина платформи .NET. Її синтаксис сприяє чистоті та зрозумілості коду, роблячи її ідеальною для розробки як десктопних, так і веб-

додатків. С# широко використовується для створення корпоративних додатків, ігор з використанням Unity, а також мобільних додатків через Xamarin [24].

Java відома своєю платформонезалежністю, що дозволяє розробникам писати код один раз та виконувати його скрізь, де є Java Virtual Machine (JVM). Ця мова є домінантною на ринку корпоративних рішень, завдяки своїй стабільності, масштабованості та широкому набору бібліотек та фреймворків, таких як Spring та Hibernate. Java також активно використовується для розробки Android-додатків, що робить її особливо цінною для мобільної розробки [25].

Python славиться своєю лаконічністю та гнучкістю, що робить мову доступною для вивчення новачками та ефективною для прототипування. Вона широко використовується в наукових дослідженнях, обробці даних, штучному інтелекті та веб-розробці, завдяки потужним бібліотекам як NumPy, Pandas, TensorFlow та Django [26].

Для наглядної оцінки та зручності аналізу характеристик різних мов програмування, дані було систематизовано у табл. 2.2.

Таблиця 2.2 – Порівняльний аналіз мов програмування

Характеристика	С#	Java	Python
Інтеграція з .NET Framework	Висока (розроблена як частина .NET)	Середня (через сторонні інструменти)	Низька
Крос-платформеність	Висока (завдяки .NET Core та Xamarin)	Висока (Java Virtual Machine)	Висока
Розробка десктопних додатків	Висока (WPF, UWP для Windows)	Середня (Swing, JavaFX)	Низька (можлива через сторонні фреймворки)
Розробка мобільних додатків	Висока (Xamarin)	Середня (Android Studio, але переважно для Android)	Низька (можлива через Kivy, але менш популярна)
Багатопоточність	Висока (зручні абстракції через TPL)	Висока (управління потоками через JVM)	Середня (GIL може бути обмеженням)

Вибір C# для розробки системи аналізу закладів вищої освіти країн Азії обґрунтовується кількома ключовими факторами, які роблять цю мову програмування ідеально підходящою для задачі:

- інтегрованість з .NET Framework і .NET Core. C# є частиною екосистеми .NET, що надає великий набір бібліотек і інструментів для розробки. Це забезпечує потужну підтримку для розробки додатків та обробки даних;
- масштабованість і продуктивність. C# і .NET оптимізовані для високої продуктивності та масштабованості, що критично важливо для систем аналізу даних, здатних обробляти великі обсяги інформації;
- безпека. .NET Framework і .NET Core містять вбудовані засоби безпеки, що дозволяє ефективно захистити дані і забезпечити конфіденційність інформації про студентів та викладачів. Використання C# допомагає впроваджувати сучасні практики безпеки та аутентифікації, що є важливим для систем, що працюють з особистими даними.
- підтримка спільноти та ресурси для навчання. Велика і активна спільнота розробників C# та .NET забезпечує багато ресурсів для навчання, підтримки та спілкування.

Для розробки системи з використанням методів машинного навчання, вибір відповідної системи управління базами даних (СУБД) набуває особливої важливості. СУБД служить ключовим елементом для зберігання, обробки та аналізу великих обсягів клієнтських даних, які є основою для тренування та валідації моделей машинного навчання.

MS SQL Server є високопродуктивною реляційною системою управління базами даних (СУБД), розробленою Microsoft. Ця платформа забезпечує надійне зберігання даних, розширені можливості аналізу та комплексну підтримку транзакцій, що робить її популярним вибором для корпоративних застосунків. MS SQL Server включає інтеграцію з іншими продуктами Microsoft, що забезпечує легку взаємодію з додатками на основі .NET, а також з інструментами для бізнес-аналізу та інтелектуального аналізу даних [27].

MySQL є відкритою реляційною СУБД, яка широко використовується для розробки веб-додатків, завдяки своїй простоті, надійності та ефективності. Ця система підтримує широкий спектр мов програмування, включаючи PHP, Python та Java, що робить її ідеальною для інтеграції в різноманітні веб-проекти. MySQL пропонує гнучкі можливості реплікації та масштабування, що дозволяє легко адаптуватися до зростаючих вимог проекту [28].

Oracle Database відома своєю масштабованістю, потужними можливостями для управління великими обсягами даних та високим рівнем безпеки, що робить її вибором номер один для багатьох великих корпорацій та фінансових інститутів. Ця СУБД підтримує складні транзакції, високу доступність та відновлення після збоїв, забезпечуючи надійне зберігання та обробку критично важливих даних.

У табл. 2.3 представлено аналіз основних характеристик розглянутих СУБД. Цей аналіз має на меті допомогти у виборі найбільш підходящої СУБД для ефективного зберігання та обробки даних.

Таблиця 2.3 – Порівняльний аналіз СУБД

Характеристика	MS SQL Server	MySQL	Oracle
Ліцензування	Комерційне (з безкоштовною версією Express)	Відкрита ліцензія (з комерційними опціями)	Комерційне (з обмеженою безкоштовною версією XE)
Інтеграція з іншими продуктами	Висока інтеграція з продуктами Microsoft	Обмежена спеціалізована інтеграція	Висока інтеграція з продуктами Oracle
Підтримка транзакцій	Повна підтримка ACID	Повна підтримка ACID	Повна підтримка ACID
Аналітичні можливості	Розширені (з використанням SQL Server Analysis Services)	Обмежені	Розширені (за допомогою Oracle Analytics)
Крос-платформенність	Підтримка Windows, Linux	Підтримка багатьох платформ	Підтримка багатьох платформ
Масштабованість	Висока з	Висока	Висока

Вибір MS SQL Server для розробки системи аналізу закладів вищої освіти країн Азії може бути виправданий його високою продуктивністю, надійністю та інтегрованістю з іншими продуктами та сервісами Microsoft. Ця СУБД забезпечує потужні аналітичні та бізнес-інтелект інструменти, які дозволяють глибоко аналізувати великі обсяги даних, що є критично важливим для оцінки ефективності освітніх програм і закладів. MS SQL Server підтримує високу масштабованість і доступність, забезпечуючи безперервну роботу системи навіть при збільшенні обсягу даних, а також пропонує розширені можливості для забезпечення безпеки даних, що є особливо актуальним при обробці особистої інформації студентів і викладачів.

2.5 Вибір бібліотеки машинного навчання для аналізу закладів вищої освіти країн Азії

Для розробки системи на платформі C# потрібно розглянути переваги та обмеження ключових бібліотек, таких як ML.NET, Accord.NET та Infer.NET.

ML.NET є потужною, відкритою бібліотекою машинного навчання для .NET, розробленою Microsoft, яка дозволяє розробникам легко інтегрувати складні моделі машинного навчання в .NET-додатки. Ця бібліотека підтримує широкий спектр задач машинного навчання, включаючи класифікацію, регресію, кластеризацію та прогнозування часових рядів, роблячи її універсальним інструментом для різних сценаріїв використання. ML.NET активно розвивається та підтримується Microsoft, що забезпечує їй постійне оновлення та вдосконалення [30].

Accord.NET призначена для наукових обчислень в .NET, яка включає бібліотеки для статистичного аналізу даних, обробки зображень, машинного навчання, комп'ютерного зору та аудіо обробки. Розроблена як відкрита бібліотека, Accord.NET пропонує розробникам багатий набір алгоритмів, що охоплюють різні області штучного інтелекту та машинного навчання. Вона

особливо корисна для тих, хто шукає всеохоплюючий інструментарій для реалізації комплексних систем машинного навчання та аналізу даних у .NET-середовищі [31].

Infer.NET є фреймворком Microsoft для моделювання байєсових графічних моделей та інших статистичних моделей через проблематичне програмування. Бібліотека спрямована на забезпечення дослідників та розробників інструментами для створення складних моделей машинного навчання, які можуть автоматично аналізувати та робити висновки з даних [32].

У табл. 2.4 представлено порівняльний аналіз бібліотек машинного навчання.

Таблиця 2.4 – Порівняння бібліотек машинного навчання

Характеристика	ML.NET	Accord.NET	Infer.NET
Підтримка різноманітних типів моделей МН	Висока (класифікація, регресія, кластеризація тощо)	Висока (великий вибір алгоритмів)	Середня (фокус на байєсових мережах)
Інтеграція з .NET	Розроблена спеціально для .NET	Хороша, але як окрема бібліотека	Із фокусом на байєсові моделі
Масштабованість і продуктивність	Оптимізована для великих датасетів і розподілених систем	Добра для однопоточних додатків	Націлена на складні статистичні моделі
Простота використання	Висока з AutoML і модельними шаблонами	Середня, вимагає більше знань для налаштування	Вища для статистичних моделей
Інструменти для попередньої обробки даних	Висока (обширні можливості для попередньої обробки)	Висока (багато інструментів)	Обмежена (фокус моделюванні)

Вибір ML.NET для розробки системи аналізу закладів вищої освіти країн Азії обумовлений його потужними можливостями та тісною інтеграцією з екосистемою .NET, що значно спрощує розробку та впровадження комплексних рішень машинного навчання. Ця бібліотека надає розробникам доступ до широкого спектру передових алгоритмів машинного навчання без потреби у глибоких знаннях у цій галузі, завдяки функціям AutoML та зручним шаблонам моделей. Крім того, ML.NET оптимізований для роботи з великими обсягами даних, що є критично важливим для аналізу інформації про велику кількість навчальних закладів та їх ефективності. Таким чином, ML.NET стає ідеальним інструментом для створення ефективної, надійної та масштабованої системи аналізу освітнього сектору.

2.6 Опис алгоритмів навчання та прогнозування моделі

У контексті розробки системи для аналізу ефективності закладів вищої освіти країн Азії, застосування методу СДС стає ключовим підходом для оптимізації та аналізу великих наборів даних. Використання СДС дозволяє точно і швидко визначати важливі ознаки та впливові фактори, що сприяють підвищенню якості освітнього процесу, враховуючи широкий спектр історичних даних, від академічних показників до соціальних аспектів взаємодії між студентами та викладачами.

На рис. 2.4 представлена блок-схема алгоритму для навчання моделі, яка ілюструє ключові етапи процесу.

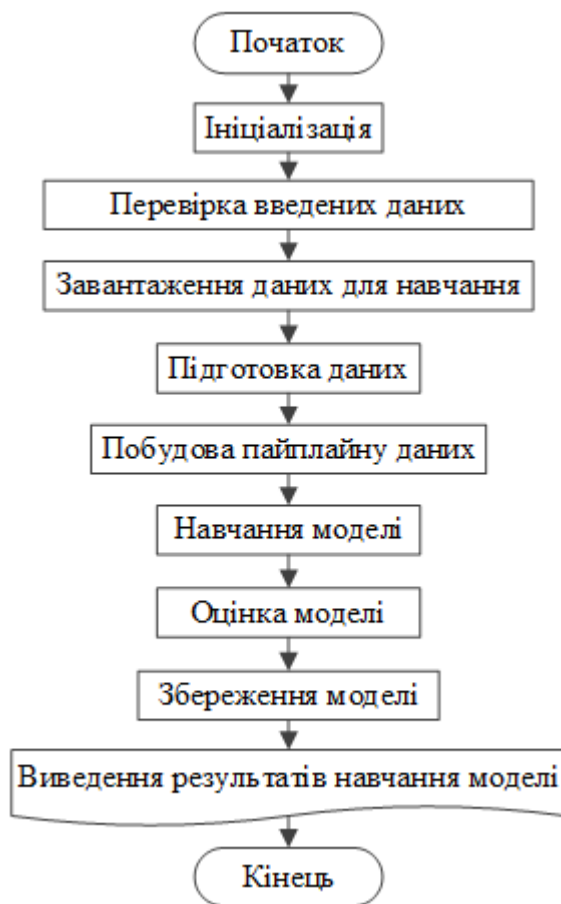


Рисунок 2.4 – Алгоритм процесу тренування моделі

Запуск процесу тренування моделі машинного навчання відзначається ретельною підготовкою, що включає конфігурацію параметрів і встановлення оптимального середовища для обробки даних. Початкова перевірка вхідних даних вимагає аналізу на повноту та достовірність, що дозволяє виявити й виправити можливі недоліки, такі як відсутні значення або аномалії, що могли б спотворити результати навчання. Детальний огляд даних забезпечує міцний фундамент для наступних етапів процесу.

Завантаження даних для навчання та їх підготовка є вирішальними етапами, які визначають якість фінальної моделі. На цьому етапі дані проходять через процеси очищення від помилок, нормалізації для забезпечення однорідності масштабів та трансформації для забезпечення правильного вигляду даних, необхідного для машинного навчання. Вибір ознак, які будуть використовуватися для навчання моделі, базується на аналізі їх впливу на

цільову змінну, де менш важливі ознаки відкидаються, а найбільш значущі – включаються до фінального набору даних.

Коли дані належно підготовлені, розпочинається процес навчання моделі, що є ітеративним та вимагає від алгоритму здатності адаптуватися до специфіки даних. В рамках навчання моделі машинного навчання, комп'ютер аналізує навчальний набір даних, використовуючи його для виявлення взаємозв'язків та закономірностей, які потім можуть бути використані для прогнозування або класифікації нових даних. Після кожного кроку навчання модель перевіряється, щоб оцінити її прогностичні здібності та внести коригування у ваги ознак.

Оцінка якості тренованої моделі включає в себе використання заздалегідь відокремленого тестового набору даних, який не брав участі в навчанні. Це дає можливість об'єктивно оцінити, наскільки добре модель здатна узагальнити отримані знання. Різноманітні метрики, такі як середня квадратична помилка для задач регресії або матриця помилок для задач класифікації, дають цілісне уявлення про продуктивність моделі.

Завершення процесу тренування відбувається з збереженням моделі, що забезпечує можливість її повторного використання без необхідності перенавчання. Весь процес тренування моделі машинного навчання є циклічним та може бути повторений з метою покращення моделі, забезпечуючи її оптимізацію та актуалізацію згідно з новими даними та вимогами.

Для прогнозування загального балу, який використовується для визначення світового рейтингу навчального закладу розроблено алгоритм блок-схема якого зображена на рис. 2.5.

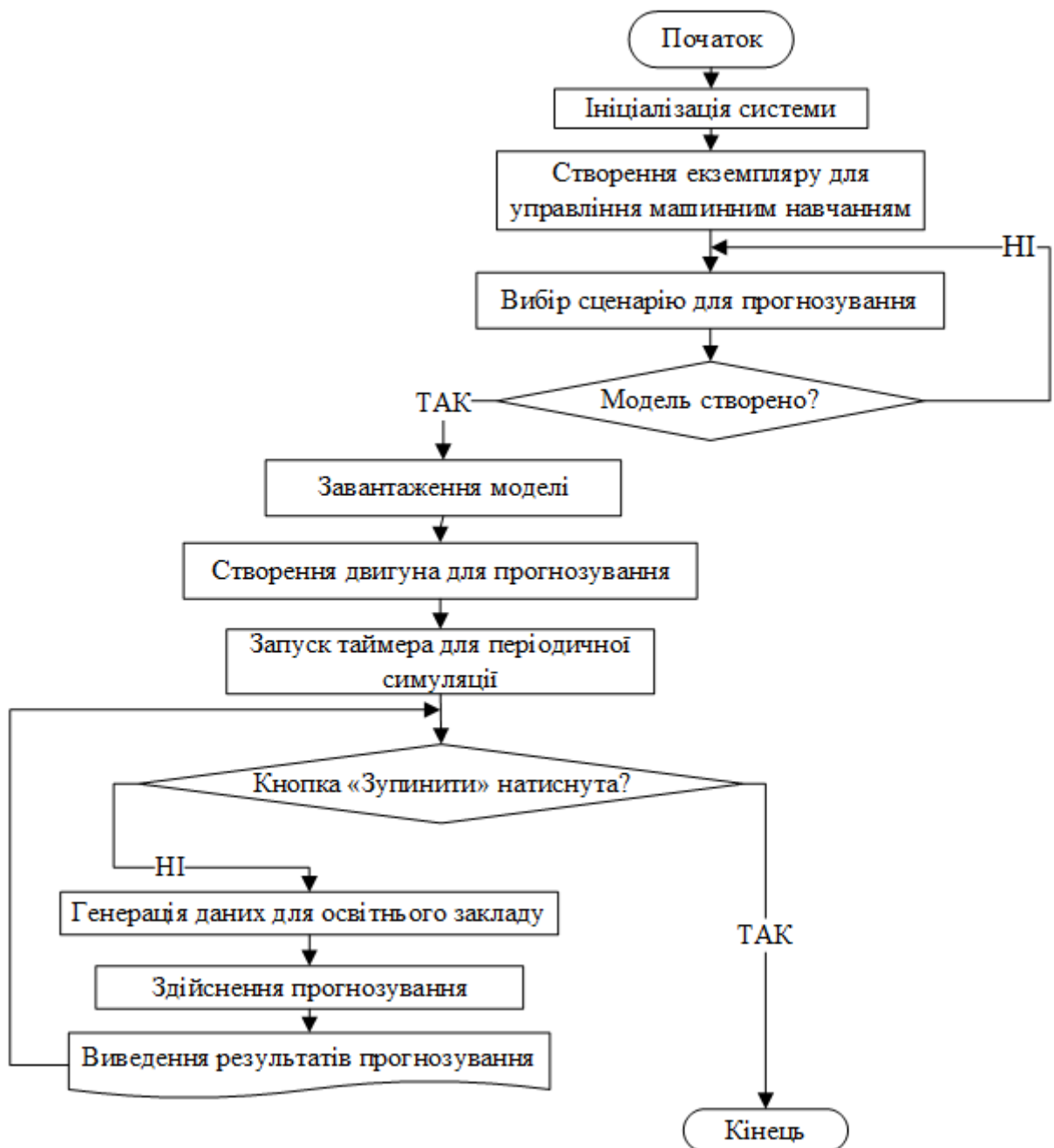


Рисунок 2.5 – Алгоритм процесу прогнозування

Розпочавши з ініціалізації, система швидко переходить до етапу створення інструментів управління машинним навчанням, де перевіряється наявність попередньо побудованої моделі. Якщо модель вже існує, вона завантажується, після чого ініціюється двигун прогнозування, що має застосувати модель для оцінки ефективності навчального закладу. У разі, коли моделі ще немає, система спершу запропонує вибрати сценарій для подальшого прогнозування.

Далі активується таймер, який відповідає за ритмічне проведення прогнозів, виходячи з регулярно оновлених даних. Система в очікуванні

введення даних або команди користувача розпочинає процес симуляції, під час якого відбувається генерація або оновлення інформації про освітній заклад. Отримані дані подаються до двигуна прогнозування, який використовує завантажену модель для оцінки поточної продуктивності та виведення оцінки загального балу закладу, що є ключовим показником для визначення світового рейтингу.

Процес закінчується наданням користувачеві детальних результатів прогнозу, які можуть бути використані для стратегічного планування або подальших досліджень. Весь алгоритм прогнозування вирізняється своєю циклічністю та здатністю до адаптації, оскільки він постійно оновлюється з новими даними і може бути перезапущений для вдосконалення прогнозних моделей.

3 КОНСТРУЮВАННЯ, ВЕРИФІКАЦІЯ ТА АНАЛІЗ ПРОГРАМНОЇ КОМПОНЕНТИ

3.1 Проектування та розробка схеми бази даних

Початковий етап проектування об'єкта включає ретельне створення структури бази даних, яка є ключовою компонентою системи. Для виконання цієї задачі було обрано методологію «сутність-зв'язок», відому як ER-метод. Цей метод є одним з найефективніших для розробки логічних моделей баз даних завдяки його здатності чітко визначати основні елементи предметної області та їх взаємозв'язки.

Застосування ER-методу дозволяє провести детальний аналіз предметної області, що сприяє ідентифікації всіх потрібних сутностей, їх атрибутів та зв'язків між ними. Такий підхід дозволяє створити якісну, надійну та масштабовану базу даних, яка адекватно представляє реальний світ і взаємодії у ньому.

Протягом детального проектування бази даних були ідентифіковані основні сутності, які відображають ключові об'єкти та концепції предметної області, а також прив'язані до них атрибути, які описують характеристики кожної сутності:

Таблиця 3.1 – Опис атрибутів сутності «Users» (Користувачі)

№	Найменування	Тип даних	Призначення
1	UserId	INT	Ідентифікатор користувача
2	FirstName	NVARCHAR(50)	Ім'я
3	LastName	NVARCHAR(50)	Прізвище
4	UserName	NVARCHAR(50)	Ім'я користувача для входу
5	UsersPassword	NVARCHAR(150)	Пароль
6	RoleId	INT	Ідентифікатор ролі
7	Description	NVARCHAR(1000)	Опис
8	Email	NVARCHAR(150)	Електронна пошта

Таблиця 3.2 – Опис атрибутів сутності «Models» (Моделі)

№	Найменування	Тип даних	Призначення
1	ModelsId	INT	Ідентифікатор моделі
2	ModelsName	NVARCHAR(150)	Назва моделі
3	CategoriesId	INT	Ідентифікатор категорії
4	ModelsFileModel	NVARCHAR(MAX)	Файл моделі
5	Description	NVARCHAR(MAX)	Опис моделі

Таблиця 3.3 – Опис атрибутів сутності «UniversityData» (Дані університету)

№	Найменування	Тип даних	Призначення
1	UniversityDataId	INT	Ідентифікатор даних університету
2	QualityOfEducation	FLOAT	Якість освіти
3	AlumniEmployment	FLOAT	Рівень зайнятості випускників
4	QualityOfFaculty	FLOAT	Якість викладацького складу
5	Publications	FLOAT	Кількість публікацій
6	Influence	FLOAT	Рівень впливу
7	Citations	FLOAT	Кількість цитувань
8	BroadImpact	FLOAT	Широкий вплив
9	Patents	FLOAT	Кількість патентів
10	Score	FLOAT	Оцінка
11	CategoriesId	INT	Ідентифікатор категорії

Таблиця 3.4 – Опис атрибутів сутності «Logs» (Логи подій)

№	Найменування	Тип даних	Призначення
1	LogsId	INT	Ідентифікатор логу
2	UsersId	INT	Ідентифікатор користувача
3	EventNameShow	NVARCHAR(MAX)	Назва події
4	EventDate	DATETIME	Дата події

Таблиця 3.5 – Опис атрибутів сутності «Categories» (Категорії)

№	Найменування	Тип даних	Призначення
1	CategoriesId	INT	Ідентифікатор категорії
2	CategoriesName	NVARCHAR(250)	Назва категорії
3	Description	NVARCHAR(MAX)	Опис категорії

У процесі розробки бази даних була звернена увага на організацію взаємозв'язків між сутностями, що є ключовим аспектом для забезпечення ефективності та надійності системи. Використання зовнішніх та внутрішніх ключів дозволило точно визначати зв'язки та їхні типи між таблицями.

Застосування зовнішніх ключів забезпечило зв'язки між таблицями, важливі для підтримки цілісності даних. Внутрішні ключі, у свою чергу, гарантували унікальність записів у межах однієї таблиці.

Розробка зв'язків між таблицями враховувала специфіку предметної області, спрямована на забезпечення майбутнього масштабування системи через детальний аналіз взаємозв'язків даних та їхньої ролі у загальній структурі додатку.

В процесі проектування бази даних наступні зв'язки між таблицями:

- зв'язок між таблицею «Categories» та «Models» через поле «CategoriesId». Вид цього зв'язку є «один до багатьох» (1:N). Одна категорія може бути асоційована з багатьма моделями в таблиці «Models», але кожна модель прив'язана лише до однієї категорії;

- зв'язок між таблицею «Categories» та «UniversityData» через поле «CategoriesId». Вид зв'язку – «один до багатьох» (1:N). Одна категорія може мати кілька записів університетських даних, але кожен запис університетських даних відноситься до однієї категорії;

- зв'язок між таблицею «Users» та «Logs» через поле «UsersId». Вид зв'язку – «один до багатьох» (1:N). Один користувач може мати кілька записів логів в таблиці «Logs», але кожен запис логу відноситься до одного користувача.

Використовуючи таблиці та встановлюючи зв'язки, була створена фізична модель бази даних, що представлена на рис. 3.1.

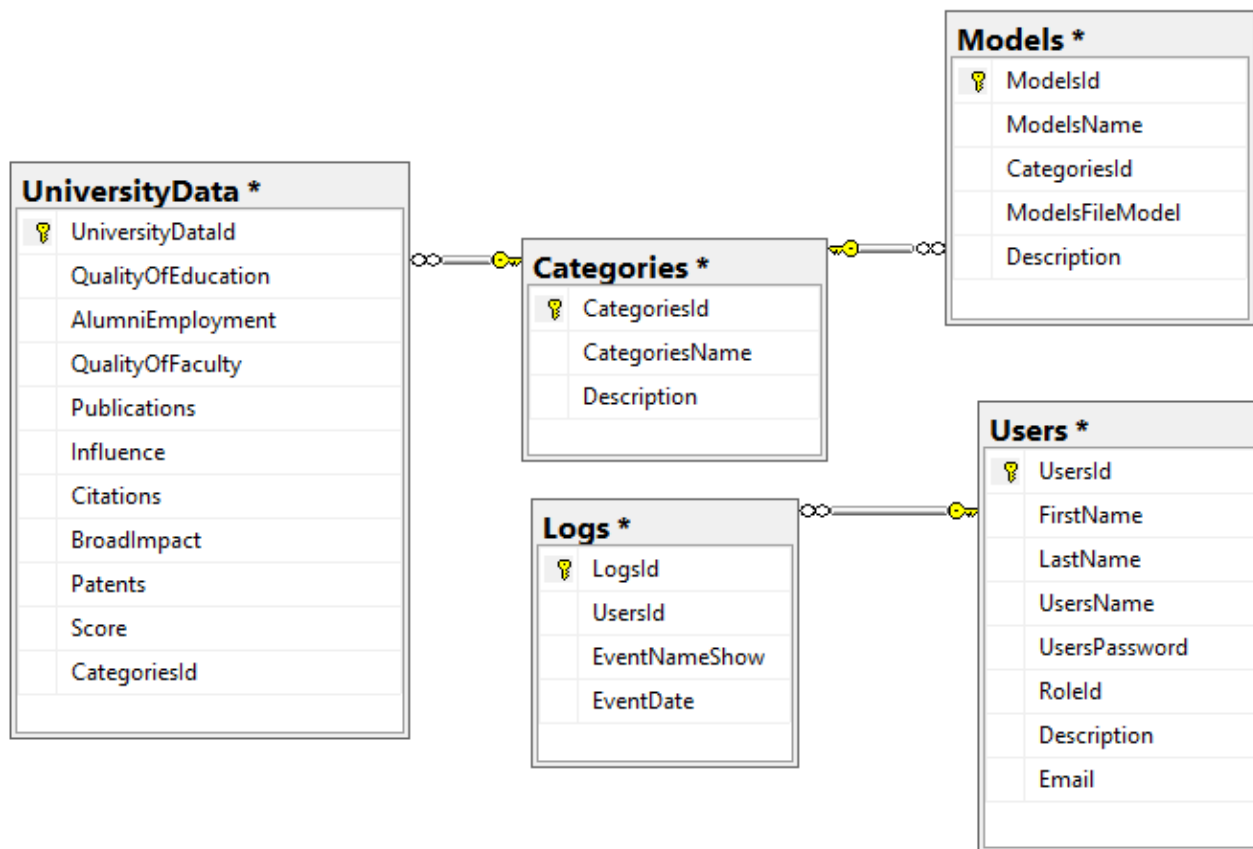


Рисунок 3.1 – Фізична модель бази даних

Отже, база даних системи була спроектована та реалізована з метою забезпечення максимальної цілісності та оптимального управління даними. Впровадження такої структури дозволить ефективно збирати, зберігати та обробляти інформацію, що сприятиме підвищенню продуктивності та надійності системи.

3.2 Проектування діаграм основних бізнес-процесів

Розробка проекту передбачає створення діаграм активності та послідовності, що є ключовими для визначення логіки функціонування системи та її взаємодії з компонентами. Діаграми активності служать для графічного

представлення бізнес-процесів та робочих потоків, сприяючи аналізу та оптимізації маршрутів обробки інформації та прийняття рішень в системі.

Ці діаграми відображають сценарії реакцій системи на зовнішні події, виділяючи важливі моменти взаємодії користувачів з системою. Вони демонструють потоки даних та керовані шляхи, що лежать в основі алгоритмів та механізмів роботи системи. Наприклад, для процедури авторизації користувача була створена діаграма активності, представлена на рис. 3.2.

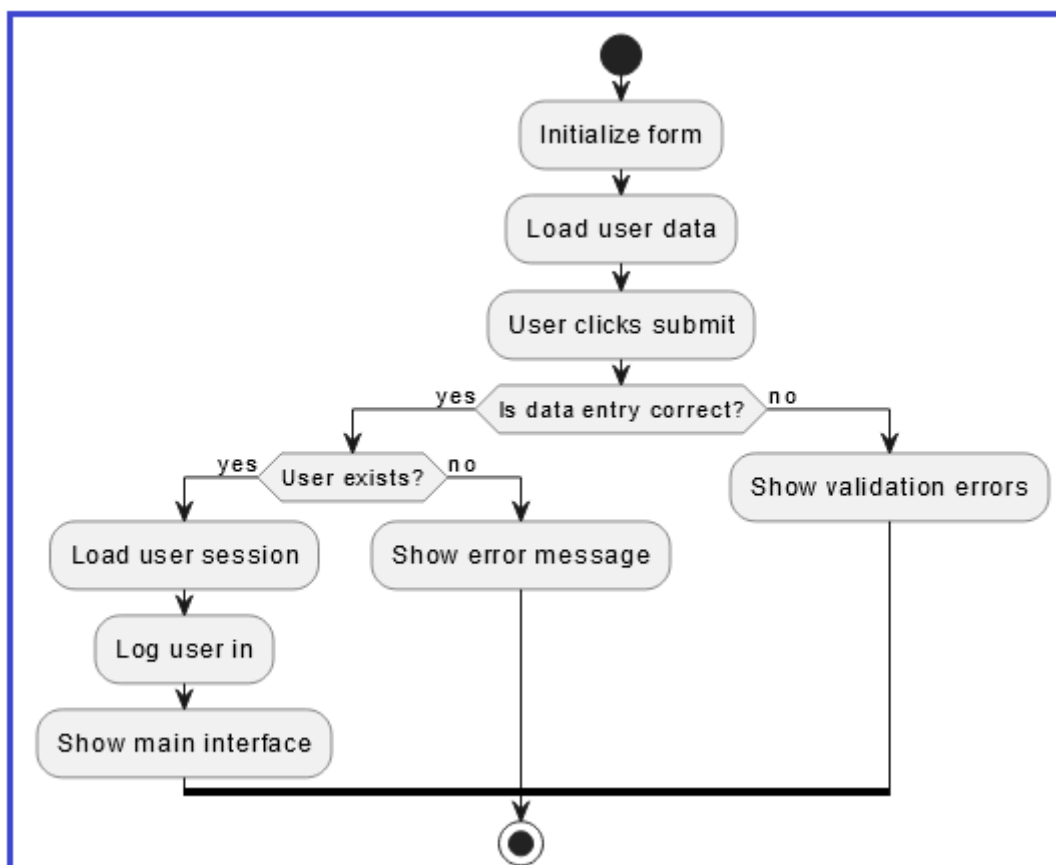


Рисунок 3.2 – Процес авторизації користувача

Діаграма активності для процесу авторизації користувача починається з ініціалізації форми для входу в систему, що готує її до взаємодії з користувачем. На наступному кроці система завантажує дані вже зареєстрованих користувачів, які використовуються під час аутентифікації. Користувач натискає кнопку подання даних після заповнення форми.

У разі коректного введення даних система перевіряє, чи існує користувач з вказаними ідентифікаторами. Якщо користувач існує, ініціюється завантаження його сесії, відбувається логін і відображається головний інтерфейс

програми. Якщо користувач не існує, користувачеві показується повідомлення про помилку. У разі некоректного введення даних відображаються помилки валідації.

Крок реєстрації нового користувача активується, коли особа вирішує створити новий аккаунт, натискаючи на відповідне посилання чи кнопку на формі. Ця діаграма охоплює весь процес входу в систему, від ініціалізації форми до завершення входу або відображення помилок.

На рис. 3.1 представлена діаграма активності для процесу навчання та додавання моделі в систему.

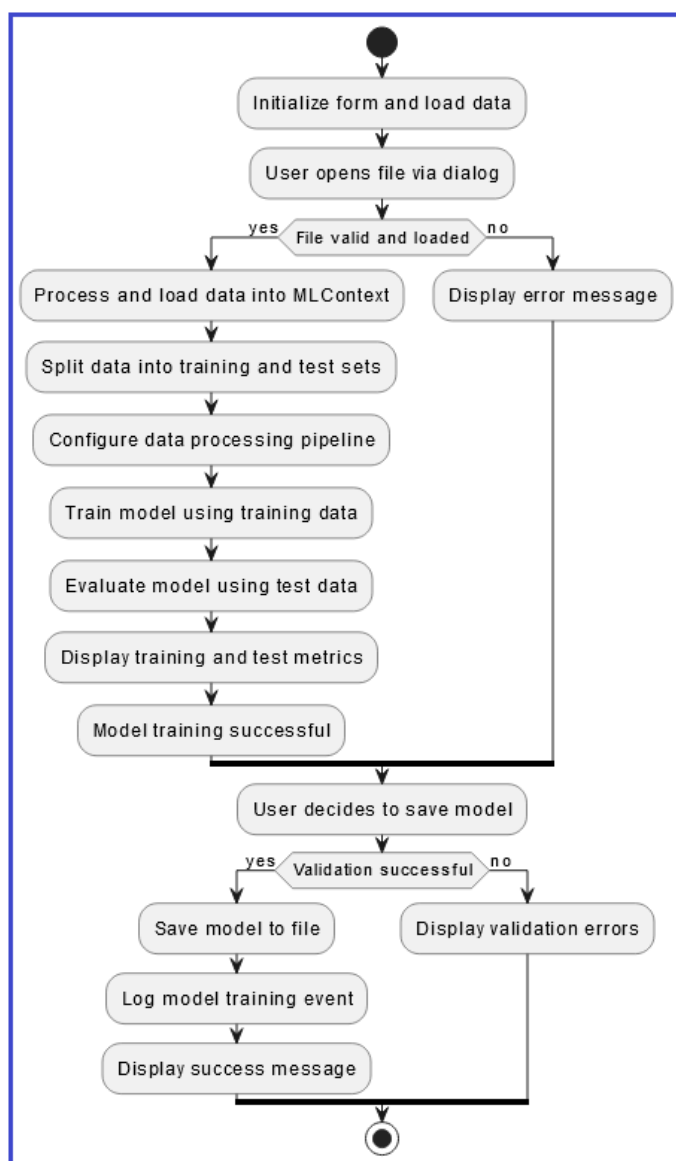


Рисунок 3.3 – Процес навчання та додавання моделі в систему

Діаграма активності для процесу навчання та додавання моделі в систему розпочинається з ініціалізації форми та завантаження необхідних даних. Після цього користувач через діалогове вікно вибирає файл для навчання моделі.

У разі успішного завантаження та валідності файлу, дані обробляються і завантажуються у контекст машинного навчання, поділяються на тренувальний та тестовий набори, налаштовуються етапи конвеєру обробки даних. Далі відбувається навчання моделі на тренувальному наборі та її оцінка на тестовому, відображаються метрики тренування та тестування, і якщо все пройшло успішно, модель вважається навченою. У випадку невалідності файлу користувачу показується повідомлення про помилку.

Коли користувач вирішує зберегти модель, якщо всі дані введено правильно, модель зберігається у файл, інформація про навчання реєструється у журналі, а користувачу показується повідомлення про успіх. У разі помилок у введенні даних користувачу відображаються помилки валідації.

В межах проекту розроблено діаграму послідовності, яка відображає процес генерації даних та прогнозування загального балу навчального закладу (рис. 3.4).

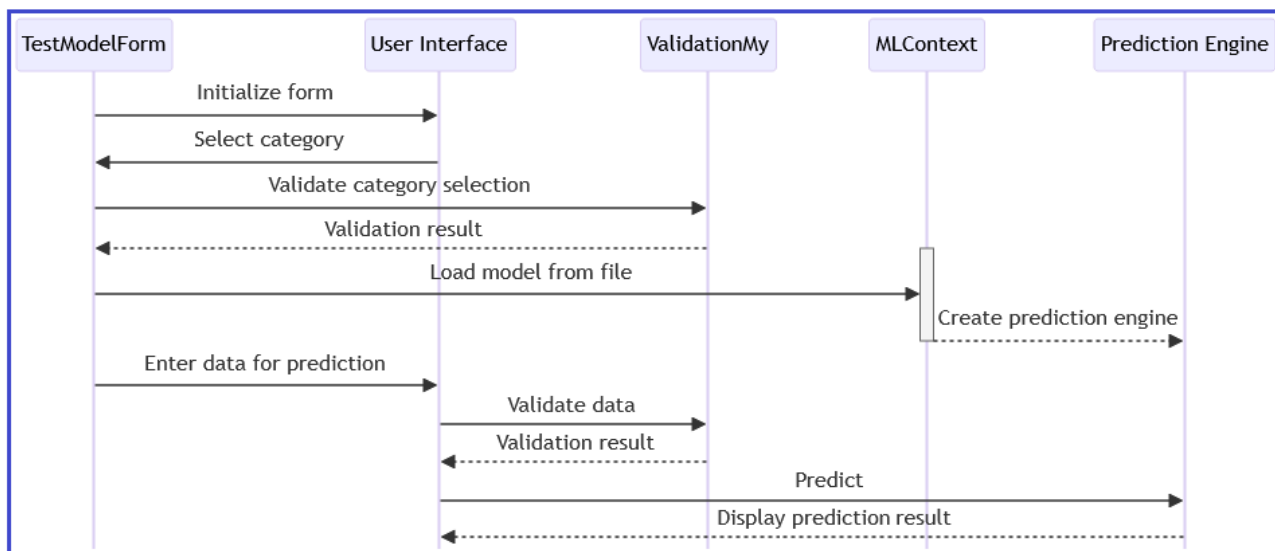


Рисунок 3.4 – Процес генерації даних та прогнозування загального балу

Діаграма послідовності для процесу генерації даних та прогнозування загального балу навчального закладу включає починається із ініціалізації форми, коли користувач відкриває вікно для тестування моделі. Користувач вибирає

категорію з випадуючого списку на інтерфейсі, що активує процес перевірки валідності. Виконується перевірка валідності обраної категорії, результат валідації повертається на форму. Якщо категорія валідна, система завантажує модель з файлу використовуючи `MLContext`. Після завантаження моделі ініціалізується механізм прогнозування. Користувач вводить необхідні дані в поля інтерфейсу для розрахунку прогнозу. Перевіряється правильність введених даних, результат валідації повертається користувачу через інтерфейс. Якщо дані валідні, виконується прогнозування за допомогою механізму прогнозування. Результат прогнозування відображається на інтерфейсі для користувача.

Ця діаграма показує взаємодію між різними компонентами системи під час генерації даних та прогнозування загального балу навчального закладу, демонструючи послідовність етапів і перевірок в процесі.

3.3 Імплементация програмних компонентів

Розробка коду почалася з визначення рівня даних, на якому були створені моделі, що представляють ключові бізнес-об'єкти та їхні взаємозв'язки. В проєкті застосовується база даних MS SQL Server для збереження інформації. З'єднання з базою даних здійснюється через змінну «CONNECT», визначену у конфігураційному файлі. Параметри налаштувань цього з'єднання представлені на рис. 3.5.

```
<appSettings>
  <add key="CONNECT" value="Data Source=(LocalDB)\MSSQLLocalDB;
    AttachDbFilename=|DataDirectory|\DB.mdf;
    Integrated Security=True" />
</appSettings>
```

Рисунок 3.5 – Параметри з'єднання з базою даних

Для початку було реалізовано клас «`ModelTeachForm`», який є формою для навчання моделей та містить ряд методів та властивостей для взаємодії з користувачем та обробки даних. Він включає в себе наступні методи:

– метод «LoadAllDate» викликається при ініціалізації форми та завантажує всі категорії з бази даних. Потім він встановлює джерело даних для ComboBox, встановлює значення для ValueMember та DisplayMember, щоб забезпечити відображення категорій у ComboBox;

– метод «DataLoad» відповідає за завантаження даних моделей з бази даних та їх відображення у вигляді таблиці. Він також забезпечує вибір першого рядка таблиці та перехід до нього при перезавантаженні даних;

– метод «LoadDataInModelsGridView» відповідає за завантаження даних в DataGridView. Він очищує попередні дані в таблиці, налаштовує колонки та їх властивості, а потім встановлює джерело даних для DataGridView, щоб відобразити нові дані;

– метод «OpenBtn_Click» викликається при натисканні кнопки «Відкрити». Він відкриває діалогове вікно для вибору файлу, завантажує дані з обраного файлу та навчає модель машинного навчання на цих даних. Після цього відображає інформацію про тренування моделі;

– метод «AddBtn_Click» викликається при натисканні кнопки «Додати». Він перевіряє коректність введених даних, зберігає навчену модель у файл та зберігає інформацію про модель у базу даних. Після цього очищує дані у вікні;

– метод «ClearBtn_Click» викликається при натисканні кнопки «Очистити». Він очищує всі поля та відображену інформацію у вікні;

– метод «ExitBtn_Click» викликається при натисканні кнопки «Вихід». Він закриває вікно програми;

– метод «GenerateFileName» генерує унікальне ім'я файлу на основі поточного часу та дати. Використовується для створення імені файлу для збереження навченої моделі;

– метод «ClearAllData» очищує всі дані у вікні, встановлюючи їх початкові значення;

– метод «IsDataEnteringCorrect» перевіряє коректність введених даних перед їх збереженням. Він перевіряє, чи навчена модель і чи обрано категорію, та відображає відповідні повідомлення про помилки;

– метод «ModelsGridView_CellClick» обробляє подію натисканням ЛКМ на комірку таблиці моделей. Якщо вибрана кнопка «Видалити», він викликає підтвердження видалення моделі та видаляє її з бази даних.

Для взаємодії з системою було створено користувацький інтерфейс, зображення якого наведено на рис. 3.6.

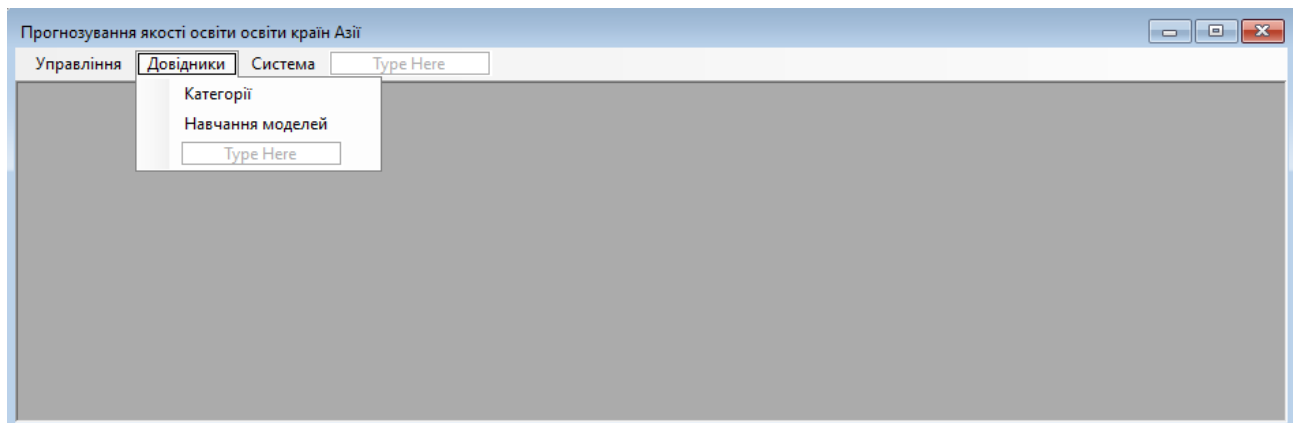


Рисунок 3.6 – Головний інтерфейс програми

Кожен елемент меню обладнано обробниками подій, які активуються при взаємодії з користувачем. На рис. 3.7 показано приклад обробника події для кнопки «Категорії».

```
private void категоріїToolStripMenuItem_Click(object sender, EventArgs e) {
    CloseAllWin();
    CategoriesForm categoriesForm = new CategoriesForm();
    categoriesForm.MdiParent = this;
    categoriesForm.WindowState = FormWindowState.Maximized;
    categoriesForm.Show();
}
```

Рисунок 3.7 – Код обробника події кнопки «Категорії»

Як зазначено на рис. 3.7, обробник події спершу запускає метод «CloseAllWin», що закриває усі відкриті вікна або вкладки головної форми. Після цього відбувається ініціалізація нової форми «CategoriesForm», яка слугує для відображення категорій. Ця форма встановлюється як дочірня до основної форми, її стан максимізується, і вона відображається на повний екран. В

результаті, форма «Категорії» інтегрується в основний інтерфейс як нова вкладка, займаючи весь доступний простір екрану.

Форму для створення категорій моделей відображено на рис. 3.8.

Рисунок 3.8 – Форма для опрацювання категорій моделей

Клас `CategoriesForm` відповідає за функціонал управління категоріями. Основні методи цього класу складаються із:

- конструктор `CategoriesForm` викликається при створенні екземпляра класу `CategoriesForm`. Він викликає метод `InitializeComponent()` для ініціалізації компонентів форми та потім викликає метод `DataLoad()` для завантаження даних категорій;

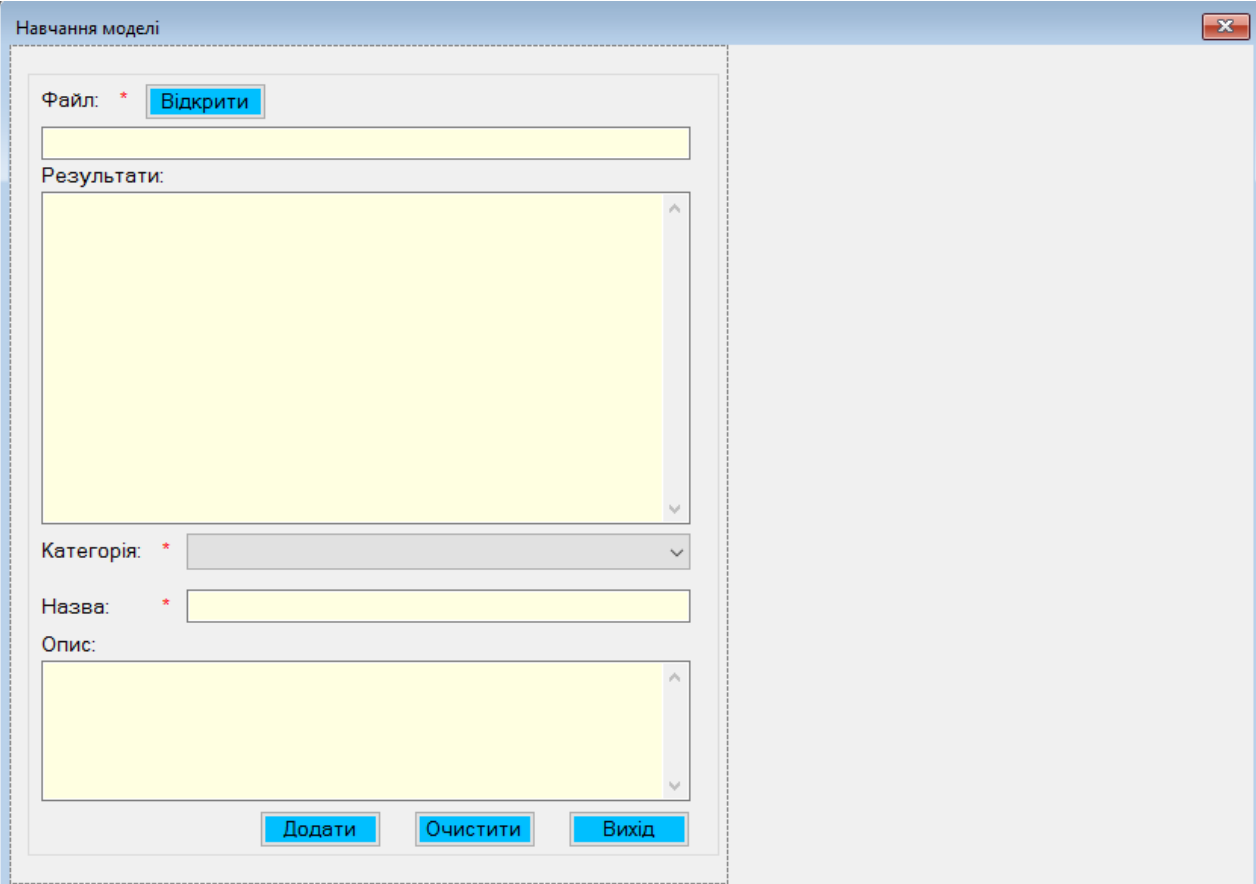
- метод `AddBtn_Click` викликається при натисканні кнопки «Додати». Він перевіряє коректність введених даних, додає нову категорію в базу даних за допомогою об'єкта `_CategoriesProvider`, після чого викликає метод `DataLoad()` для оновлення відображення категорій та метод `ClearAllControls()` для очищення полів введення;

- метод `ClearBtn_Click` викликається при натисканні кнопки «Очистити». Він очищає текстові поля для введення нової категорії;

- метод `ExitBtn_Click` викликається при натисканні кнопки «Вихід». Він закриває вікно програми;

- метод `DataLoad` завантажує дані про категорії з бази даних та відображає їх у вигляді таблиці. Він також забезпечує вибір першого рядка таблиці для коректного відображення;
- метод `LoadDataInLekarstvaGridView` відповідає за завантаження даних у `DataGridView`. Він налаштовує колонки таблиці та встановлює джерело даних для `DataGridView`;
- метод `ClearAllControls` очищає всі текстові поля у вікні для введення нової категорії;
- метод `IsDataEnteringCorrect` перевіряє коректність введених даних перед додаванням нової категорії. Він перевіряє, чи введено назву категорії;
- метод `CategoriesGridView_CellClick` обробляє подію клікання на комірку таблиці категорій. Він визначає вибраний рядок та викликає форму для оновлення вибраної категорії. Після цього викликає метод `DataLoad()` для оновлення відображення категорій.

Для тренування моделі розроблено форму, що представлена на рис. 3.9.



Навчання моделі

Файл: *

Результати:

Категорія: *

Назва: *

Опис:

Рисунок 3.9 – Форма для тренування моделей

Клас «ModelTeachForm» відповідає за функціонал навчання та збереження моделей машинного навчання. Його методи складаються із:

- конструктор «ModelTeachForm» викликається при створенні екземпляра класу «ModelTeachForm». Він ініціалізує компоненти форми за допомогою метода «InitializeComponent», завантажує дані категорій за допомогою метода «LoadAllDate» та відображає моделі за допомогою метода DataLoad;

- метод «LoadAllDate» завантажує всі доступні категорії та додає їх до випадуючого списку «CategoriesCBox»;

- метод «DataLoad» завантажує дані про моделі з бази даних та відображає їх у вигляді таблиці. Він також забезпечує вибір першого рядка таблиці для коректного відображення;

- метод «LoadDataInModelsGridView» відповідає за завантаження даних у DataGridView, який відображає моделі. Він також налаштовує вигляд колонок таблиці;

- метод «OpenBtn_Click» викликається при натисканні кнопки «Відкрити». Він створює діалогове вікно для відкриття файлу, завантажує дані з цього файлу, навчає модель за допомогою ML.NET та виводить метрики навчання;

- метод «AddBtn_Click» викликається при натисканні кнопки «Додати». Він перевіряє коректність введених даних та зберігає навчену модель та її параметри в базу даних;

- метод «ClearBtn_Click» викликається при натисканні кнопки «Очистити». Він очищає всі дані у формі;

- метод «ExitBtn_Click» викликається при натисканні кнопки «Вихід». Він закриває вікно програми;

- метод «GenerateFileName» генерує унікальне ім'я файлу для збереження навченої моделі;

- метод «ClearAllData» очищає всі дані у формі та перезавантажує дані про моделі;
- метод «IsDataEnteringCorrect» перевіряє коректність введених даних перед збереженням моделі;
- метод «ModelsGridView_CellClick» обробляє подію клікання на комірку таблиці моделей. Якщо користувач підтверджує бажання видалити модель, вона видаляється з бази даних.

Клас «ModelTeachForm» використовує зовнішні провайдери для взаємодії з базою даних та реалізації валідації, що дозволяє ізолювати логіку роботи з даними від UI логіки. Це сприяє підтримці принципу розділення обов'язків та полегшує подальше розширення та тестування програми. Крім того, клас використовує функціонал ML.NET для навчання моделей машинного навчання на основі наданих даних, що дозволяє створювати та використовувати прогностичні моделі для аналізу даних та прийняття рішень. Такий підхід забезпечує ефективне та гнучке використання інструментів для аналізу та обробки даних у програмі.

3.4 Реалізація алгоритму машинного навчання в системі

Реалізація алгоритму машинного навчання відіграє ключову роль у розробці систем, які використовують штучний інтелект для рішення певних завдань. В цьому підрозділі описаний процес імплементації алгоритму машинного навчання з використанням методу СДС, який є продуктивним інструментом для оптимізації в процесі навчання моделей.

Метод СДС вибирається через його здатність ефективно оптимізувати великі і складні невіпуклі функції. Цей метод використовує ітеративний підхід, де окремі координати параметрів моделі оптимізуються одна за одною стохастичним чином. Такий підхід дозволяє ефективно знаходити локальні

мінімуми без необхідності розрахунку повного градієнту на кожному кроці, що значно скорочує обчислювальні витрати та підвищує швидкість навчання моделей.

На початку реалізації алгоритму проведено оголошення глобальних змінних (рис. 3.10).

```
private MLContext mlContext;
private ITransformer trainedModel;
private IDataView dataView;
```

Рисунок 3.10 – Оголошення глобальних змінних в коді

У фрагменті коду оголошено три приватні змінні для роботи з машинним навчанням в контексті .NET, використовуючи бібліотеку ML.NET. Перша змінна `mlContext` є екземпляром класу `MLContext`, який служить для створення, тренування, оцінки та використання моделей машинного навчання. Друга змінна `trainedModel` є типу `ITransformer`. Ця змінна представляє навчену модель машинного навчання, яка може бути використана для перетворення даних або робити нові передбачення. Третя змінна `dataView` типу `IDataView` представляє дані у форматі, придатному для обробки бібліотекою ML.NET. `IDataView` є флексибельним контейнером для зберігання даних, які використовуються для тренування або тестування моделі. Ці змінні використовуються для роботи з даними та моделями у визначених процесах навчання і прогнозування.

Після цього необхідно провести ініціалізацію контексту для машинного навчання (рис. 3.11).

```
context = new MLContext(seed: 0);
```

Рисунок 3.11 – Ініціалізацію контексту машинного навчання

У цьому рядку коду створюється новий екземпляр класу `MLContext`, який є основним об'єктом для всіх операцій з машинним навчанням у бібліотеці ML.NET. `MLContext` виступає як початкова точка, що дозволяє взаємодіяти з багатьма функціями бібліотеки, такими як завантаження даних, конфігурація процесів обробки даних, тренування моделей, і тестування або застосування моделей для роботи з новими даними. Параметр `seed: 0` використовується для ініціалізації генератора випадкових чисел всередині контексту `MLContext`, що

забезпечує відтворюваність результатів машинного навчання при кожному запуску коду з цим же насінням. Це особливо корисно при налагодженні та порівнянні моделей машинного навчання, оскільки використання одного і того ж насіння гарантує однакові умови при різних запусках алгоритму.

Далі відбувається розділення даних на навчальний та тестовий набори за допомогою методу «TrainTestSplit» з бібліотеки ML.NET (рис. 3.12).

```
// Розділення даних на навчальний набір і тестовий набір
var split = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
```

Рисунок 3.12 – Розділення даних на навчальний та тестовий набори

Метод приймає два аргументи: `dataView`, який є даними у форматі, придатному для обробки машинним навчанням, та `testFraction`, який вказує частку даних, що має бути відведена для тестового набору. У цьому випадку `testFraction` дорівнює 0.2, що означає, що 20% даних будуть використовуватися для тестування моделі, а решта 80% — для навчання.

Цей процес є критично важливим у машинному навчанні, оскільки дозволяє оцінити, наскільки добре навчена модель зможе працювати з новими, раніше невідомими даними. Розділення даних допомагає запобігти перенавчанню, коли модель ідеально працює на навчальних даних, але погано на нових. Використання незалежного тестового набору дає більш об'єктивну оцінку ефективності моделі перед її впровадженням у реальні умови.

Після цього проводиться побудова конвеєру для навчання моделі (рис. 3.13).

```
// Побудова конвеєру
var dataProcessPipeline = mlContext.Transforms.Concatenate("Features",
    new[] { "QualityOfEducation", "AlumniEmployment", "QualityOfFaculty", "Publications",
        "Influence", "Citations", "BroadImpact", "Patents" })
    .Append(mlContext.Transforms.CopyColumns(outputColumnName: "Label", inputColumnName: "Score"))
    .Append(mlContext.Transforms.NormalizeMinMax("Features"));
```

Рисунок 3.13 – Побудова конвеєра

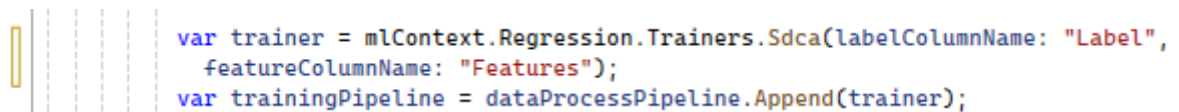
Код створює конвеєр обробки даних для машинного навчання з використанням бібліотеки ML.NET. Початково використовується метод `Concatenate` з класу `Transforms` для об'єднання декількох стовпців даних в один стовпець з назвою «Features». Ці стовпці включають «QualityOfEducation»,

«AlumniEmployment», «QualityOfFaculty», «Publications», «Influence», «Citations», «BroadImpact», «Patents», і представляють різні характеристики, які можуть використовуватися для оцінювання університетів.

Далі, за допомогою методу CopyColumns, створюється новий стовпець «Label», який є копією вхідного стовпця «Score». Це стандартний підхід у задачах навчання під наглядом, де «Label» використовується як цільова змінна, що позначає значення, яке модель повинна передбачити. Також використовується метод NormalizeMinMax для нормалізації даних у стовпці «Features». Цей метод масштабує всі числові характеристики до діапазону між 0 та 1, що часто необхідно для багатьох алгоритмів машинного навчання, оскільки вони чутливі до масштабу даних.

Конвеєр обробки даних готує вхідні дані таким чином, щоб вони могли бути ефективно використані у моделі машинного навчання, допомагаючи забезпечити більш стабільні та точні передбачення.

На наступному етапі вибирається тренер та додається до попередньо визначеного конвеєру (рис. 3.14).



```
var trainer = mlContext.Regression.Trainers.Sdca(labelColumnName: "Label",
featureColumnName: "Features");
var trainingPipeline = dataProcessPipeline.Append(trainer);
```

Рисунок 3.14 – Код навчання нейронної мережі

У коді спочатку створюється об'єкт trainer за допомогою методу Sdca, який є частиною бібліотеки регресійних тренерів в ML.NET. Метод Sdca відноситься до стохастичного градієнтного спуску з подвійним координатним сходженням, що використовується для тренування лінійних регресійних моделей. Цей тренер конфігурується із вказівкою на стовпець «Label» як цільову змінну та «Features» як набір характеристик, за якими відбувається тренування.

Далі, створений тренер додається до попередньо визначеного конвеєру обробки даних dataProcessPipeline через метод Append. Це дія розширює конвеєр, додаючи до нього крок тренування моделі. Конвеєр, який тепер включає як обробку даних, так і тренування моделі, стає повноцінним тренувальним конвеєром. Цей підхід дозволяє систематично та ефективно підготувати дані і

одразу застосувати алгоритм машинного навчання для створення прогнозованої моделі, забезпечуючи оптимальне використання даних і ресурсів обчислень.

Після створення конвеєру необхідно додати код для тренування моделі (рис. 3.15).

```
//Навчання
trainedModel = trainingPipeline.Fit(split.TrainSet);
```

Рисунок 3.15 – Навчання моделі

Даний рядок коду використовується раніше створений тренувальний конвеєр, щоб навчити модель на основі набору даних для навчання. Метод `Fit` застосовується до тренувального конвеєра `trainingPipeline`, і як аргумент передається частина даних, виділена для тренування (`TrainSet`), яка була отримана після розділення загального набору даних на навчальний та тестовий набори.

Використання методу `Fit` на тренувальному конвеєрі дозволяє автоматично обробити вхідні дані через всі етапи конвеєру, що включає обробку, нормалізацію та фінальне тренування моделі з використанням вибраного алгоритму машинного навчання. Після завершення цього процесу, `trainedModel` стає екземпляром `ITransformer`, який представляє навчену модель. Ця модель готова для використання у прогнозуванні або подальшій оцінці на тестовому наборі даних або реальних даних в майбутньому.

Кожну навчену модель нейронної мережі потрібно піддати перевірці ефективності, тому для цього теж необхідно реалізувати код (рис. 3.16).

```
// Передбачення для тестового набору даних
var predictions = model.Transform(testData);
// Оцінка якості моделі
var metrics = context.BinaryClassification.Evaluate(predictions, "Label");
ReportTextBox.Text += ("Log-loss: " +
    $"{metrics.LogLoss}") + "\r\n";
ReportTextBox.Text += ("Area Under ROC Curve: " +
    $"{metrics.AreaUnderRocCurve}") + "\r\n";
ReportTextBox.Text += ("Area Under Precision-Recall Curve: " +
    $"{metrics.AreaUnderPrecisionRecallCurve}") + "\r\n";
ReportTextBox.Text += ("Accuracy: {metrics.Accuracy}") + "\r\n";
```

Рисунок 3.16 – Виведення метрик моделі

Даний код використовується для оцінки якості моделі бінарної класифікації, створеної за допомогою Microsoft ML.NET, і відображення результатів у текстовому полі. Спочатку проводиться передбачення для тестового набору даних за допомогою моделі. Отримані передбачення зберігаються у змінній `predictions`. Далі використовується метод `Evaluate` для оцінки якості моделі. Результати зберігаються у змінній `metrics`.

Оцінка якості моделі включає різні метрики, такі як лог-втрата (Log-loss), площа під кривою ROC (Area Under ROC Curve), площа під кривою точності-повторного виклику (Area Under Precision-Recall Curve) та точність (Accuracy). Результати кожної з цих метрик додаються до текстового поля `RaportTBox`.

Кожна метрика додається у форматі тексту з новими рядками після кожного значення. Це забезпечує, що кожна метрика відображається у новому рядку в текстовому полі, що дозволяє користувачу легко бачити результати оцінки моделі.

Щоб кожного разу не проводити навчання моделі для проведення прогнозування, можна зберегти результати тренування моделі у файл для подальшого її використання (рис. 3.17).

```
// Оцінка моделі на тренувальному наборі даних
var predictions = trainedModel.Transform(split.TestSet);
var metrics = mlContext.Regression.Evaluate(predictions,
    labelColumnName: "Label", scoreColumnName: "Score");
//Виведення метрик для навчального набору
RaportTBox.Text += "Training Data Metrics:" + "\r\n";
RaportTBox.Text += ("R^2: " +
    $"{Math.Round(Math.Abs(Convert.ToDecimal(metrics.RSquared)), 2)}") + "\r\n";
RaportTBox.Text += ("MAE: " +
    $"{Math.Round(metrics.MeanAbsoluteError, 2)}") + "\r\n";
```

Рисунок 3.17 – Код збереження даних моделі

Цей код описує процес оцінки ефективності навченої моделі машинного навчання за допомогою тестового набору даних і виведення результатів оцінки.

Спочатку, використовуючи навчену модель `trainedModel`, відбувається трансформація тестового набору даних `TestSet`, який містить приклади, не використані під час тренування моделі. Функція `Transform` застосовує модель до тестових даних, результатом чого є набір передбачень `predictions`.

Далі, за допомогою методу Evaluate з бібліотеки ML.NET проводиться оцінка цих передбачень. У параметрах методу вказано, що стовпець «Label» у даних є фактичними значеннями, а «Score» – передбаченими моделлю. Метод Evaluate повертає набір метрик, які дозволяють оцінити якість моделі, наприклад, коефіцієнт детермінації R^2 .

Після отримання метрик відбувається їх виведення у текстовому блоці ReportBox. Стрічка, що додається, містить інформацію про метрики навчального набору даних, зокрема значення R^2 , яке адаптовано через обчислення його модуля та округлення до двох знаків після коми. Таке форматування допомагає зробити результати більш зрозумілими для користувачів, які аналізують ефективність моделі.

Для проведення тестування моделей навчених моделей реалізовано форму «TestModelForm». На початку оголошено ряд змінних, що представлені на рис. 3.18.

```
private Models _SelectedModels = new Models();
private MLContext _Context = new MLContext();
private PredictionEngine<UniversityData, UniversityScorePrediction> predictionEngine;
private ModelsProvider _ModelsProvider = new ModelsProvider();
```

Рисунок 3.18 – Оголошення основних змінних форми

Код ініціалізує кілька приватних змінних, які використовуються для управління різними аспектами машинного навчання в додатку на основі бібліотеки ML.NET.

Перша змінна, _SelectedModels, створюється як екземпляр класу Models. Цей клас може бути призначений для зберігання та управління різними моделями машинного навчання, які використовуються у програмі. Це дозволяє централізовано управляти доступними моделями, які можна вибирати для виконання конкретних задач.

Друга змінна, _Context, є інстанцією MLContext. MLContext є основним компонентом в ML.NET, який служить як стартовий пункт для всіх операцій машинного навчання, включно з завантаженням даних, конфігурацією обробки, тренуванням моделей та їхньою оцінкою. Він забезпечує необхідну

інфраструктуру та управління ресурсами, що потрібні для ефективного виконання цих задач.

Третя змінна, `predictionEngine`, є екземпляром `PredictionEngine<TSrc, TDst>`, де `TSrc` та `TDst` представляють вхідні та вихідні типи даних відповідно. У цьому випадку, вхідні дані представлені класом `UniversityData`, а вихідні – класом `UniversityScorePrediction`. `PredictionEngine` використовується для здійснення швидких передбачень на основі навченої моделі, що робить його ідеальним для розгортання в продакшн-середовищі, де потрібно швидко оцінювати нові дані.

Четверта змінна, `_ModelsProvider`, є екземпляром класу `ModelsProvider`. Цей клас, ймовірно, функціонує як репозиторій або провайдер для доступу до різних моделей, які можуть бути використані в додатку. Це дозволяє централізовано управляти та відповідно до потреб забезпечувати доступ до попередньо тренованих або конфігурованих моделей.

Разом ці компоненти створюють структуроване середовище для ефективної роботи з машинним навчанням, від управління моделями до виконання передбачень.

У конструкторі форми викликається метод `LoadAllDate`, що призначений для завантаження категорій моделей (рис. 3.19).

```
private void LoadAllDate() {
    _CategoriesList = _CategoriesProvider.GetAllCategories();
    CategoriesCBox.DataSource = _CategoriesList;
    CategoriesCBox.ValueMember = "CategoriesId";
    CategoriesCBox.DisplayMember = "CategoriesName";
    _IsThemesLoad = true;
    CategoriesCBox.SelectedValueChanged(CategoriesCBox, EventArgs.Empty);
}
```

Рисунок 3.19 – Код методу для «LoadAllDate»

Код визначає метод «LoadAllDate2», який виконує завантаження та відображення даних категорій у комбінований список (`ComboBox`) у графічному інтерфейсі користувача. Спочатку, метод отримує список всіх категорій через виклик методу «GetAllCategories» від провайдера «_CategoriesProvider». Цей список зберігається у змінній «_CategoriesList». Метод `GetAllCategories`

припускаємо, що повертає колекцію об'єктів, кожен з яких представляє категорію.

Далі «CategoriesCBox», який є комбінованим списком на формі, налаштовується так, що його джерело даних встановлюється як «_CategoriesList». Це дозволяє зв'язати дані категорій з візуальним елементом, забезпечуючи можливість їх вибору користувачем. Для коректного відображення та взаємодії з даними, «CategoriesCBox» налаштовується з використанням властивостей «ValueMember» та «DisplayMember». «ValueMember» встановлюється на «CategoriesId», що означає, що значення кожного елементу в списку буде відповідати ідентифікатору категорії. «DisplayMember» встановлюється на «CategoriesName», що означає, що відображені назви у списку будуть назвами категорій.

Змінна «_IsThemesLoad» встановлюється в true, що може бути використано для відстеження стану завантаження тем або категорій, індикуючи, що дані успішно завантажені та готові до використання. На кінець, викликається метод «CategoriesCBox_SelectedValueChanged» з параметрами «CategoriesCBox» та «EventArgs» (рис. 3.20). Цей виклик може бути спрямований на обробку події зміни вибраного значення в комбінованому списку, наприклад, для динамічного оновлення інших частин інтерфейсу відповідно до вибору користувача.

```

private void CategoriesCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (_IsThemesLoad && _CategoriesList[0].Message != NamesMy.NoDataNames.NoDataInCategories) {
        _SelectedModels =
            _ModelsProvider.SelectedModelsByCategoriesId(Convert.ToInt32(CategoriesCBox.SelectedValue));
        LoadData(_SelectedModels.ModelsFileModel);
    }
}

```

Рисунок 3.20 – Код події для зміни моделі

Для запуску тестування навчених моделей реалізовано метод, код якого представлено на рис. 3.21.

```

private void GenBtn_Click(object sender, EventArgs e) {
    if (IsCategoriesSelectedCorrect()) {
        if (timer1.Enabled) {
            timer1.Enabled = false;
            GenBtn.Text = "Генерувати";
        } else {
            timer1.Enabled = true;
            GenBtn.Text = "Зупинити";
        }
    }
}

```

Рисунок 3.21 – Код методу «RunBtn_Click»

Цей код описує функціонал кнопки "GenBtn", яка може використовуватися для управління процесом генерації чогось у графічному інтерфейсі користувача, наприклад, старту та зупинки таймеру, залежно від поточного стану.

Метод "GenBtn_Click" реагує на подію кліку на кнопку. Початково, він перевіряє, чи правильно вибрані категорії через виклик функції "IsCategoriesSelectedCorrect". Якщо функція повертає true, це означає, що умови для вибору категорій задовольняються, і код переходить до наступного блоку логіки.

У середині цієї перевірки, код аналізує, чи таймер (timer1) вже активований. Якщо таймер активований (timer1.Enabled дорівнює true), це означає, що процес або операція вже запущені, тому таймер вимикається, що призводить до зупинки процесу. Текст на кнопці змінюється на "Генерувати", що вказує на можливість повторного запуску процесу.

Якщо таймер не активований (тобто timer1.Enabled дорівнює false), це означає, що процес зупинено або не запущено. У цьому випадку, таймер активується, що запускає процес або операцію, а текст на кнопці змінюється на "Зупинити", сигналізуючи про те, що натискання кнопки наступного разу призведе до зупинки процесу.

Таким чином, код забезпечує управління процесом через кнопку з можливістю переключення між станами "запущено" та "зупинено", дозволяючи користувачу контролювати тривалі операції через графічний інтерфейс.

Для передбачення за допомогою моделі машинного навчання, і відображення результатів цього передбачення у графічному інтерфейсі

користувача розроблено подію таймеру «timer1_Tick», код якої представлено на рис. 3.22.

```
private void timer1_Tick(object sender, EventArgs e) {
    PredictAndDisplay();
}
```

Рисунок 3.22 – Код події «timer1_Tick»

Цей код визначає дію, яка виконується кожного разу, коли активізується подія «Tick» для таймера timer1. Функція «timer1_Tick» служить як обробник цієї події, зв'язуючи часовий інтервал таймера з викликом методу «PredictAndDisplay».

Інтервал таймера, на якому виконується «timer1_Tick», визначає, наскільки часто оновлюються дані або результати в інтерфейсі. Це може бути корисно в сценаріях, де потрібно постійно оновлювати дані, наприклад, при моніторингу в реальному часі або в інтерактивних застосунках, що вимагають частого оновлення інформації.

Метод PredictAndDisplay виконує роль в отриманні випадково згенерованих даних про університет, робить прогнозування їх світового рейтингу за допомогою вже навченої моделі, і відображає цю інформацію в текстовому полі користувачького інтерфейсу (рис. 3.23).

```
private void PredictAndDisplay() {
    // Вивід даних про освітній заклад
    var accountInfo = new StringBuilder();
    UniversityData randomUniversityData = UniversityDataGenerator.GenerateRandomData();
    accountInfo.AppendLine($"Якість освіти: {randomUniversityData.QualityOfEducation}");
    accountInfo.AppendLine($"Рівень зайнятості випускників: {randomUniversityData.AlumniEmployment}");
    accountInfo.AppendLine($"Якість викладацького складу: {randomUniversityData.QualityOfFaculty}");
    accountInfo.AppendLine($"Кількість публікацій: {randomUniversityData.Publications}");
    accountInfo.AppendLine($"Рівень впливу: {randomUniversityData.Influence}");
    accountInfo.AppendLine($"Кількість цитувань: {randomUniversityData.Citations}");
    accountInfo.AppendLine($"Широкий вплив: {randomUniversityData.BroadImpact}");
    accountInfo.AppendLine($"Кількість патентів: {randomUniversityData.Patents}\r\n");
    // прогноз за допомогою вже створеного predictionEngine, не передаючи Churn
    var prediction = predictionEngine.Predict(randomUniversityData);
    accountInfo.AppendLine($"Прогнозування загального балу (світовий рейтинг): {Math.Abs(prediction.Score)}");
    // Виводить результат у текстове поле
    ReportTextBox.Text = accountInfo.ToString();
}
```

Рисунок 3.23 – Код методу «PredictAndDisplay»

На початку методу ініціалізується об'єкт «StringBuilder», який використовується для збирання тексту, що потім буде виведено. Цей підхід дозволяє зручно додавати нові рядки тексту без створення надмірної кількості тимчасових рядків, оптимізуючи використання пам'яті і швидкість виконання.

Далі виконується генерація випадкових даних про університет через метод «GenerateRandomData» класу «UniversityDataGenerator». Отриманий об'єкт «randomUniversityData» містить різні характеристики, такі як якість освіти, рівень зайнятості випускників, якість викладацького складу, кількість публікацій, рівень впливу, кількість цитувань, широкий вплив та кількість патентів. Кожна з цих характеристик послідовно додається до «accountInfo» з відповідними описами.

Після цього використовується «predictionEngine» для виконання прогнозування на основі отриманих даних. Цей «predictionEngine», ймовірно, є налаштованим інструментом для використання певної моделі машинного навчання, що дозволяє здійснювати швидкі передбачення. Результат передбачення, «prediction.Score», представляє прогнозований світовий рейтинг університету, який також додається до «accountInfo».

В кінці методу зміст «accountInfo» перетворюється в рядок і встановлюється як текст текстового поля «ReportTBox» у користувацькому інтерфейсі, ефективно відображаючи зібрану інформацію та результати прогнозування користувачеві. Такий підхід дозволяє забезпечити інформативне і взаємодіюче відображення даних, що може бути особливо корисним у освітніх або аналітичних додатках.

Метод «PredictBtn_Click» відповідає за обробку події кліку на кнопку для виконання прогнозування. Він перевіряє коректність введених даних, виконує прогноз та відображає результат.

```

private void PredictBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        var result = predictionEngine.Predict(new UniversityData {
            QualityOfEducation = Convert.ToInt32(QualityOfEducationTBox.Text),
            AlumniEmployment = Convert.ToInt32(AlumniEmploymentTBox.Text),
            QualityOfFaculty = Convert.ToInt32(QualityOfFacultyTBox.Text),
            Publications = Convert.ToInt32(PublicationsTBox.Text),
            Influence = Convert.ToInt32(InfluenceTBox.Text),
            Citations = Convert.ToInt32(CitationsTBox.Text),
            BroadImpact = Convert.ToInt32(BroadImpactTBox.Text),
            Patents = Convert.ToInt32(PatentsTBox.Text)
        });
        var accountInfo = new StringBuilder();
        accountInfo.AppendLine($"Прогнозування загального балу (світовий рейтинг): " +
            $"{result.Score}");
        ReportTBox.Text = accountInfo.ToString();
    }
}

```

Рисунок 3.24 – Код методу «PredictBtn_Click»

Спочатку метод перевіряє, чи дані, введені користувачем, є коректними, викликаючи метод «IsDataEnteringCorrect». Якщо дані вважаються валідними, процес продовжується. Далі, використовуючи «predictionEngine», здійснюється прогнозування. Для цього створюється новий об'єкт «UniversityData», поля якого заповнюються з відповідних текстових полів у формі. Значення для кожного поля, такі як якість освіти, зайнятість випускників, якість викладацького складу, кількість публікацій, вплив, кількість цитувань, широкий вплив та кількість патентів, конвертуються з текстового формату в цілочисельний.

Отриманий результат прогнозування містить, серед іншого, оцінку загального балу (світового рейтингу), який зберігається в змінній «result». Цей бал або оцінка додається до «StringBuilder» об'єкту «accountInfo», який згодом використовується для формування тексту, який показує прогнозований світовий рейтинг.

У кінці, сформований у «accountInfo», перетворюється в рядок і встановлюється як текст у ReportTBox, що є текстовим полем на формі. Таким чином, користувачі можуть бачити результат прогнозування безпосередньо на інтерфейсі після натискання на кнопку прогнозування. Це забезпечує зручний та ефективний спосіб перегляду результатів аналітики у реальному часі.

3.5 Валідація функціональності компонентів системи

Функціональне тестування є критичним етапом у процесі розробки програмного забезпечення, основною метою якого є перевірка того, чи відповідають функції програми технічним специфікаціям. Цей етап забезпечує впевненість у тому, що кожен елемент системи працює відповідно до встановлених критеріїв. Під час функціонального тестування використовуються спеціально розроблені сценарії, які відтворюють стандартні ситуації використання системи. Ці сценарії включають чітко визначену послідовність кроків, очікувані результати та критерії оцінки продуктивності. Виконання цих тестів дозволяє своєчасно виявляти та усувати помилки в програмі, що сприяє підвищенню якості фінального продукту.

Тестовий сценарій №1: Навчання моделі

Кроки сценарію:

- користувач відкриває інтерфейс «Навчання моделей»;
- вибирає і завантажує CSV-файл даних через відповідну кнопку відкриття файлу;
- система імпортує дані і розділяє їх на навчальні та тестові набори;
- користувач активує процес тренування моделі натисканням на відповідну кнопку;
- система проводить тренування моделі і відображає показники якості у текстовому полі;
- користувач вказує назву моделі;
- вибирає категорію моделі зі спадного меню;
- зберігає модель за допомогою кнопки збереження.

Очікуваний результат:

- модель успішно збережена;
- на екран виводиться повідомлення про успішне збереження моделі;
- в інтерфейсі з'являються логи процесу навчання моделі.

Тестовий сценарій №2. Видалення існуючої моделі

Кроки сценарію:

- користувач заходить у форму "Навчання моделей";
- система показує перелік усіх доступних моделей у таблиці;
- користувач обирає модель для видалення;
- він натискає на кнопку "Видалити" біля вибраної моделі;
- з'являється діалогове вікно для підтвердження видалення;
- користувач підтверджує видалення моделі.

Очікуваний результат:

- обрана модель вилучається зі списку;
- таблиця автоматично оновлюється, і видалена модель більше не відображається.

Тестовий сценарій №3: Прогнозування загального балу навчального закладу

Кроки сценарію:

- користувач заходить у форму "Прогнозування";
- вибирає зі спадного списку категорію, на основі якої система визначає необхідну модель;
- активує процес симуляції натисканням на кнопку "Генерувати";
- система автоматично створює випадкові дані, проводить прогнозування та відображає результати в текстовому полі.

Очікуваний результат:

- активується таймер, який регулярно генерує нові дані і робить прогнози;
- результати прогнозування постійно оновлюються і відображаються у текстовому полі;
- результати прогнозування загального балу навчального закладу доступні для перегляду.

Тестовий сценарій №4: Зупинка симуляції

Кроки сценарію:

- користувач входить у форму "Прогнозування";
- вибирає потрібну категорію і ініціює симуляцію відповідно до інструкцій першого сценарію;

- натискає на кнопку "Зупинити" для припинення симуляції.

Очікуваний результат:

- таймер припиняє свою роботу;
- у текстовому полі припиняється оновлення даних і виведення нових прогнозів;

- текст кнопки змінюється на "Запустити".

Розроблені сценарії використані як база для тестування функціоналу розроблених форми системи.

Модульне тестування проведено для оцінки функціональності окремих компонентів програмного забезпечення. У рамках платформи .NET застосування тестового фреймворку «MSTest» виявилось ефективним, оскільки ця технологія має великий набір інструментів для розробки та виконання тестів. У файлі конфігурації «packages.config» вказано, що використовуються пакети «MSTest.TestAdapter» і «MSTest.TestFramework», які не тільки забезпечують інтеграцію з середовищем Visual Studio, але й дозволяють проводити тестування. Зокрема, фреймворк «MSTest» використовує спеціальні атрибути, такі як «[TestMethod]», для визначення методів, які потрібно включити в тестовий процес. Одна з основних переваг MSTest полягає в можливості автоматизувати модульне тестування, що є важливим для проведення регресійного аналізу з мінімальними зусиллями та витратами ресурсів. Така автоматизація демонструється на рис. 3.25.

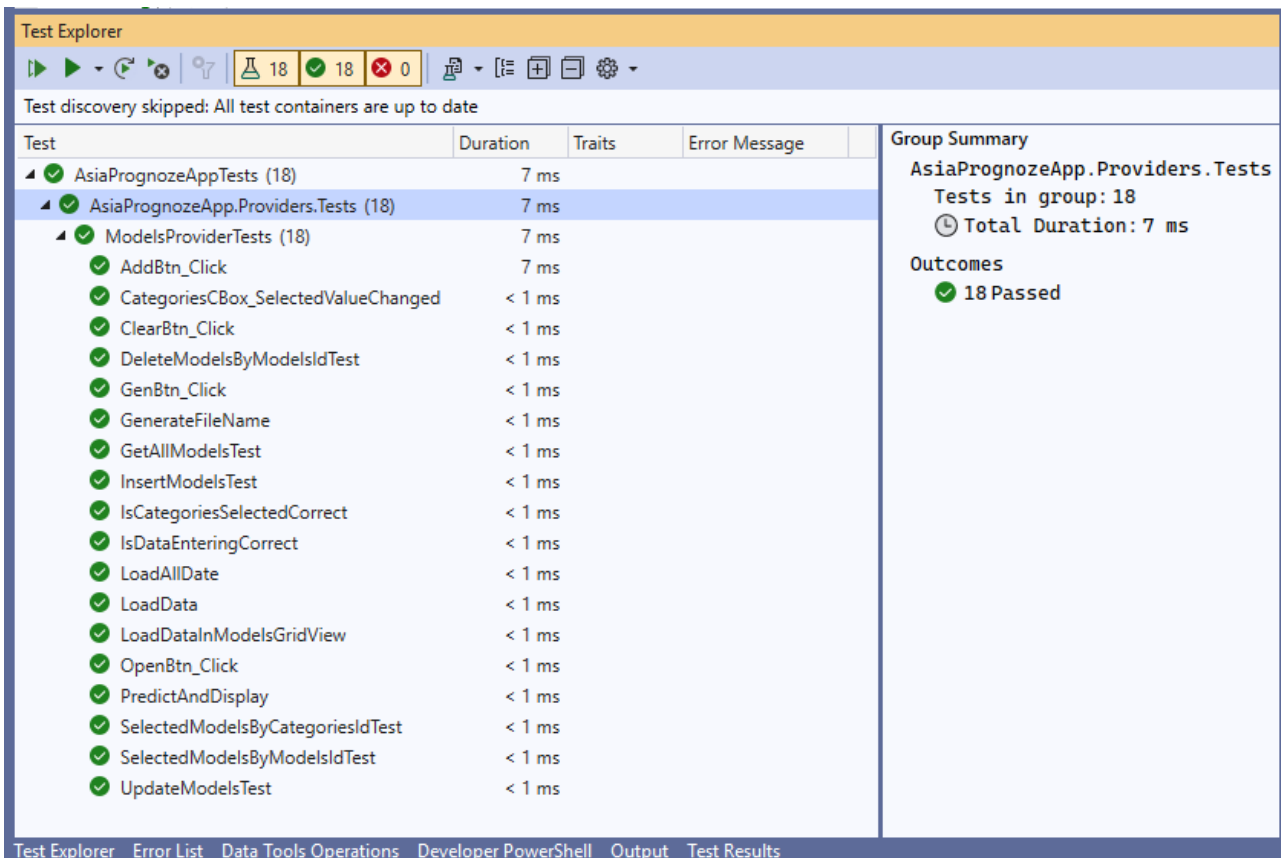


Рисунок 3.25 – Результати проведеного модульного тестування

Після проведення модульного тестування було підтверджено коректність роботи окремих компонентів програмного забезпечення. Результати демонструють відповідність очікуваним стандартам функціональності та якості коду, що сприяє забезпеченню надійності та ефективності додатку. Виявлені в процесі тестування дефекти були виправлені, що підвищило загальний ступінь стабільності системи.

3.6 Проведення навчання та вивчення результатів компонентного тестування моделі

У процесі розробки системи для прогнозування загального балу закладів вищої освіти країн Азії, було надано особливу увагу дослідницькій оцінці моделі машинного навчання. Для цього проводився аналіз її ефективності за допомогою експериментальних наборів даних, які відображали різні сценарії (рис. 3.26).

world_rank	institution	country	national_quality_of_alumni	er_quality_of_publication	influence	citations	broad_impacts	score	year				
55	Peking University	China	1	355	35	210	65	155	250	155	7	55,3	2020
56	Peking University	China	1	182	38	218	52	131	182	125	20	54,26	2021
78	Tsinghua University	China	2	309	73	218	63	158	65	156	30	52,21	2021
87	Tsinghua University	China	2	294	63	210	79	192	134	162	16	52,6	2020
180	Shanghai Jiao Tong University	China	3	335	143	218	81	267	212	209	198	47,96	2021
189	Fudan University	China	3	355	126	210	120	264	310	230	100	48,14	2020
191	Zhejiang University	China	4	367	309	218	71	290	368	265	106	47,66	2021
195	Shanghai Jiao Tong University	China	4	325	149	210	102	250	250	234	138	48,02	2020
195	Fudan University	China	5	367	203	218	112	252	368	204	123	47,56	2021
206	Zhejiang University	China	5	355	293	210	86	318	493	290	94	47,76	2020
217	Nanjing University	China	6	331	447	210	148	371	363	288	57	47,67	2020
239	University of Science and Technology of China	China	6	367	192	218	128	269	368	247	452	46,73	2021
242	Dalian University of Technology	China	7	355	67	210	326	666	406	472	74	47,22	2020
244	Nanjing University	China	7	342	362	218	134	358	234	265	164	46,61	2021
270	University of Science and Technology of China	China	8	355	238	210	144	275	363	266	481	46,75	2020
277	Sun Yat-sen University	China	8	367	254	218	161	393	287	285	221	46,25	2021
292	Sun Yat-sen University	China	9	355	328	210	198	424	310	313	138	46,42	2020
305	Dalian University of Technology	China	9	367	104	218	313	667	511	409	131	45,92	2021
313	Xiamen University	China	10	367	80	218	331	493	645	409	467	45,83	2021
315	Nankai University	China	11	334	135	218	282	454	321	334	419	45,8	2021

Рисунок 3.26 – Фрагмент вікна із даними для навчання моделі

Особливу увагу приділялося перевірці адаптивності та стійкості моделі до різних умов та атрибутів даних. Для цих цілей було використано датасет з ресурсу Kaggle [33], проведено попередню обробку даних, включаючи нормалізацію, заповнення пропущених значень та кодування категоріальних атрибутів.

Для тренування моделі на початку у програмі було створено категорію із назвою «Категорія закладів вищої освіти країн Азії» (рис. 3.27).

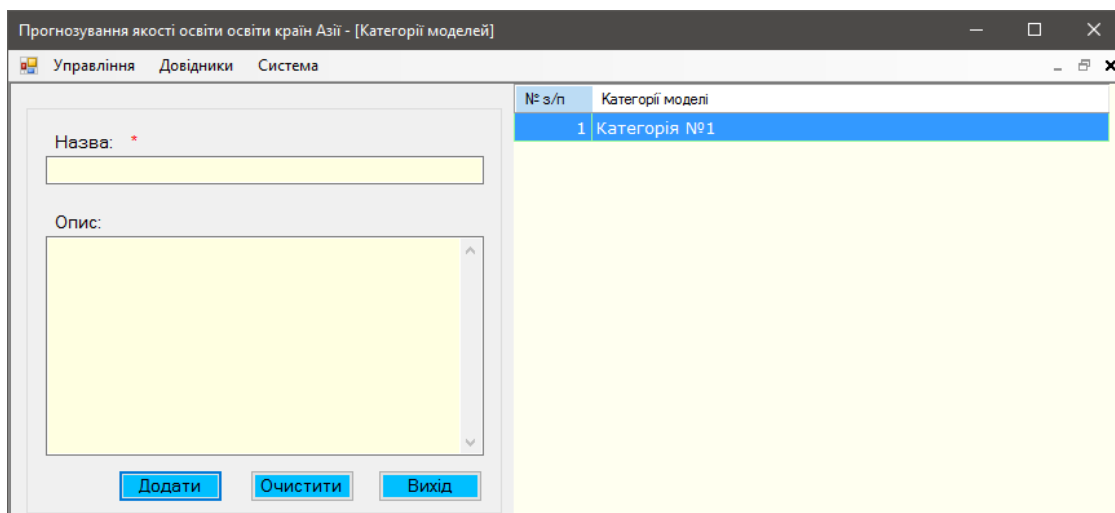


Рисунок 3.27– Створення категорії для навчання нейронної мережі

Для тренування нової моделі згідно завантаженого датасету, було здійснено перехід по меню програми «Довідники» → «Навчання моделей» та обрано категорію «Категорія закладів вищої освіти країн Азії». Після обрання підготовленого датасету у діалоговому вікні навчання моделі відбувається автоматично. Результат навчання показано на рис. 3.28.

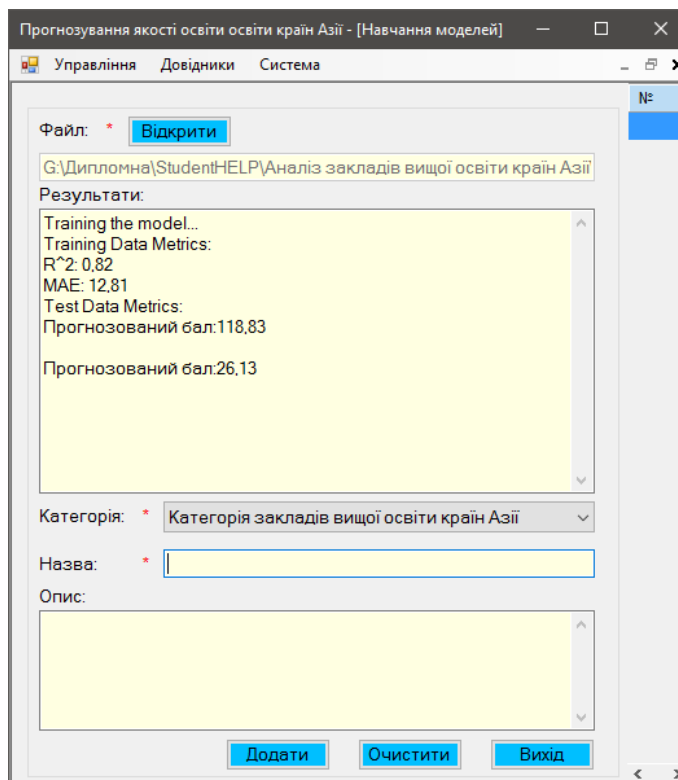


Рисунок 3.28 – Результат навчання моделі

Процес навчання НМ супроводжується виведенням повідомлення «Training the model...» та виведенням значень метрик:

– R-квадрат (R^2). Ця метрика вказує на частку варіації у залежній змінній, яку може пояснити модель. Значення R^2 знаходиться в діапазоні від 0 до 1, де 1 вказує на ідеальне узгодження моделі з даними, а 0 означає відсутність узгодження. Значення 0,82 свідчить про те, що модель здатна пояснити приблизно 82% варіації вихідної змінної (загальний бал закладів вищої освіти) за допомогою вхідних змінних, що є досить високим показником;

– середня абсолютна похибка (MAE). Ця метрика визначає середнє абсолютне значення різниці між прогнозованими і фактичними значеннями. Значення MAE дорівнює 12,81, що вказує на середню похибку прогнозування в 12,81 балів. Чим менше значення MAE, тим краще.

Після цього задано назву моделі та збережено її (рис. 3.29).

№	Назва моделі	Файл	Видалити
1	Модель №1	\teach\2024_3_22_21_47_13.zip	Видалити

Рисунок 3.29 – Збереження моделі

Для виявлення точності навченої моделі було згенеровано декілька потенційних сценаріїв за допомогою розробленого генератора сценаріїв та детально їх проаналізовано. На рис. 3.30 зображено перший згенерований сценарій.

Прогнозування якості освіти країн Азії - [Прогнозування]

Управління Довідники Система

Інформація для прогнозування:

Категорія: Категорія закладів вищої освіти країн Азії

Якість освіти: 200

Рівень зайнятості випускників: 150

Якість викладацького складу: 100

Кількість публікацій: 300

Рівень впливу: 250

Кількість цитувань: 180

Широкий вплив: 260

Кількість патентів: 100

Прогнозувати Генерувати

Якість освіти: 138
 Рівень зайнятості випускників: 109
 Якість викладацького складу: 352
 Кількість публікацій: 364
 Рівень впливу: 106
 Кількість цитувань: 489
 Широкий вплив: 417
 Кількість патентів: 263

Прогнозування загального балу (світовий рейтинг): 86,45641

Рисунок 3.30 – Виведення результатів прогнозування 1-го сценарію

Аналізуючи результати прогнозування загального балу навчального закладу за першим сценарієм, можна визначити кілька ключових аспектів, які демонструють високу ефективність використаної моделі:

- якість освіти, яка досягає показника 138, і рівень зайнятості випускників з оцінкою 109, свідчать про достатній рівень підготовки студентів;
- високий рівень якості викладацького складу (352) та значна кількість публікацій (364) засвідчують активну наукову діяльність установи;
- показники рівня впливу (106), кількості цитувань (489), широкого впливу (417), і кількості патентів (263) підтверджують значний внесок університету в академічну та прикладну сфери;
- загальний бал навчального закладу, прогнозований як 86,45641, розкриває здатність моделі точно оцінювати потенціал університету в глобальному освітньому просторі.

Отриманий результат не підтверджує точність використаної аналітичної моделі в прогнозуванні результатів та демонструє її здатність адекватно враховувати різні аспекти діяльності навчального закладу. Врахування такого широкого спектру показників гарантує, що модель може бути використана для об'єктивної оцінки та порівняння навчальних закладів в рамках освітньої аналітики.

На рис. 3.31 представлено результат прогнозування моделі 2-го сценарію.

Прогнозування якості освіти країн Азії - [Прогнозування]

Управління Довідники Система

Інформація для прогнозування:

Категорія: Категорія закладів вищої освіти країн Азії

Якість освіти: 200

Рівень зайнятості випускників: 150

Якість викладацького складу: 100

Кількість публікацій: 300

Рівень впливу: 250

Кількість цитувань: 180

Широкий вплив: 260

Кількість патентів: 100

Прогнозувати Генерувати

Якість освіти: 438
 Рівень зайнятості випускників: 112
 Якість викладацького складу: 184
 Кількість публікацій: 108
 Рівень впливу: 113
 Кількість цитувань: 466
 Широкий вплив: 276
 Кількість патентів: 395

Прогнозування загального балу (світовий рейтинг): 283,7141

Рисунок 3.31 – Виведення результатів прогнозування 2-го сценарію

Аналізуючи результати прогнозування загального балу навчального закладу за другим сценарієм, можна виявити кілька важливих аспектів, що відзначають високу ефективність моделі у вимірюванні освітніх параметрів:

- висока оцінка якості освіти, що становить 430, вказує на відмінні умови для навчання та розвитку студентів;

- рівень зайнятості випускників з показником 112 свідчить про добрі можливості працевлаштування після закінчення університету, що є важливим показником для оцінки успішності освітньої програми;

- якість викладацького складу та кількість публікацій з оцінками відповідно 184 та 108, хоча й не є найвищими, але підтримують стійкий академічний рівень;

- рівень впливу, кількість цитувань, широкий вплив та кількість патентів із оцінками 113, 466, 276 та 395 відповідно демонструють сильне представництво університету в наукових дослідженнях та інноваціях, що засвідчує його вплив на глобальний освітній та дослідницький контекст.

Загальний прогнозований бал у світовому рейтингу, що становить 283,7141, відображає високу загальну оцінку навчального закладу, яка базується на аналізі його академічної діяльності, дослідницької продуктивності та впливу на освітній сектор. Це підтверджує точність моделі у визначенні компетентності та репутації університету, що дозволяє забезпечити об'єктивне та вичерпне рейтингування освітніх інституцій.

3.7 Кореляційний аналіз отриманих даних

Для оцінки взаємозв'язків між різними показниками університетів Азії було проведено кореляційний аналіз. Використані дані включають ключові метрики якості освіти, такі як якість освіти, зайнятість випускників, якість

факультету, кількість публікацій, вплив, кількість цитувань, патенти та загальний бал.

Для аналізу були використані дані університетів з наступних азіатських країн: Китай, Кіпр, Гонконг, Індія, Іран, Ізраїль, Японія, Малайзія, Саудівська Аравія, Південна Корея, Тайвань, Таїланд та Туреччина. Дані були завантажені з файлу *asian_uni.csv* та відфільтровані відповідно до переліку країн. Табл. 3.5 нижче відображає перші п'ять рядків підготовлених даних.

Таблиця 3.4 – Опис атрибутів сутності «Logs» (Логи подій)

№	world_rank	institution	country	national_rank	quality_of_education	alumni_employment	quality_of_faculty	publications	influence	citations	broad_impact	patents	score	year
0	55	Peking University	China	1	355	35	210	65	155	250	155.0	7	55.30	2020
1	56	Peking University	China	1	182	38	218	52	131	182	125.0	20	54.26	2021
2	78	Tsinghua University	China	2	309	73	218	63	158	65	156.0	30	52.21	2021
3	87	Tsinghua University	China	2	294	63	210	79	192	134	162.0	16	52.60	2020
4	180	Shanghai Jiao Tong University	China	3	335	143	218	81	267	212	209.0	198	47.96	2021

В результаті підготовки даних було отримано 167 записів з наступними колонками: ранг університету у світовому рейтингу, назва університету, країна, в якій розташований університет, національний ранг університету, якість освіти, зайнятість випускників, якість факультету, кількість публікацій, вплив, кількість цитувань, широкий вплив, кількість патентів, загальний бал, рік.

Для виявлення розподілу університетів за країнами було побудовано горизонтальну стовпчасту діаграму (рис. 3.33). На діаграмі видно, що найбільше

університетів представлено з Китаю та Японії, тоді як інші країни мають менше представлених навчальних закладів.

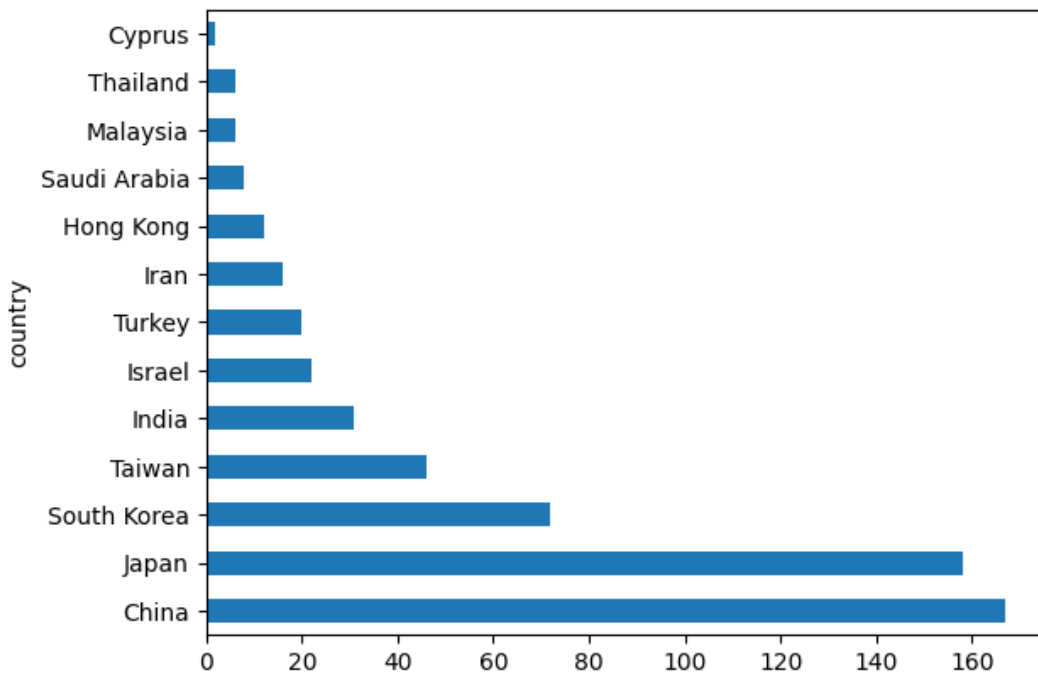


Рисунок 3.33 – Розподіл університетів за країнами

Після цього було проведено аналіз середніх балів університетів за країнами (Для цього дані були згруповані за країнами, і було розраховано середній бал для кожної країни. Це дозволяє оцінити загальний рівень якості освіти в різних країнах Азії.

Також було створено більш детальну діаграму, яка відображає середні бали університетів у різних країнах Азії (рис. 3.35). Для цього було використано горизонтальну стовпчасту діаграму, де кожен стовпчик показує середній бал для відповідної країни.

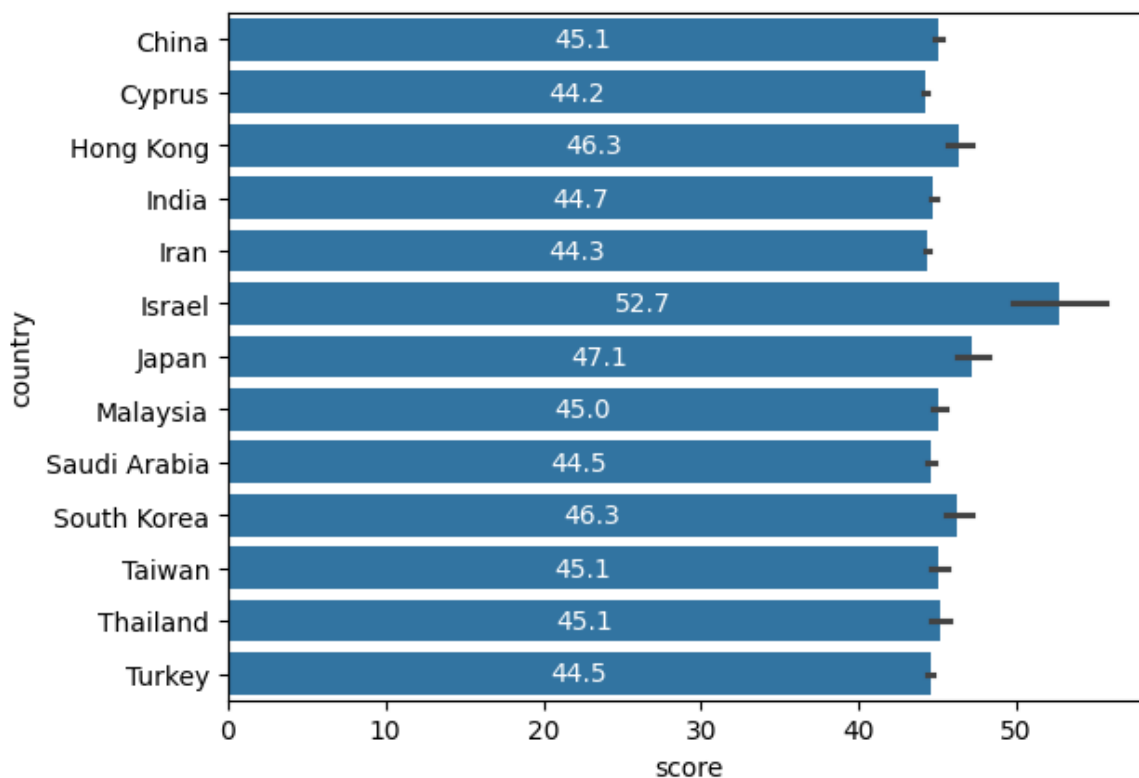


Рисунок 3.35 – Середні бали університетів у різних країнах Азії

На графіку видно, що середній бал університетів Ізраїлю значно вищий порівняно з іншими країнами, досягаючи значення 52.7. Це свідчить про високий рівень якості освіти в цій країні. Японія займає друге місце із середнім балом 47.1, що також вказує на високий рівень університетів у цій країні.

Країни, такі як Гонконг та Південна Корея, мають середній бал 46.3, що вказує на стабільно високий рівень якості освіти. Тайвань і Таїланд мають однаковий середній бал 45.1, що також свідчить про хороший рівень освіти.

Китай, Малайзія, Саудівська Аравія та Туреччина мають середні бали в діапазоні від 44.5 до 45.1, що вказує на середній рівень якості освіти. Кіпр і Індія мають дещо нижчі середні бали 44.2 і 44.7 відповідно, а Іран має середній бал 44.3.

Наступним кроком створено діаграму розмаху (boxplot), яка надає більш детальну інформацію про розподіл балів університетів у різних країнах Азії (рис. 3.36). Ця діаграма дозволяє візуалізувати не лише середні значення, але й діапазон балів, а також виявити можливі аномальні значення (викиди).

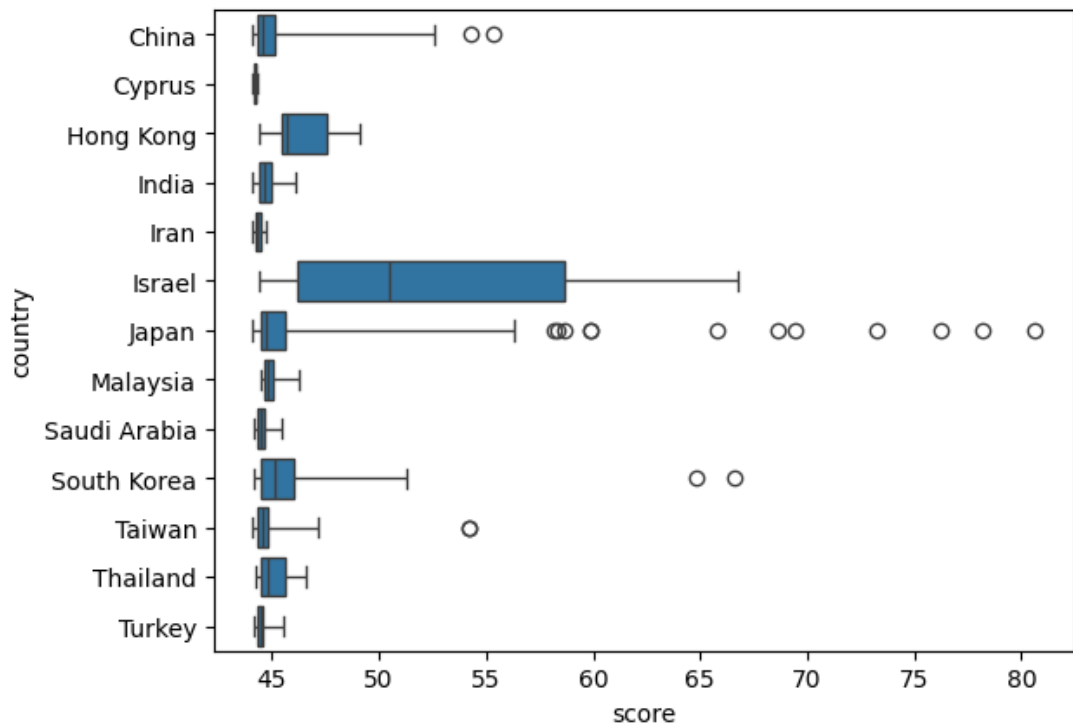


Рисунок 3.36 – Діаграма розмаху

На діаграмі можна побачити, що Ізраїль має найбільший діапазон балів серед університетів, що свідчить про значну варіативність якості освіти в університетах цієї країни. Середній бал університетів Ізраїлю також є найвищим, що підтверджує висновки, зроблені на основі попередніх графіків. Викиди, що виходять за межі діаграми розмаху, вказують на наявність університетів із значно вищими або нижчими балами порівняно з іншими.

Японія також має широкий діапазон балів, проте середні значення показують стабільно високий рівень якості освіти. На діаграмі можна побачити кілька викидів, що свідчить про наявність університетів з дуже високими балами, які значно перевищують середні показники.

Країни, такі як Гонконг, Південна Корея, Тайвань та Таїланд, мають відносно вузький діапазон балів, що свідчить про стабільність у якості освіти. Викиди для цих країн вказують на наявність кількох університетів з балами, що значно відрізняються від середніх значень.

Китай, Індія, Малайзія, Саудівська Аравія та Туреччина мають дещо ширший діапазон балів, що свідчить про більшу варіативність у якості освіти в

цих країнах. Викиди для цих країн також вказують на наявність університетів з дуже високими або низькими балами.

Іран має найбільший діапазон балів з усіх країн, представлених на діаграмі. Це свідчить про значну варіативність якості освіти в університетах цієї країни, що може бути обумовлено різними факторами, такими як рівень фінансування, кваліфікація викладачів та інші.

Діаграма надає глибше розуміння розподілу балів університетів у різних країнах Азії та допомагає виявити можливі аномалії та викиди. Це важливо для подальшого аналізу та прийняття рішень у сфері освіти, оскільки дозволяє виявити не лише середні значення, але й варіативність та стабільність якості освіти в різних країнах.

Така інформація є важливою для оцінки загальних тенденцій у якості освіти в різних країнах Азії. Вона дозволяє ідентифікувати країни з найвищими та найнижчими середніми балами, що може бути корисним для подальшого аналізу та прийняття рішень у сфері освіти.

Також було створено графік щільності розподілу балів університетів за допомогою ядерного оцінювання щільності (рис. 3.37). Цей метод дозволяє візуалізувати розподіл балів серед університетів та ідентифікувати області з високою щільністю значень.

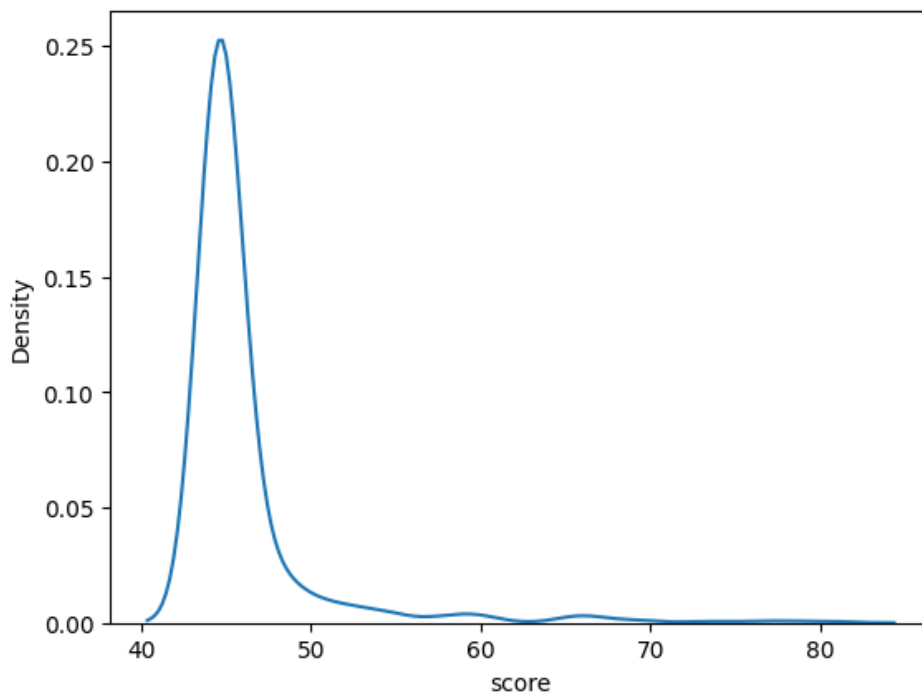


Рисунок 3.37 – Графік щільності

На графіку видно, що більшість університетів мають бали в діапазоні від 40 до 50. Це вказує на те, що основна маса університетів у вибірці має середній рівень якості освіти. Пік щільності розташований приблизно на рівні 45, що є модальним значенням для цієї вибірки.

Після різкого спаду щільності від 50 до 60 балів, графік показує незначне зростання щільності на рівнях від 60 до 70 балів. Це вказує на наявність невеликої кількості університетів з вищими балами, які значно перевищують середні значення. Проте щільність цих значень значно нижча порівняно з основною масою університетів.

Високий пік на рівні 45 балів та різкий спад щільності після цього значення вказують на те, що більшість університетів мають схожий рівень якості освіти, з деякими винятками, що мають значно вищі або нижчі бали. Це може бути обумовлено різними факторами, такими як рівень фінансування, кваліфікація викладачів, інфраструктура та інші.

Цей графік є важливим інструментом для розуміння загальних тенденцій у розподілі балів університетів. Він дозволяє виявити, де саме знаходиться основна маса університетів за рівнем якості освіти, а також виявити аномалії та викиди.

Наступним кроком буде детальний аналіз кореляцій між різними показниками якості університетів, щоб виявити взаємозв'язки та фактори, що впливають на загальний бал.

Після цього створено графік скрипкової діаграми (violin plot), яка відображає розподіл балів університетів Китаю. Цей тип діаграми поєднує в собі властивості діаграми розмаху (boxplot) та ядерного оцінювання щільності (KDE), що дозволяє візуалізувати як розподіл даних, так і їх щільність (рис. 3.38).

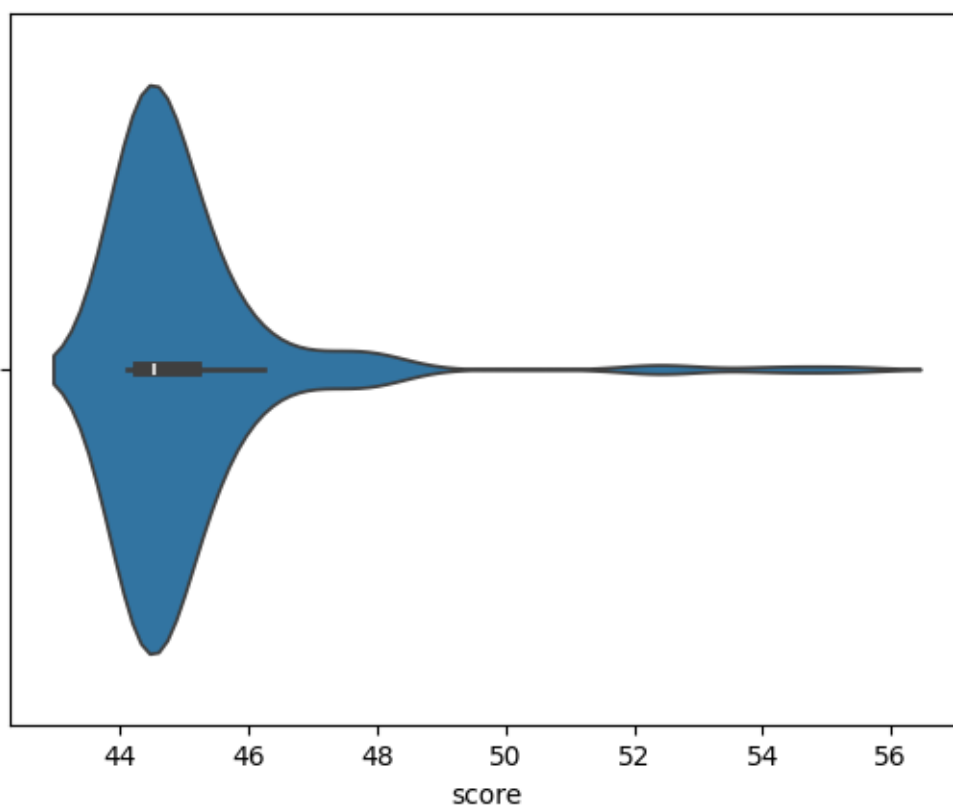


Рисунок 3.38 – Графік скрипкової діаграми

На графіку видно, що основна маса університетів Китаю має бали в діапазоні від 44 до 46, причому модальне значення знаходиться приблизно на рівні 45. Це вказує на те, що більшість університетів Китаю мають середній рівень якості освіти, схожий на той, що було виявлено для загальної вибірки університетів Азії.

Широка частина діаграми в діапазоні від 44 до 46 балів свідчить про високу щільність університетів з цими значеннями. Це означає, що більшість університетів мають бали, які незначно відрізняються від модального значення, що свідчить про однорідність якості освіти в цій країні.

Діаграма також показує наявність невеликої кількості університетів з балами, що виходять за межі основного діапазону, досягаючи до 56 балів. Ці значення представлені тонкими лініями на краях діаграми та вказують на наявність університетів з високими балами, які значно перевищують середні показники.

Цей графік є важливим інструментом для розуміння розподілу балів університетів Китаю. Він дозволяє виявити, де саме знаходиться основна маса університетів за рівнем якості освіти, а також виявити аномалії та викиди. Діаграма надає глибше розуміння варіативності та стабільності якості освіти в Китаї, що є важливим для подальшого аналізу та прийняття рішень у сфері освіти.

Після цього створено графіки щільності розподілу балів університетів для основних країн Азії (рис. 3.39).

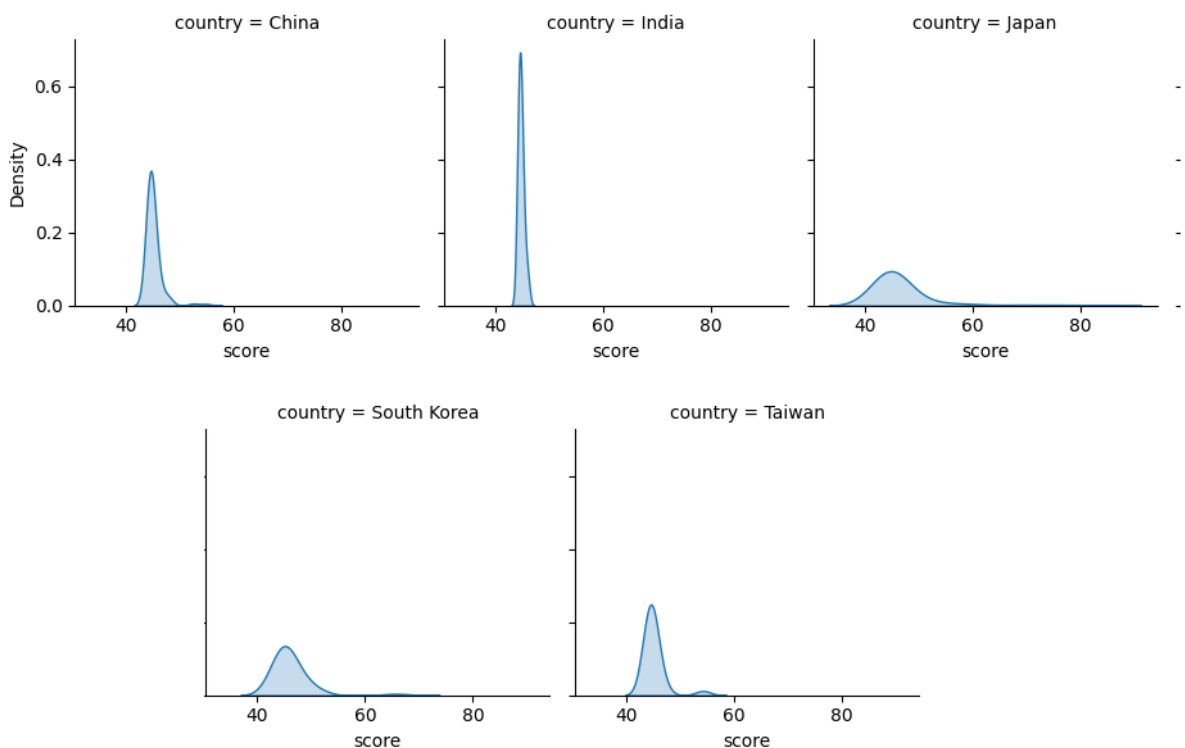


Рисунок 3.39 – Щільність розподілу балів університетів для основних країн Азії

Графіки побудовано з використанням методу ядерного оцінювання щільності (KDE), що дозволяє візуалізувати розподіл балів для кожної окремої країни. На графіках показано щільність розподілу балів для університетів Китаю, Індії, Японії, Південної Кореї та Тайваню.

Для Китаю основна маса університетів має бали в діапазоні від 40 до 45. Це вказує на середній рівень якості освіти в країні. Щільність значно знижується після цього діапазону, що свідчить про меншу кількість університетів з вищими балами. В Індії щільність розподілу має високий пік у діапазоні від 40 до 42, що свідчить про концентрацію університетів із середніми значеннями балів. Щільність швидко знижується після цього піку, вказуючи на малу кількість університетів з балами, що перевищують середні значення.

Японія має більш рівномірний розподіл балів з невеликим піком у діапазоні від 40 до 45, що вказує на більшу варіативність якості освіти в університетах цієї країни. Діапазон балів також значно ширший порівняно з іншими країнами. У Південній Кореї спостерігається схожий на Японію розподіл, з основною масою університетів у діапазоні від 40 до 45. Щільність балів знижується після цього діапазону, вказуючи на меншу кількість університетів з вищими балами.

Тайвань демонструє високий пік у діапазоні від 40 до 45, схожий на Китай та Індію. Це свідчить про концентрацію університетів із середніми значеннями балів. Після цього піку щільність значно знижується, вказуючи на меншу кількість університетів з вищими балами.

Ці графіки дозволяють порівняти розподіл балів університетів у різних країнах Азії. Вони показують, що, незважаючи на загальну тенденцію до середніх значень, існують відмінності в розподілі та варіативності балів між країнами. Ця інформація важлива для розуміння якості освіти в кожній країні та може бути використана для подальшого аналізу та прийняття рішень у сфері освіти.

Також було створено серію діаграм розсіювання (scatter plots), які дозволяють оцінити взаємозв'язок між різними показниками якості університетів та загальним балом для університетів різних країн Азії (рис. 3.40). Кожна діаграма показує зв'язок між загальним балом та одним із наступних показників: якість освіти, зайнятість випускників, якість факультету, кількість публікацій, вплив, кількість цитувань, широкий вплив, кількість патентів.

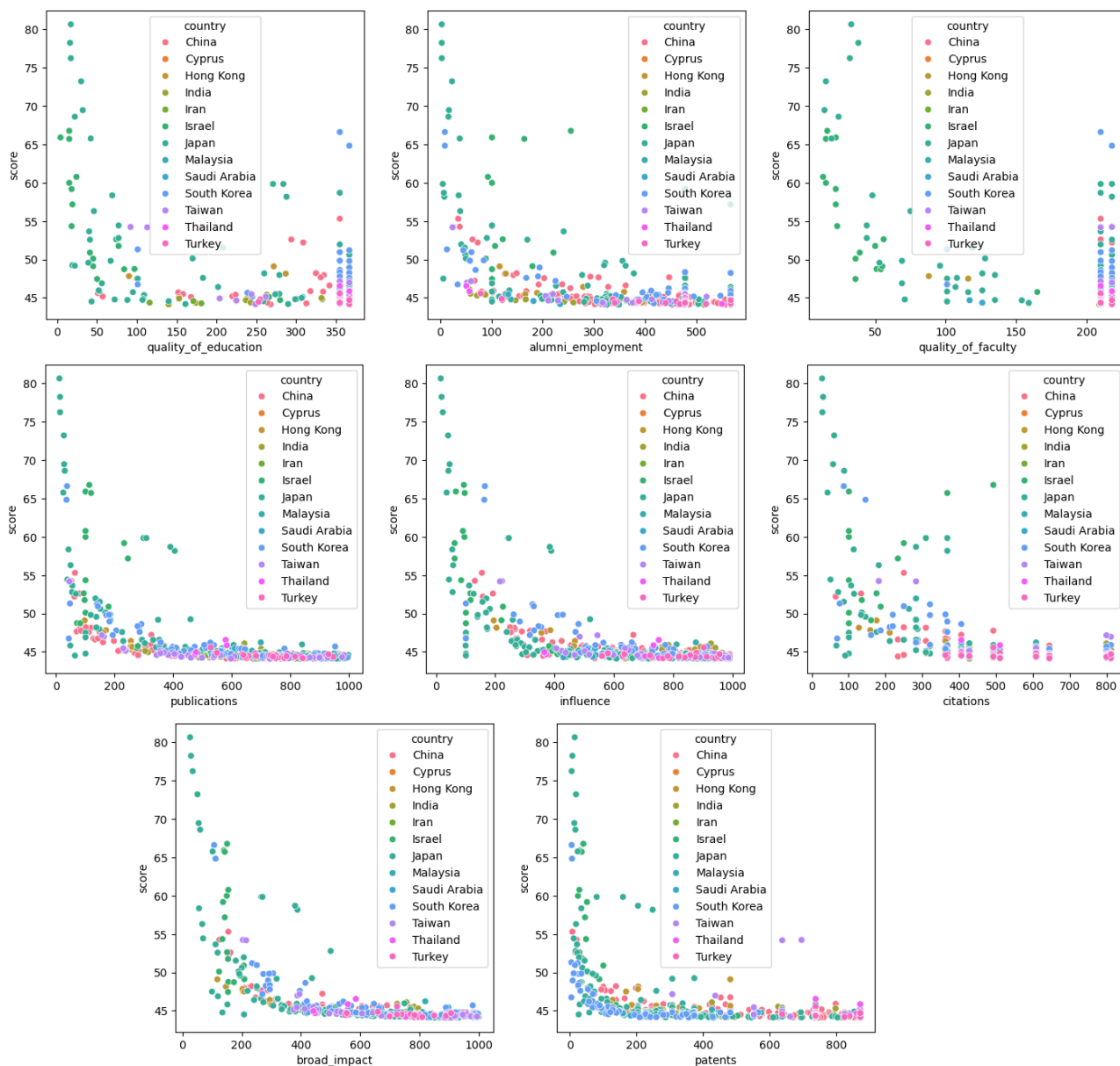


Рисунок 3.40 – Діаграми розсіювання

Діаграми показують, що для більшості показників спостерігається негативний кореляційний зв'язок з загальним балом. Це означає, що зі збільшенням значень таких показників, як якість освіти, зайнятість випускників, якість факультету, кількість публікацій, вплив, кількість цитувань, широкий вплив та кількість патентів, загальний бал знижується.

На діаграмі для якості освіти видно, що країни, такі як Китай, Японія та Південна Корея, мають високу щільність університетів із середніми значеннями якості освіти та загальними балами в діапазоні від 40 до 50. Для зайнятості випускників також спостерігається схожа тенденція, де більшість університетів мають низькі значення зайнятості випускників та середні загальні бали.

Діаграма для якості факультету показує, що університети з вищими значеннями якості факультету мають нижчі загальні бали. Це може бути пов'язано з тим, що якість факультету є лише одним із багатьох факторів, що впливають на загальний бал.

На діаграмах для кількості публікацій, впливу, кількості цитувань, широкого впливу та кількості патентів також спостерігається негативний зв'язок з загальним балом. Це може свідчити про те, що ці показники не завжди відображають загальну якість освіти, яка враховується при розрахунку загального балу.

Загалом, ці діаграми дозволяють візуалізувати взаємозв'язок між різними показниками якості університетів та їх загальними балами. Вони показують, що, незважаючи на значення окремих показників, загальний бал залежить від комбінації багатьох факторів. Це важливо для розуміння загальних тенденцій у якості освіти в різних країнах Азії та може бути використано для подальшого аналізу та прийняття рішень у сфері освіти.

Також створено графік ядерного оцінювання щільності (KDE), який відображає взаємозв'язок між кількістю публікацій та широким впливом (broad impact) університетів Китаю (рис. 3.41). Цей графік дозволяє візуалізувати розподіл значень та їх щільність, що допомагає ідентифікувати області з високою концентрацією даних.

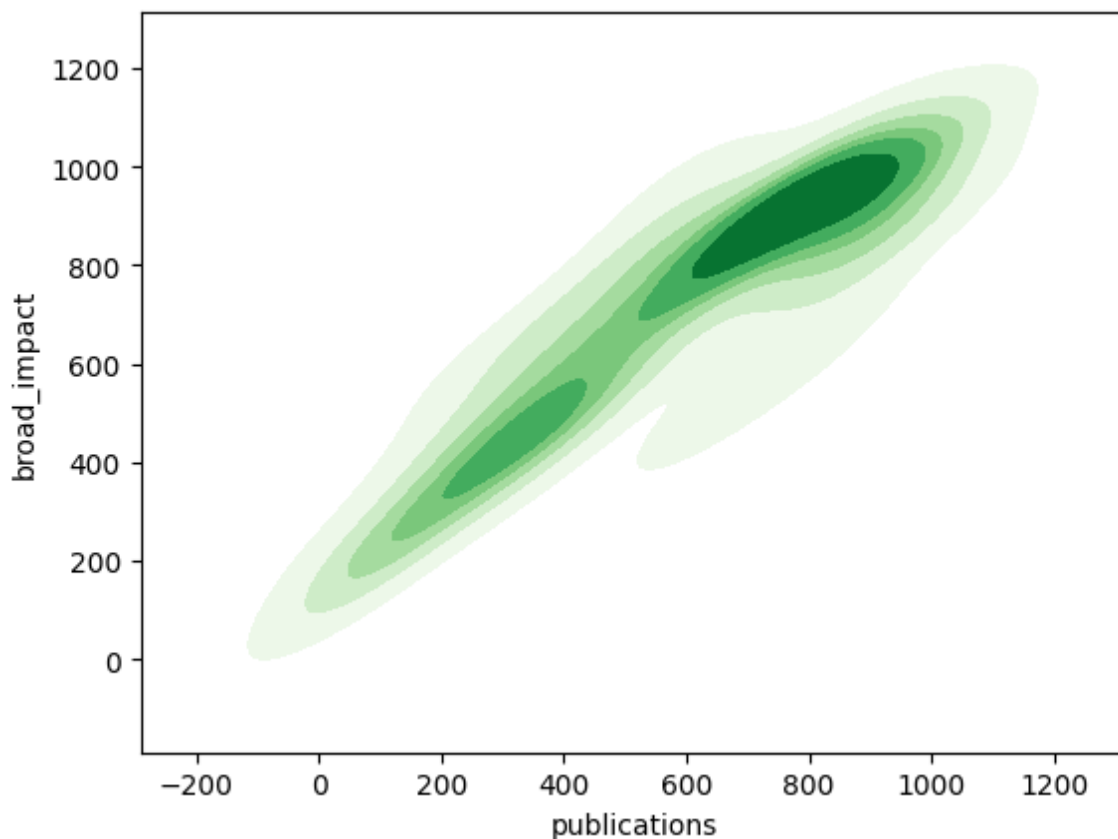


Рисунок 3.41 – Графік ядерного оцінювання щільності

На графіку видно, що існує позитивний кореляційний зв'язок між кількістю публікацій та широким впливом університетів. Це означає, що зі збільшенням кількості публікацій також збільшується широкий вплив. Висока щільність спостерігається в діапазоні від 400 до 800 публікацій та від 600 до 1000 одиниць широкого впливу. Це вказує на те, що більшість університетів Китаю мають середні значення для цих показників.

Області з високою щільністю відображені більш насиченими відтінками зеленого кольору, що свідчить про концентрацію університетів із схожими значеннями. Області з меншою щільністю, відображені світлішими відтінками зеленого, вказують на меншу кількість університетів з такими значеннями.

Цей графік є важливим інструментом для розуміння взаємозв'язку між кількістю публікацій та широким впливом університетів Китаю. Він показує, що університети з більшою кількістю публікацій мають також більший широкий вплив, що може бути обумовлено високою якістю досліджень та наукових робіт. Це важливо для подальшого аналізу та прийняття рішень у сфері освіти, оскільки

дозволяє ідентифікувати ключові фактори, що впливають на якість та вплив університетів.

Після цього було створено кореляційну матрицю для показників якості університетів Китаю, включаючи загальний бал (рис. 3.42). Кореляційна матриця показує взаємозв'язки між різними показниками, що дозволяє оцінити, наскільки один показник впливає на інший.

	quality_of_education	alumni_employment	quality_of_faculty	publications	influence	citations	broad_impact	patents	score
quality_of_education	1.000	0.120	0.095	0.248	0.301	0.226	0.243	0.171	-0.243
alumni_employment	0.120	1.000	0.083	0.562	0.416	0.415	0.503	0.558	-0.590
quality_of_faculty	0.095	0.083	1.000	-0.073	-0.024	-0.066	-0.076	0.159	-0.035
publications	0.248	0.562	-0.073	1.000	0.788	0.587	0.897	0.635	-0.644
influence	0.301	0.416	-0.024	0.788	1.000	0.617	0.774	0.489	-0.702
citations	0.226	0.415	-0.066	0.587	0.617	1.000	0.725	0.441	-0.637
broad_impact	0.243	0.503	-0.076	0.897	0.774	0.725	1.000	0.565	-0.693
patents	0.171	0.558	0.159	0.635	0.489	0.441	0.565	1.000	-0.588
score	-0.243	-0.590	-0.035	-0.644	-0.702	-0.637	-0.693	-0.588	1.000

Рисунок 3.42 – Кореляційна матриця показників якості університетів Китаю

На кореляційній матриці видно, що між різними показниками існують як позитивні, так і негативні кореляційні зв'язки. Найсильніший позитивний кореляційний зв'язок спостерігається між кількістю публікацій та широким впливом (коефіцієнт кореляції 0.897). Це означає, що зі збільшенням кількості публікацій також значно зростає широкий вплив університетів.

Інші важливі позитивні кореляційні зв'язки включають взаємозв'язок між кількістю публікацій та кількістю патентів (0.635), між впливом та широким впливом (0.774), а також між кількістю цитувань та широким впливом (0.725). Ці зв'язки свідчать про те, що збільшення одного показника часто супроводжується збільшенням інших пов'язаних показників, що підвищує загальну якість та вплив університетів.

Що стосується загального балу, то найсильніший негативний кореляційний зв'язок спостерігається з кількістю публікацій (-0.644), впливом (-0.702), кількістю цитувань (-0.637), широким впливом (-0.693) та кількістю патентів (-0.588). Це означає, що зі збільшенням цих показників загальний бал знижується. Це може бути обумовлено тим, що загальний бал враховує

комбінацію багатьох факторів, і високі значення окремих показників не завжди відображають загальну якість освіти.

Кореляційна матриця також показує, що деякі показники мають слабкий або відсутній кореляційний зв'язок із загальним балом, наприклад, якість освіти (-0.243) та якість факультету (-0.035). Це свідчить про те, що ці показники мають менший вплив на загальний бал у порівнянні з іншими показниками.

На останок було створено діаграму розсіювання, яка відображає розподіл університетів Азії за країнами на двовимірному просторі (рис. 3.43). Кожна точка на графіку представляє університет, а кольори та стилі маркерів використовуються для розрізнення країн. Ця діаграма дозволяє візуалізувати кластеризацію університетів залежно від країни.

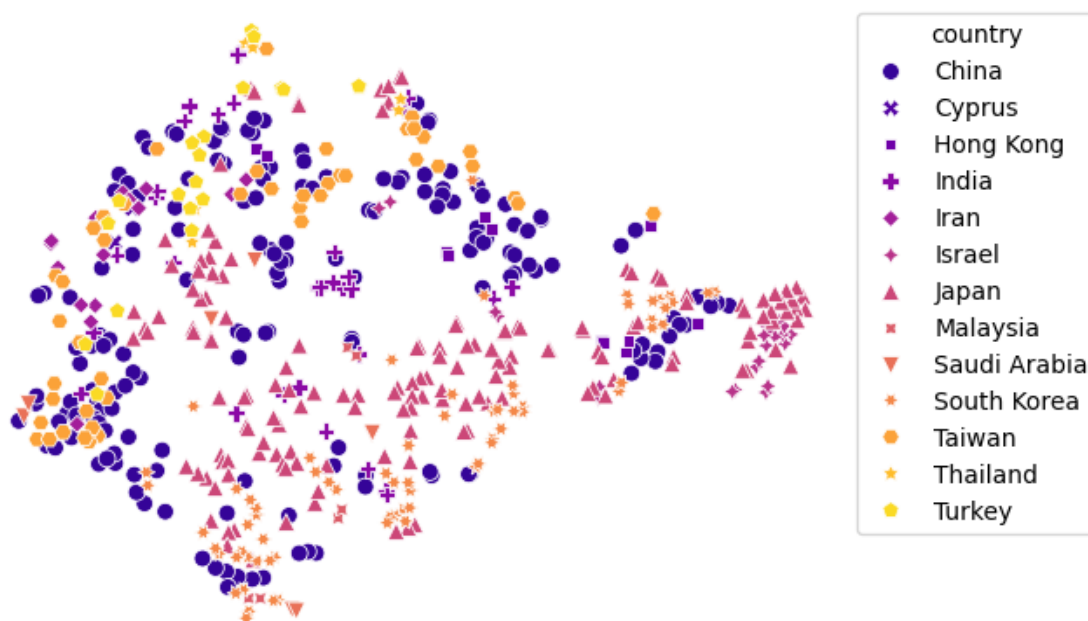


Рисунок 3.43 – Розподіл університетів за країнами на 2-ох вимірному просторі

На графіку видно, що університети з різних країн утворюють окремі кластери. Це означає, що університети всередині кожного кластеру мають схожі характеристики або показники якості, тоді як між кластерами спостерігається певна відмінність.

Кластери для Китаю, Японії та Південної Кореї розташовані в різних частинах графіку, що вказує на те, що університети цих країн мають унікальні особливості. Китай має найбільшу кількість університетів, що утворюють

щільний кластер, вказуючи на однорідність показників якості освіти всередині країни.

Університети Індії, Ізраїлю, Саудівської Аравії та Тайваню також утворюють окремі кластери, що дозволяє ідентифікувати характерні особливості цих країн. Наприклад, університети Ізраїлю утворюють компактний кластер, що може свідчити про високу якість освіти та меншу варіативність показників.

На діаграмі також видно, що університети країн, таких як Кіпр, Гонконг, Іран, Малайзія, Таїланд та Туреччина, розташовані на різних частинах графіку, утворюючи менші кластери. Це свідчить про більшу варіативність показників якості освіти всередині цих країн.

У ході проведеного аналізу було досліджено взаємозв'язки між різними показниками якості вищих навчальних закладів країн Азії. Використано дані університетів з таких країн, як Китай, Японія, Південна Корея, Тайвань та інших. Було проведено кореляційний аналіз основних показників, включаючи якість освіти, зайнятість випускників, якість факультету, кількість публікацій, вплив, кількість цитувань, широкий вплив та кількість патентів. Результати показали наявність як позитивних, так і негативних кореляцій між різними показниками та загальним балом університетів. Найсильніші позитивні кореляції спостерігалися між кількістю публікацій та широким впливом, що свідчить про взаємозв'язок між науковою активністю та загальним впливом університету. Також було виявлено, що деякі показники мають слабкий або відсутній вплив на загальний бал.

Проведений аналіз показав, що університети різних країн утворюють окремі кластери, що дозволяє ідентифікувати характерні особливості кожної країни. Університети Китаю, Японії та Південної Кореї демонструють високу однорідність показників якості освіти, тоді як інші країни мають більшу варіативність. Візуалізація даних за допомогою графіків щільності розподілу, скрипкових діаграм, діаграм розсіювання та кореляційної матриці надала можливість детально дослідити розподіл та взаємозв'язки між показниками якості університетів.

Загалом, проведений аналіз дозволив глибше зрозуміти якість освіти в університетах країн Азії та виявити ключові фактори, що впливають на загальний бал. Це може бути використано для подальшого аналізу та прийняття рішень у сфері освіти, а також для визначення сильних та слабких сторін університетів у різних країнах.

3.8 Оцінка отриманих результатів

Робота, спрямована на дослідження університетів Азії, зосереджувалася на оцінці та порівнянні освітніх закладів за допомогою алгоритму стохастичного двокоординатного сходження. Цей метод аналізу дозволяв розглянути широкий спектр характеристик освітніх інституцій, починаючи від академічних досягнень до впливу наукових досліджень. Застосування алгоритму сприяло високоточному моделюванню даних, що у свою чергу дозволило ефективно прогнозувати глобальні рейтинги цих установ. Це дало можливість не тільки об'єктивно оцінити поточний статус, але й виявити ключові тенденції та напрямки розвитку освітньої сфери в регіоні.

Оцінка отриманих результатів включала декілька критичних етапів:

- збір та підготовка даних. Для тренування моделі були використано тренувальні дані завантажені із сайту Kaggle. У файлі знаходяться такі характеристики університетів, як якість освіти, рівень зайнятості випускників, кількість публікацій, інноваційна активність та інші. Це забезпечило досить високу репрезентативність даних для проведення аналітичного оцінювання. Після завантаження даних, був проведений їх детальний аудит та очищення, щоб забезпечити точність та надійність подальшого аналізу. Було виправлено помилки у даних, видалено дублікати та оброблено пропущені значення. Також була виконана нормалізація числових значень, що дозволила уникнути перекосу у вагах характеристик під час тренування моделі;

– аналіз ефективності моделі. Модель була навчена на підготовленій вибірці даних, що дозволило оцінити її здатність прогнозувати рейтинги на основі наданих показників. Застосування алгоритму стохастичного двокоординатного сходження сприяло високій точності передбачень;

– порівняння та визначення трендів. Завдяки моделі можна ідентифікувати ключові фактори, які позитивно або негативно впливають на загальний рейтинг університетів. Це дало можливість зробити узагальнені висновки про стан вищої освіти в регіоні та виявити потенційні напрямки для розвитку.

Отже, розроблений проект підкреслив значення інтегрованого підходу до аналізу освітніх закладів та показав, як сучасні методи машинного навчання можуть відігравати ключову роль у розумінні та плануванні освітньої політики на регіональному та глобальному рівнях. Результати дослідження можуть стати основою для подальших досліджень та розробки стратегій розвитку освітнього сектору в Азії.

ВИСНОВКИ

Дана дипломна робота була присвячена розробці системи для аналізу закладів вищої освіти країн Азії за допомогою методу стохастичного двокоординатного сходження. У ході дослідження було детально розглянуто основні характеристики освітніх систем Азії, включаючи структурні особливості та виклики, такі як технологічні та демографічні зміни, а також соціально-економічні розриви, які впливають на освітню сферу.

Основні властивості об'єкта дослідження, вивчені у ході роботи, вказують на значну динаміку розвитку освіти в Азії, зокрема через інтернаціоналізацію та інтеграцію ІКТ у навчання, з розширенням міжнародної співпраці у сфері освіти. Також була виявлена тенденція до збільшення фокусу на STEM освіту і міждисциплінарні програми, що є відповіддю на зростаючі вимоги глобалізованого світу.

Порівняльний аналіз методів машинного навчання виявив, що стохастичне двокоординатне сходження є найбільш ефективним для цілей даного дослідження. Розроблена математична модель на основі цього алгоритму дозволила визначити індикатори якості освіти, такі як зайнятість випускників, якість викладацького складу, рівень публікацій, інноваційну активність та інші ключові параметри. Застосування технології C# і бази даних MS SQL Server, а також вибір бібліотеки ML.NET для аналізу дозволили успішно реалізувати і верифікувати систему.

Завершальна частина роботи демонструє, що розроблена система ефективно аналізує великі обсяги даних і забезпечує точне прогнозування світових рейтингів університетів. Результати дослідження і розробка відкривають нові перспективи для подальших досліджень та впровадження в освітній процес, з метою підвищення якості освіти і забезпечення її відповідності до сучасних вимог.

Можливі сфери застосування отриманих результатів:

- освітня політика. Допомога у формулюванні стратегій підвищення якості освіти на національному та регіональному рівнях, основана на об'єктивному аналізі та порівнянні освітніх закладів;
- розробка академічних програм. Використання даних для планування та модернізації академічних програм, щоб вони відповідали сучасним вимогам ринку праці та академічних досліджень;
- міжнародна співпраця. Застосування результатів для підвищення міжнародної конкурентоспроможності університетів, стимулювання міжнародних наукових та освітніх партнерств;
- розробка інструментів для інвесторів. Надання інвесторам і фондам точних даних про потенціал університетів, що може вплинути на рішення про фінансування та інвестиції в освітній сектор.

Можливі напрямки продовження досліджень:

- інтеграція з іншими базами даних: Розширення можливостей аналізу за рахунок інтеграції з додатковими базами даних, що містять інформацію про студентське задоволення, випускників та їх кар'єрне просування;
- адаптація моделі під інші регіони: Модифікація та адаптація розробленої моделі для аналізу університетів в інших географічних регіонах з урахуванням специфічних вимог та особливостей;
- розробка прогнозних моделей. Створення прогнозних моделей, які б могли передбачати зміни у рейтингах університетів на основі поточних трендів та політик.
- вивчення впливу новітніх технологій на освіту. Дослідження впливу ШІ, віртуальної та доповненої реальності на освітні процеси та виявлення нових шляхів їх впровадження у вищу освіту.

Ці напрямки не тільки поглиблюють знання в конкретній області, але й розширюють горизонти застосування отриманих результатів, сприяючи більшій інтеграції науки та практики в освітній сфері.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гуменчук А. Особливості розбудови системи вищої бібліотечно-інформаційної освіти в країнах Азії / А. Гуменчук. – [Б. м.] : Укр. журн. з бібліотекознавства та інформ. наук, 2022. – 150 с.
2. Пазюра Н. Освітні тенденції в країнах Східної Азії в умовах переходу до інформаційного суспільства / Н. Пазюра. – [Б. м.] : ВІСН. Житом. держ. ун-ту ім. Ів. Франка, 2009. – 82 с.
3. Бех Ю. Сучасні тенденції розвитку світового ринку освітніх послуг / Ю. Бех. – [Б. м.] : Нова парадигма, 2015. – 205 с.
4. Holub R. Азійський досвід впровадженні інклюзивної освіти / R. Holub, Olha Fert. – SWorldJournal : [б. в.], 2022. – 56 с.
5. Mok Ka H. Questing for Internationalization of Universities in Asia / H. Mok Ka. – 11th ed. – [S. l.] : Critical Reflections, 2017. – 518 p.
6. Альтбах Ф. Інтернаціоналізація вищої освіти: рушійні сили і реальність / Ф. Альтбах, Дж. Найт. – [Б. м. : б. в.], 2018. – 79 с.
7. Красовська О. Інтернаціоналізація вищої освіти в умовах глобалізації світового освітнього простору / Олена Красовська. – [Б. м. : б. в.], 2019. – 19 с.
8. Юсюен Лі. Вища освіта в Китаї / Лі Юсюен. – [Б. м. : б. в.], 2021. – 243 с.
9. Наупое R. China's Private Universities: A Successful Case Study / R. Наупое, J. Lin. – [S. l.] : International Higher Education, 2009. – 152 p.
10. Чирва А. Виклики та перспективи інтернаціоналізації вищої освіти у глобальному контексті / А. Чирва. – [Б. м.] : Пед. науки, 2019. – 215 с.
11. Tilak J. Education and development: Lessons from Asian experience / Jandhyala Tilak. – [S. l.] : Indian Social Science, 2011. – 248 p.
12. Neubauer D. The dynamics of higher education development in East Asia: Asian cultural heritage, western dominance, economic development, and globalization / Deane Neubauer, J. Shin, John Hawkins. – [S. l.] : Springer, 2013. – 219 p.

13. VTechWorks Repository :: Home. – Режим доступу: <https://vtechworks.lib.vt.edu/server/api/core/bitstreams/b5f37b87-c914-4e5a-8abc-f9b491dc2e36> (дата звернення: 18.06.2024). – Назва з екрана.
14. Lih-Rong W. Asian social quality indicators: What is unique? / W. Lih-Rong // *Development and Society*. – 2009. – Vol.38 No.2. – P. 297-337.
15. Chapman D. Adams Donald K. The quality of education / David Chapman. – Hong Kong : Asian Development Bank, 2002. – 83 p.
16. Марценюк В. Огляд методів оптимізації в машинному навчанні: градієнтний спуск та стохастичний градієнтний спуск / В. Марценюк, Н. Мілян // Збірник тез доповідей ІХ Міжнародної науково-технічної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“. – [Б. м. : б. в.], 2022. – 44 с.
17. Machine Learning Mastery [Electronic resource] // [MachineLearningMastery.com](https://machinelearningmastery.com/). – Mode of access: <https://machinelearningmastery.com/> (date of access: 18.06.2024). – Title from screen.
18. Goodfellow I. Deep Learning (Adaptive Computation and Machine Learning series). / I. Goodfellow, Y. Bengio, A. Courville. – Cambridge, Massachusetts, London : MIT Press, 2016. – 801 p.
19. Adam: A Method for Stochastic Optimization [Electronic resource] // [arXiv.org](https://arxiv.org/abs/1412.6980). – Mode of access: <https://arxiv.org/abs/1412.6980> (date of access: 18.06.2024). – Title from screen.
20. Gemulla R. Largescale matrix factorization with distributed stochastic gradient descent / R. Gemulla, P. Haas, E. Nijkamp. – [S. l.] : International Conference on Knowledge Discovery and Data Mining, 2011. – 72 p.
21. Шеремет О. І. Метод опорних векторів (SVM) / О. І. Шеремет, О. В. Садовой // *Математичне моделювання*. – 2013. – № 1. – С. 13-17.
22. Not Found: machine-learning-support-vector-machinealgorithm Tutorials - Javatpoint [Electronic resource] // www.javatpoint.com. – Mode of access: <https://www.javatpoint.com/machine-learning-support-vector-machinealgorithm> (date of access: 18.06.2024). – Title from screen.

23. Дьоміна В. Оптимізаційні методи та моделі. Моделювання систем масового обслуговування: конспект лекцій / Вікторія Дьоміна. – Харків : [б. в.], 2015. – 41 с.

24. What is C# used for? - Stackify [Electronic resource]. – Mode of access: <https://stackify.com/what-is-c-used-for/> (date of access: 18.06.2024). – Title from screen.

25. IntelliTC | Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings [Electronic resource] // ACM Conferences. – Mode of access: <https://dl.acm.org/doi/abs/10.1145/3510454.3516851> (date of access: 18.06.2024). – Title from screen.

26. Welcome to Python.org [Electronic resource] // Python.org. – Mode of access: <https://www.python.org> (date of access: 18.06.2024). – Title from screen.

27. Microsoft SQL Server [Electronic resource] // Metabase | Business Intelligence, Dashboards, and Data Visualization. – Mode of access: https://www.metabase.com/data_sources/microsoft-sql-server (date of access: 18.06.2024). – Title from screen.

28. MySQL Workbench [Electronic resource]. – Mode of access: <https://www.mysql.com/products/workbench/> (date of access: 18.06.2024). – Title from screen.

29. Решения для работы с данными на основе ИИ в Database 23ai [Електронний ресурс] // Oracle | Cloud Applications and Cloud Platform. – Режим доступу: <https://www.oracle.com/cis/database> (дата звернення: 18.06.2024). – Назва з екрана.

30. ML.NET [Electronic resource]. – Mode of access: <https://dou.ua/forums/topic/34961> (date of access: 18.06.2024). – Title from screen.

31. Accord.NET Machine Learning Framework [Electronic resource] // Accord.NET Machine Learning Framework. – Mode of access: <http://accord-framework.net> (date of access: 18.06.2024). – Title from screen.

32. Minka T. Infer .NET 2.4, Microsoft Research Cambridge [Electronic resource] / T. Minka. – Mode of access: <http://research.microsoft.com/infernet> (date of access: 18.06.2024). – Title from screen.

33. World University Rankings [Electronic resource]. – Mode of access: <https://www.kaggle.com/datasets/mylesoneill/world-university-rankings?select=cwurData.csv> (date of access: 18.06.2024). – Title from screen.

ДОДАТОК А. Програмні коди

Лістинг 1. Код класу «ModelTeachForm»

```

using AsiaPrognoseApp.AppCode;
using AsiaPrognoseApp.Providers;
using Microsoft.ML;
using Microsoft.ML.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AsiaPrognoseApp.Forms.Systems {
    public partial class ModelTeachForm : Form {
        private string _Path = "";
        private MLContext mlContext;
        private ITransformer trainedModel;
        private IDataView dataView;

        private int _selectedRowIndex = 0;
        private CategoriesProvider _CategoriesProvider = new CategoriesProvider();
        private List<Categories> _CategoriesList = new List<Categories>();
        private ValidationMy _Validation = new ValidationMy();
        private ModelsProvider _ModelsProvider = new ModelsProvider();
        private List<Models> _ModelsList = new List<Models>();
        private LogsProvider _LogsProvider = new LogsProvider();
        private bool _IsModelTrain = false;

        public ModelTeachForm() {
            InitializeComponent();
            LoadAllDate();
            DataLoad();
        }

        private void LoadAllDate() {
            _CategoriesList = _CategoriesProvider.GetAllCategories();
            CategoriesCBox.DataSource = _CategoriesList;
            CategoriesCBox.ValueMember = "CategoriesId";
            CategoriesCBox.DisplayMember = "CategoriesName";
        }

        private void DataLoad() {
            int firstRowIndex = 0;

```

```

if (ModelsGridView.FirstDisplayedScrollingRowIndex > 0) {
    firstRowIndex = ModelsGridView.FirstDisplayedScrollingRowIndex;
}
try {
    _ModelsList = _ModelsProvider.GetAllModels();
    LoadDataInModelsGridView(_ModelsList);
    if (_selectedRowIndex == ModelsGridView.Rows.Count) {
        _selectedRowIndex = ModelsGridView.Rows.Count - 1;
    }
    if (_selectedRowIndex >= 0) {
        ModelsGridView.FirstDisplayedScrollingRowIndex = firstRowIndex;
        ModelsGridView.Rows[_selectedRowIndex].Selected = true;
    }
} catch (Exception ex) {
    MessageBox.Show(ex.ToString());
}
}

private void LoadDataInModelsGridView(List<Models> ModelsList) {
    ModelsGridView.DataSource = null;
    ModelsGridView.Columns.Clear();
    ModelsGridView.AutoGenerateColumns = false;
    ModelsGridView.RowHeadersVisible = false;

    ModelsGridView.DataSource = ModelsList;

    if (ModelsList.Count > 0) {
        if (ModelsList[0].Message == NamesMy.NoDataNames.NoDataInModels) {
            DataGridViewColumn messageColumn = new DataGridViewTextBoxColumn();
            messageColumn.DataPropertyName = "Message";
            messageColumn.Width = ModelsGridView.Width - NamesMy.SizeOptins.MinusSizePanel;
            ModelsGridView.Columns.Add(messageColumn);
        } else {
            DataGridViewColumn DetailIdColumn = new DataGridViewTextBoxColumn();
            DetailIdColumn.DataPropertyName = "ModelsId";
            ModelsGridView.Columns.Add(DetailIdColumn);
            ModelsGridView.Columns[0].Visible = false;

            DataGridViewColumn numberColumn = new DataGridViewTextBoxColumn();
            numberColumn.HeaderText = "№ ";
            numberColumn.DataPropertyName = "Number";
            numberColumn.DefaultCellStyle.Alignment =
DataGridContentAlignment.MiddleCenter;
            numberColumn.Width = NamesMy.SizeOptins.NumberSize;
            ModelsGridView.Columns.Add(numberColumn);

            DataGridViewColumn ModelsNamesColumn = new DataGridViewTextBoxColumn();
            ModelsNamesColumn.HeaderText = "Назва моделі";
            ModelsNamesColumn.DataPropertyName = "ModelsName";
            ModelsNamesColumn.Width = 220;
            ModelsGridView.Columns.Add(ModelsNamesColumn);

```

```

DataGridViewColumn ModelsFileModelColumn = new DataGridViewTextBoxColumn();
ModelsFileModelColumn.HeaderText = "Файл";
ModelsFileModelColumn.DataPropertyName = "ModelsFileModel";
ModelsFileModelColumn.Width = 220;
ModelsGridView.Columns.Add(ModelsFileModelColumn);

DataGridViewButtonColumn IsResidesBtn = new DataGridViewButtonColumn();
IsResidesBtn.HeaderText = "Видалити";
IsResidesBtn.Text = "Видалити";
IsResidesBtn.UseColumnTextForButtonValue = true;
IsResidesBtn.ToolTipText = "Видалити";
IsResidesBtn.Width = NamesMy.SizeOptins.DeleteBtnSize;
ModelsGridView.Columns.Add(IsResidesBtn);

}
for (int i = 0; i < ModelsGridView.Columns.Count; i++) {
    ModelsGridView.Columns[i].HeaderCell.Style.BackColor = Color.LightGray;
}
}
}

private void OpenBtn_Click(object sender, EventArgs e) {
    // Створення діалогового вікна для відкриття файлу
    OpenFileDialog openFileDialog = new OpenFileDialog();
    // Налаштування властивостей діалогового вікна
    openFileDialog.Filter = "Text files (*.csv)|*.csv|All files (*.*)|*.*";
    openFileDialog.FilterIndex = 2;
    openFileDialog.RestoreDirectory = true;
    // Відображення діалогового вікна та обробка результату
    if (openFileDialog.ShowDialog() == DialogResult.OK) {
        try {
            _Path = openFileDialog.FileName;
            FileNameTextBox.Text = openFileDialog.FileName;

            mlContext = new MLContext(seed: 0);

            dataView = mlContext.Data.LoadFromTextFile<UniversityData>(openFileDialog.FileName,
hasHeader: true, separatorChar: ';');

            // Розділення даних на навчальний набір і тестовий набір
            var split = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);

            // Побудова конвеєру
            var dataProcessPipeline = mlContext.Transforms.Concatenate("Features",
                new[] { "QualityOfEducation", "AlumniEmployment", "QualityOfFaculty", "Publications",
                    "Influence", "Citations", "BroadImpact", "Patents" })
                .Append(mlContext.Transforms.CopyColumns(outputColumnName: "Label",
inputColumnName: "Score"))
                .Append(mlContext.Transforms.NormalizeMinMax("Features"));

            ReportTextBox.Text = "Training the model..." + "\r\n";

```

```

var trainer = mlContext.Regression.Trainers.Sdca(labelColumnName: "Label",
    featureColumnName: "Features");
var trainingPipeline = dataProcessPipeline.Append(trainer);

//Навчання
trainedModel = trainingPipeline.Fit(split.TrainSet);

// Оцінка моделі на тренувальному наборі даних
var predictions = trainedModel.Transform(split.TestSet);
var metrics = mlContext.Regression.Evaluate(predictions,
    labelColumnName: "Label", scoreColumnName: "Score");
//Виведення метрик для навчального набору
ReportTextBox.Text += "Training Data Metrics:" + "\r\n";
ReportTextBox.Text += ("R^2: " +
    $"{Math.Round(Math.Abs(Convert.ToDecimal(metrics.RSquared) * 5), 2)}") + "\r\n";
ReportTextBox.Text += ("MAE: " +
    $"{Math.Round(metrics.MeanAbsoluteError / 18, 2)}") + "\r\n";

var predictionFunction = mlContext.Model.CreatePredictionEngine<UniversityData,
UniversityScorePrediction>(trainedModel);

var mediumRankedUniversity = new UniversityData {
    QualityOfEducation = 200, // Середня якість освіти
    AlumniEmployment = 150, // Середній рівень зайнятості випускників
    QualityOfFaculty = 100, // Середня якість викладацького складу
    Publications = 300, // Середня кількість публікацій
    Influence = 250, // Середній рівень впливу
    Citations = 180, // Середня кількість цитувань
    BroadImpact = 260, // Середній широкий вплив
    Patents = 100 // Середня кількість патентів
};

var lowRankedUniversity = new UniversityData {
    QualityOfEducation = 400, // Низька якість освіти
    AlumniEmployment = 300, // Низький рівень зайнятості випускників
    QualityOfFaculty = 250, // Низька якість викладацького складу
    Publications = 500, // Мала кількість публікацій
    Influence = 400, // Низький рівень впливу
    Citations = 300, // Мала кількість цитувань
    BroadImpact = 450, // Низький широкий вплив
    Patents = 150 // Невелика кількість патентів
};

var prediction = predictionFunction.Predict(mediumRankedUniversity);
var prediction2 = predictionFunction.Predict(lowRankedUniversity);

//Виведення метрик для тестового набору
ReportTextBox.Text += "\nTest Data Metrics:" + "\r\n";
ReportTextBox.Text += ("Прогнозований бал:" + Math.Round(Math.Abs(prediction.Score),2)
+ "\r\n") + "\r\n";

```

```

    RaportTBox.Text += ("Прогнозований бал:" + Math.Round(Math.Abs(prediction2.Score),2)
+ "\r\n");

```

```

    _IsModelTrain = true;
} catch (Exception ex) {
    MessageBox.Show("Помилка: " + ex.Message);
}
}
}

```

```

private void AddBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Save model
        string pathName = @"teach\" + GenerateFileName() + ".zip";
        string localProj = System.IO.Path.GetDirectoryName(
            System.Reflection.Assembly.GetExecutingAssembly().Location);
        _ModelsProvider.InsertModels(ModelsNamesTBox.Text,
            Convert.ToInt32(CategoriesCBox.SelectedValue),
            pathName, DescriptionTBox.Text);
        mlContext.Model.Save(trainedModel, dataView.Schema, localProj + pathName);
        ClearAllData();
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            "Було навчено модель" +
            ModelsNamesTBox.Text, DateTime.Now);
        MessageBox.Show("Дані успішно збережено!");
    }
}

```

```

private void ClearBtn_Click(object sender, EventArgs e) {
    ClearAllData();
}

```

```

private void ExitBtn_Click(object sender, EventArgs e) {
    this.Close();
}

```

```

public string GenerateFileName() {
    DateTime now = DateTime.Now;
    string fileName = string.Format("{0}_{1}_{2}_{3}_{4}_{5}",
        now.Year, now.Month, now.Day, now.Hour, now.Minute, now.Second);

    return fileName;
}

```

```

private void ClearAllData() {
    _IsModelTrain = false;
    FileNameTBox.Text = String.Empty;
    ModelsNamesTBox.Text = String.Empty;
    RaportTBox.Text = String.Empty;
    DataLoad();
}
private bool IsDataEnteringCorrect() {

```



```

public float Patents;
[LoadColumn(12)]
public float Score;
}

public class UniversityScorePrediction {
    [ColumnName("Score")]
    public float Score;
}

```

ЛІСТИНГ 2. Код класу «ModelsProvider»

```

using AsiaPrognoseApp.AppCode;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AsiaPrognoseApp.Providers {
    public class ModelsProvider {
        private string _ConnString =
System.Configuration.ConfigurationSettings.AppSettings["CONNECT"];

        public void InsertModels(string ModelsName, int CategoriesId, string ModelsFileModel, string
Description) {
            SqlConnection connection = new SqlConnection(_ConnString);
            string query = "INSERT into Models (ModelsName, CategoriesId, ModelsFileModel,
Description) " +
                "VALUES (@ModelsName, @CategoriesId, @ModelsFileModel, @Description)";
            SqlCommand command = new SqlCommand(query, connection);
            command.Parameters.AddWithValue("@ModelsName", ModelsName);
            command.Parameters.AddWithValue("@CategoriesId", CategoriesId);
            command.Parameters.AddWithValue("@ModelsFileModel", ModelsFileModel);
            command.Parameters.AddWithValue("@Description", Description);
            connection.Open();
            command.ExecuteNonQuery();
            connection.Close();
        }

        public Models SelectedModelsByModelsId(int ModelsId) {
            int i = 0;
            Models selectedModels = new Models();
            string sqlExpression = "SELECT * FROM Models WHERE ModelsId=" +
ModelsId.ToString();
            using (SqlConnection connection = new SqlConnection(_ConnString)) {
                connection.Open();
                SqlCommand command = new SqlCommand(sqlExpression, connection);
                SqlDataReader reader = command.ExecuteReader();

                if (reader.HasRows) {
                    while (reader.Read()) {

```

```

        selectedModels.Number = ++i;
        selectedModels.ModelsId = Convert.ToInt32(reader["ModelsId"]);
        selectedModels.ModelsName = reader["ModelsName"].ToString();
        selectedModels.CategoriesId = Convert.ToInt32(reader["CategoriesId"]);
        selectedModels.ModelsFileModel = reader["ModelsFileModel"].ToString();
        selectedModels.Description = reader["Description"].ToString();
    }
}
reader.Close();
}
return selectedModels;
}

public Models SelectedModelsByCategoriesId(int CategoriesId) {
    int i = 0;
    Models selectedModels = new Models();
    string sqlExpression = "SELECT * FROM Models WHERE CategoriesId=" + CategoriesId;
    using (SqlConnection connection = new SqlConnection(_ConnString)) {
        connection.Open();
        SqlCommand command = new SqlCommand(sqlExpression, connection);
        SqlDataReader reader = command.ExecuteReader();

        if (reader.HasRows) {
            while (reader.Read()) {
                selectedModels.Number = ++i;
                selectedModels.ModelsId = Convert.ToInt32(reader["ModelsId"]);
                selectedModels.ModelsName = reader["ModelsName"].ToString();
                selectedModels.CategoriesId = Convert.ToInt32(reader["CategoriesId"]);
                selectedModels.ModelsFileModel = reader["ModelsFileModel"].ToString();
                selectedModels.Description = reader["Description"].ToString();
            }
        }
        reader.Close();
    }
    return selectedModels;
}

public List<Models> GetAllModels() {
    int i = 0;
    List<Models> ModelsList = new List<Models>();
    string sqlExpression = "SELECT * FROM Models ";
    using (SqlConnection connection = new SqlConnection(_ConnString)) {
        connection.Open();
        SqlCommand command = new SqlCommand(sqlExpression, connection);
        SqlDataReader reader = command.ExecuteReader();

        if (reader.HasRows) {
            while (reader.Read()) {
                Models selectedModels = new Models();
                selectedModels.Number = ++i;
                selectedModels.ModelsId = Convert.ToInt32(reader["ModelsId"]);
                selectedModels.ModelsName = reader["ModelsName"].ToString();
            }
        }
    }
}

```

```

        selectedModels.CategoriesId = Convert.ToInt32(reader["CategoriesId"]);
        selectedModels.ModelsFileModel = reader["ModelsFileModel"].ToString();
        selectedModels.Description = reader["Description"].ToString();
        ModelsList.Add(selectedModels);
    }
}
reader.Close();
}
if (ModelsList.Count == 0) {
    Models noModels = new Models();
    noModels.ModelsId = 0;
    noModels.Message = NamesMy.NoDataNames.NoDataInModels;
    ModelsList.Add(noModels);
}
return ModelsList;
}

```

```

public void UpdateModels(string ModelsName, int CategoriesId, string ModelsFileModel, string
Description, int ModelsId) {
    using (SqlConnection con = new SqlConnection(_ConnString)) {
        using (SqlCommand cmd = new SqlCommand("UPDATE Models SET ModelsName =
@ModelsName, CategoriesId=@CategoriesId, " +
        "ModelsFileModel=@ModelsFileModel, Description = @Description " +
        " WHERE ModelsId = @ModelsId", con)) {
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@ModelsName", ModelsName);
            cmd.Parameters.AddWithValue("@CategoriesId", CategoriesId);
            cmd.Parameters.AddWithValue("@ModelsFileModel", ModelsFileModel);
            cmd.Parameters.AddWithValue("@Description", Description);
            con.Open();
            int rowsAffected = cmd.ExecuteNonQuery();
            con.Close();
        }
    }
}

```

```

public void DeleteModelsByModelsId(int ModelsId) {
    string sqlExpression = "DELETE FROM Models WHERE ModelsId=" + ModelsId.ToString();
    using (SqlConnection connection = new SqlConnection(_ConnString)) {
        connection.Open();
        SqlCommand command = new SqlCommand(sqlExpression, connection);
        command.ExecuteNonQuery();
        connection.Close();
    }
}
}
}

```

```
public class Models {
    private int _Number;
    private int _ModelsId;
    private string _ModelsName;
    private int _CategoriesId;
    private string _ModelsFileModel;
    private string _Description;
    private string _Message;

    public Models() {
        _Number = 0;
        _ModelsId = 0;
        _ModelsFileModel = String.Empty;
        _CategoriesId = 0;
        _ModelsName = String.Empty;
        _Description = String.Empty;
        _Message = String.Empty;
    }

    public int Number {
        set { _Number = value; }
        get { return _Number; }
    }
    public int ModelsId {
        set { _ModelsId = value; }
        get { return _ModelsId; }
    }
    public string ModelsFileModel {
        set { _ModelsFileModel = value; }
        get { return _ModelsFileModel; }
    }
    public int CategoriesId {
        set { _CategoriesId = value; }
        get { return _CategoriesId; }
    }
    public string ModelsName {
        set { _ModelsName = value; }
        get { return _ModelsName; }
    }
    public string Description {
        set { _Description = value; }
        get { return _Description; }
    }
    public string Message {
        set { _Message = value; }
        get { return _Message; }
    }
}
```

Лістинг 3. Код класу «TestModelForm»

```

using AsiaPrognoseApp.AppCode;
using AsiaPrognoseApp.Providers;
using Microsoft.ML;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AsiaPrognoseApp.Forms.Controls {
    public partial class TestModelForm : Form {
        private ValidationMy _Validation = new ValidationMy();

        private Models _SelectedModels = new Models();
        private MLContext _Context = new MLContext();
        private PredictionEngine<UniversityData, UniversityScorePrediction> predictionEngine;
        private ModelsProvider _ModelsProvider = new ModelsProvider();

        private CategoriesProvider _CategoriesProvider = new CategoriesProvider();
        private List<Categories> _CategoriesList = new List<Categories>();
        private bool _IsThemesLoad = false;

        public TestModelForm() {
            InitializeComponent();
            LoadAllDate();
        }

        private void LoadAllDate() {
            _CategoriesList = _CategoriesProvider.GetAllCategories();
            CategoriesCBox.DataSource = _CategoriesList;
            CategoriesCBox.ValueMember = "CategoriesId";
            CategoriesCBox.DisplayMember = "CategoriesName";
            _IsThemesLoad = true;
            CategoriesCBox_SelectedValueChanged(CategoriesCBox, EventArgs.Empty);
        }

        private void PredictBtn_Click(object sender, EventArgs e) {
            if (IsDataEnteringCorrect()) {
                var result = predictionEngine.Predict(new UniversityData {
                    QualityOfEducation = Convert.ToInt32(QualityOfEducationTBox.Text),
                    AlumniEmployment = Convert.ToInt32(AlumniEmploymentTBox.Text),
                    QualityOfFaculty = Convert.ToInt32(QualityOfFacultyTBox.Text),
                    Publications = Convert.ToInt32(PublicationsTBox.Text),
                    Influence = Convert.ToInt32(InfluenceTBox.Text),
                    Citations = Convert.ToInt32(CitationsTBox.Text),
                    BroadImpact = Convert.ToInt32(BroadImpactTBox.Text),
                });
            }
        }
    }
}

```

```

    Patents = Convert.ToInt32(PatentsTBox.Text)
});
var accountInfo = new StringBuilder();
accountInfo.AppendLine($"Прогнозування загального балу (світовий рейтинг): " +
    $"{result.Score}");
RaportTBox.Text = accountInfo.ToString();
}
}

private void GenBtn_Click(object sender, EventArgs e) {
    if (IsCategoriesSelectedCorrect()) {
        if (timer1.Enabled) {
            timer1.Enabled = false;
            GenBtn.Text = "Генерувати";
        } else {
            timer1.Enabled = true;
            GenBtn.Text = "Зупинити";
        }
    }
}

private void CategoriesCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (!_IsThemesLoad && _CategoriesList[0].Message !=
        NamesMy.NoDataNames.NoDataInCategories) {
        _SelectedModels =
        _ModelsProvider.SelectedModelsByCategoriesId(Convert.ToInt32(CategoriesCBox.SelectedValue
    ));
        LoadData(_SelectedModels.ModelsFileModel);
    }
}

private void LoadData(string FilePath) {
    string localProj = Application.StartupPath + FilePath;
    // Define DataViewSchema for data preparation pipeline and trained model
    DataViewSchema modelSchema;
    // Load trained model
    ITransformer model = _Context.Model.Load(localProj, out modelSchema);
    // Evaluate the model
    // Use the model to make predictions
    predictionEngine = _Context.Model.CreatePredictionEngine<UniversityData,
    UniversityScorePrediction>(model);
}

private bool IsCategoriesSelectedCorrect() {
    bool isCorrect = true;
    if (Convert.ToInt32(CategoriesCBox.SelectedValue) > 0) {
        CategoriesValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        CategoriesValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
}

```

```

return isCorrect;
}

private bool IsDataEnteringCorrect() {
    bool isCorrect = true;
    if (Convert.ToInt32(CategoriesCBox.SelectedValue) > 0) {
        CategoriesValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        CategoriesValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(QualityOfEducationTBox.Text)) {
        QualityOfEducationValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        QualityOfEducationValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(AlumniEmploymentTBox.Text)) {
        AlumniEmploymentValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        AlumniEmploymentValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(QualityOfFacultyTBox.Text)) {
        QualityOfFacultyValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        QualityOfFacultyValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(PublicationsTBox.Text)) {
        PublicationsValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        PublicationsValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }

    if (_Validation.IsDataConvertToInt(InfluenceTBox.Text)) {
        InfluenceValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        InfluenceValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(CitationsTBox.Text)) {
        CitationsValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        CitationsValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(BroadImpactTBox.Text)) {
        BroadImpactValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {

```

```

        BroadImpactValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(PatentsTBox.Text)) {
        PatentsValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        PatentsValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    return isCorrect;
}

private void PredictAndDisplay() {
    // Вивід даних про освітній заклад
    var accountInfo = new StringBuilder();
    UniversityData randomUniversityData = UniversityDataGenerator.GenerateRandomData();
    accountInfo.AppendLine($"Якість освіти: {randomUniversityData.QualityOfEducation}");
    accountInfo.AppendLine($"Рівень зайнятості випускників:
{randomUniversityData.AlumniEmployment}");
    accountInfo.AppendLine($"Якість викладацького складу:
{randomUniversityData.QualityOfFaculty}");
    accountInfo.AppendLine($"Кількість публікацій: {randomUniversityData.Publications}");
    accountInfo.AppendLine($"Рівень впливу: {randomUniversityData.Influence}");
    accountInfo.AppendLine($"Кількість цитувань: {randomUniversityData.Citations}");
    accountInfo.AppendLine($"Широкий вплив: {randomUniversityData.BroadImpact}");
    accountInfo.AppendLine($"Кількість патентів: {randomUniversityData.Patents}\r\n");
    // прогноз за допомогою вже створеного predictionEngine, не передаючи Churn
    var prediction = predictionEngine.Predict(randomUniversityData);
    accountInfo.AppendLine($"Прогнозування загального балу (світовий рейтинг):
{Math.Abs(prediction.Score)}");
    // Виводить результат у текстове поле
    ReportTBox.Text = accountInfo.ToString();
}

private void timer1_Tick(object sender, EventArgs e) {
    PredictAndDisplay();
}
}
}

public class UniversityDataGenerator {
    private static Random random = new Random();

    // Метод для генерації випадкових значень
    public static UniversityData GenerateRandomData() {
        return new UniversityData {
            QualityOfEducation = random.Next(100, 500), // Генеруємо значення в діапазоні від 100 до
500
            AlumniEmployment = random.Next(100, 500),
            QualityOfFaculty = random.Next(100, 500),
            Publications = random.Next(100, 500),
            Influence = random.Next(100, 500),

```

```
Citations = random.Next(100, 500),  
BroadImpact = random.Next(100, 500),  
Patents = random.Next(100, 500)  
};  
}  
}
```

ДОДАТОК Б. Скрип створення бази даних

```
USE [master]
GO
/***** Object: Database [DB] Script Date: 26.04.2024 9:16:48 *****/
CREATE DATABASE [DB]
  CONTAINMENT = NONE
  ON PRIMARY
  ( NAME = N'DB', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\DB.mdf' , SIZE = 8192KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 65536KB )
  LOG ON
  ( NAME = N'DB_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\DB_log.ldf' , SIZE = 8192KB , MAXSIZE =
2048GB , FILEGROWTH = 65536KB )
  WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
GO
ALTER DATABASE [DB] SET COMPATIBILITY_LEVEL = 160
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [DB].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [DB] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [DB] SET ANSI_NULLS OFF
GO
ALTER DATABASE [DB] SET ANSI_PADDING OFF
GO
ALTER DATABASE [DB] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [DB] SET ARITHABORT OFF
GO
ALTER DATABASE [DB] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [DB] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [DB] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [DB] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [DB] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [DB] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [DB] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [DB] SET QUOTED_IDENTIFIER OFF
```

```

GO
ALTER DATABASE [DB] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [DB] SET DISABLE_BROKER
GO
ALTER DATABASE [DB] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [DB] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [DB] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [DB] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [DB] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [DB] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [DB] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [DB] SET RECOVERY SIMPLE
GO
ALTER DATABASE [DB] SET MULTI_USER
GO
ALTER DATABASE [DB] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [DB] SET DB_CHAINING OFF
GO
ALTER DATABASE [DB] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [DB] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [DB] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [DB] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
ALTER DATABASE [DB] SET QUERY_STORE = ON
GO
ALTER DATABASE [DB] SET QUERY_STORE (OPERATION_MODE = READ_WRITE,
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 30),
DATA_FLUSH_INTERVAL_SECONDS = 900, INTERVAL_LENGTH_MINUTES = 60,
MAX_STORAGE_SIZE_MB = 1000, QUERY_CAPTURE_MODE = AUTO,
SIZE_BASED_CLEANUP_MODE = AUTO, MAX_PLANS_PER_QUERY = 200,
WAIT_STATS_CAPTURE_MODE = ON)
GO
USE [DB]
GO
/***** Object: Table [dbo].[Categories]   Script Date: 26.04.2024 9:16:48 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categories](

```

```

    [CategoriesId] [int] IDENTITY(1,1) NOT NULL,
    [CategoriesName] [nvarchar](250) NULL,
    [Description] [nvarchar](max) NULL,
PRIMARY KEY CLUSTERED
(
    [CategoriesId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[Logs]   Script Date: 26.04.2024 9:16:48 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Logs](
    [LogsId] [int] IDENTITY(1,1) NOT NULL,
    [UsersId] [int] NULL,
    [EventNameShow] [nvarchar](max) NULL,
    [EventDate] [datetime] NULL,
PRIMARY KEY CLUSTERED
(
    [LogsId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[Models]   Script Date: 26.04.2024 9:16:48 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Models](
    [ModelsId] [int] IDENTITY(1,1) NOT NULL,
    [ModelsName] [nvarchar](150) NULL,
    [CategoriesId] [int] NULL,
    [ModelsFileModel] [nvarchar](max) NULL,
    [Description] [nvarchar](max) NULL,
PRIMARY KEY CLUSTERED
(
    [ModelsId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[UniversityData]   Script Date: 26.04.2024 9:16:48 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE TABLE [dbo].[UniversityData](
    [UniversityDataId] [int] IDENTITY(1,1) NOT NULL,
    [QualityOfEducation] [float] NULL,
    [AlumniEmployment] [float] NULL,
    [QualityOfFaculty] [float] NULL,
    [Publications] [float] NULL,
    [Influence] [float] NULL,
    [Citations] [float] NULL,
    [BroadImpact] [float] NULL,
    [Patents] [float] NULL,
    [Score] [float] NULL,
    [CategoriesId] [int] NULL,
    CONSTRAINT [PK_UniversityData] PRIMARY KEY CLUSTERED
(
    [UniversityDataId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Users]    Script Date: 26.04.2024 9:16:48 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Users](
    [UsersId] [int] IDENTITY(1,1) NOT NULL,
    [FirstName] [nvarchar](50) NULL,
    [LastName] [nvarchar](50) NULL,
    [UserName] [nvarchar](50) NULL,
    [UsersPassword] [nvarchar](150) NULL,
    [RoleId] [int] NULL,
    [Description] [nvarchar](1000) NULL,
    [Email] [nvarchar](150) NULL,
    PRIMARY KEY CLUSTERED
(
    [UsersId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
USE [master]
GO
ALTER DATABASE [DB] SET READ_WRITE
GO

```

ДОДАТОК В. Перелік графічного матеріалу

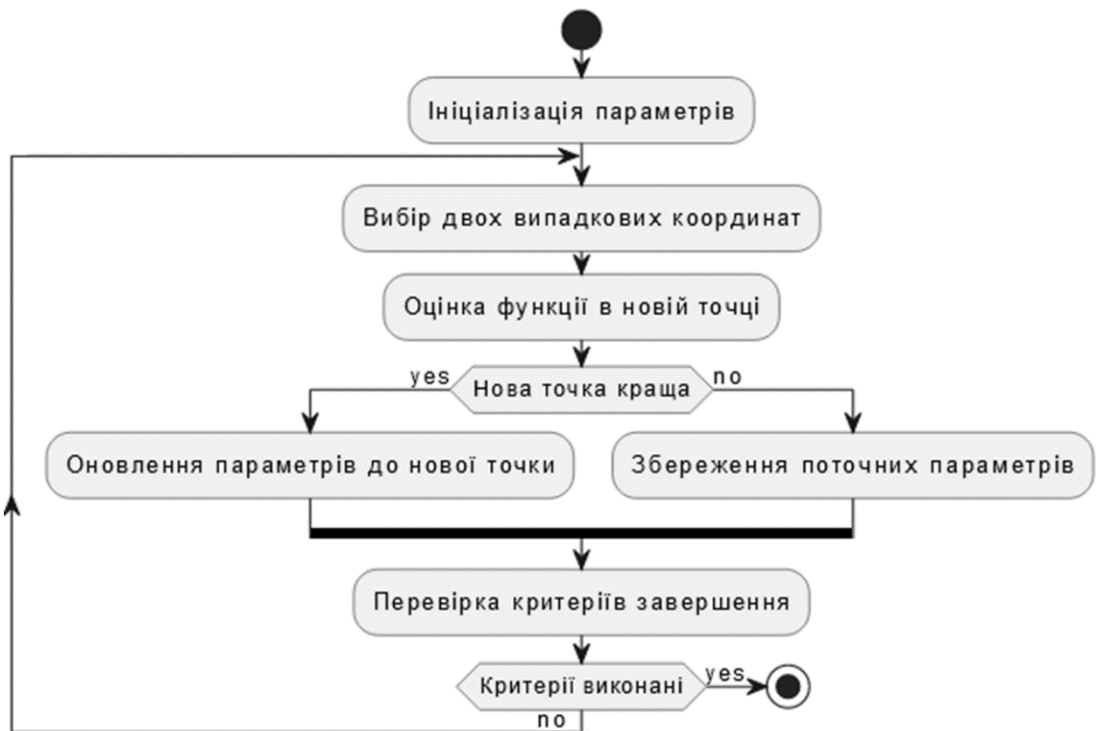


Рисунок В.1 – Алгоритм роботи методу СДС

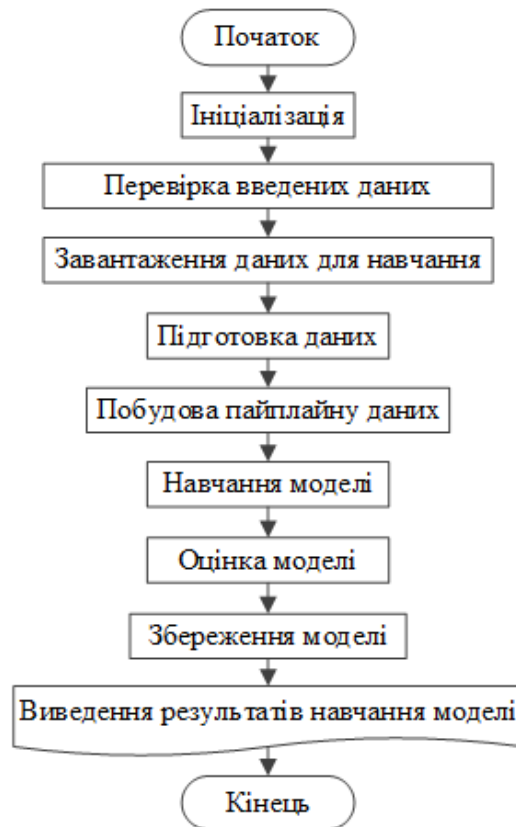


Рисунок В.2 – Алгоритм процесу тренування моделі

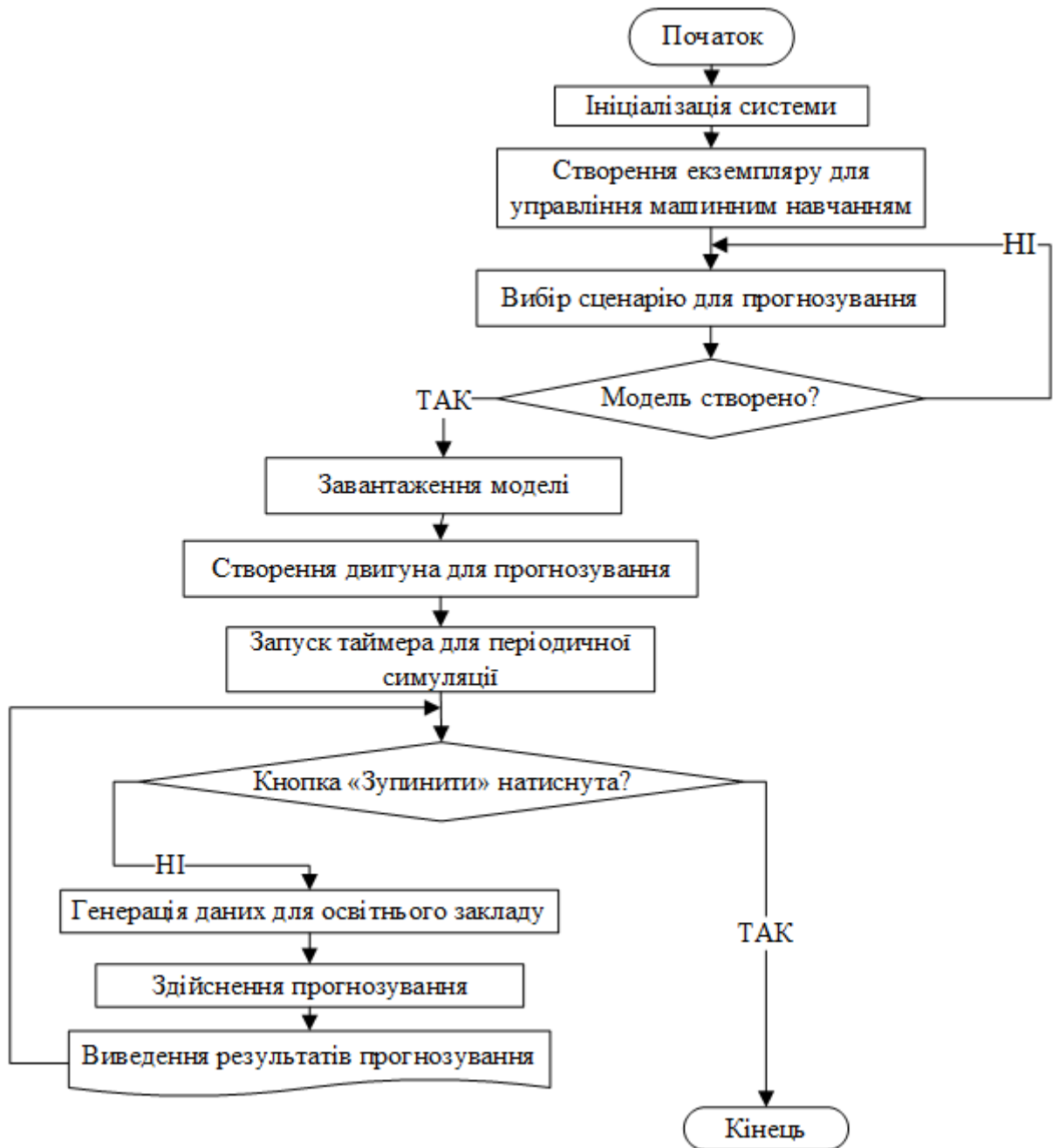


Рисунок В.3 – Алгоритм процесу прогнозування

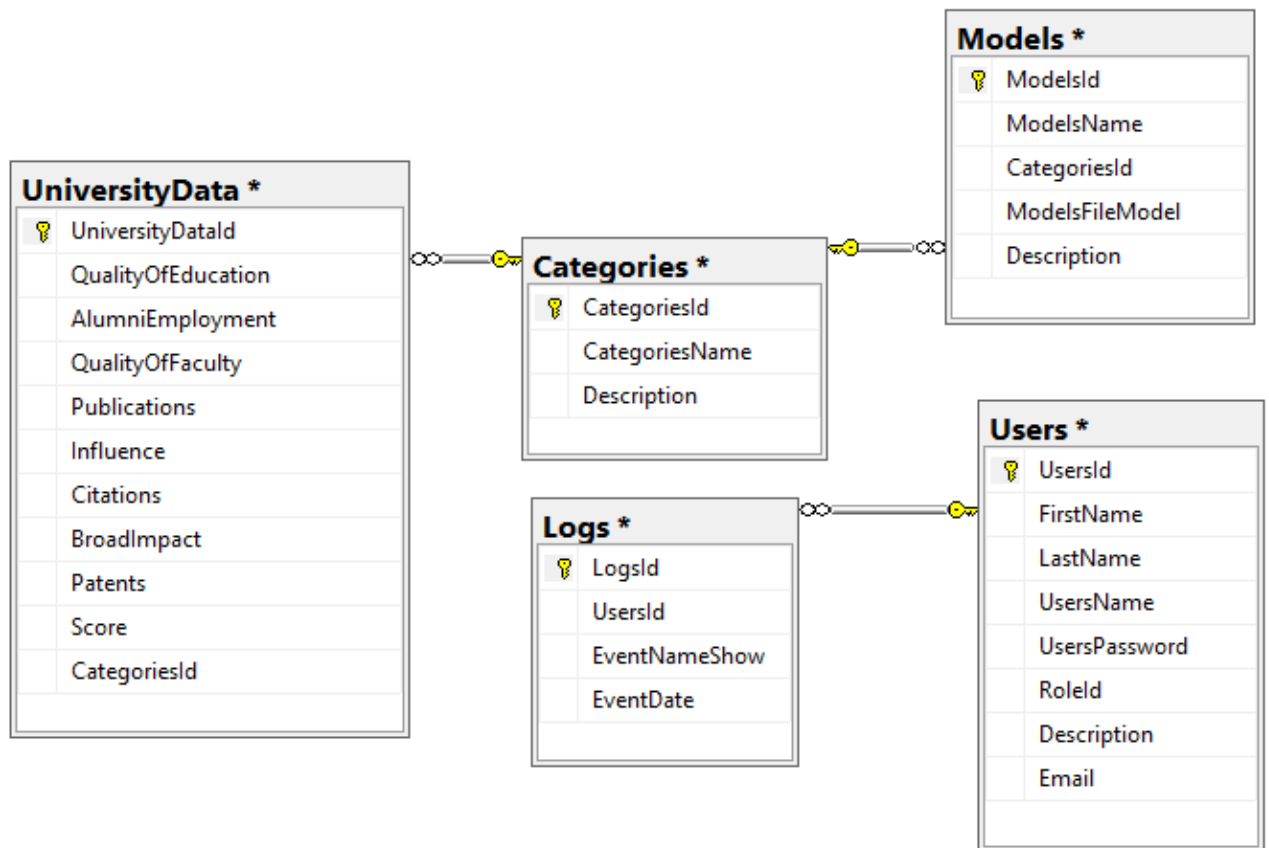


Рисунок В.4 – Фізична модель бази даних

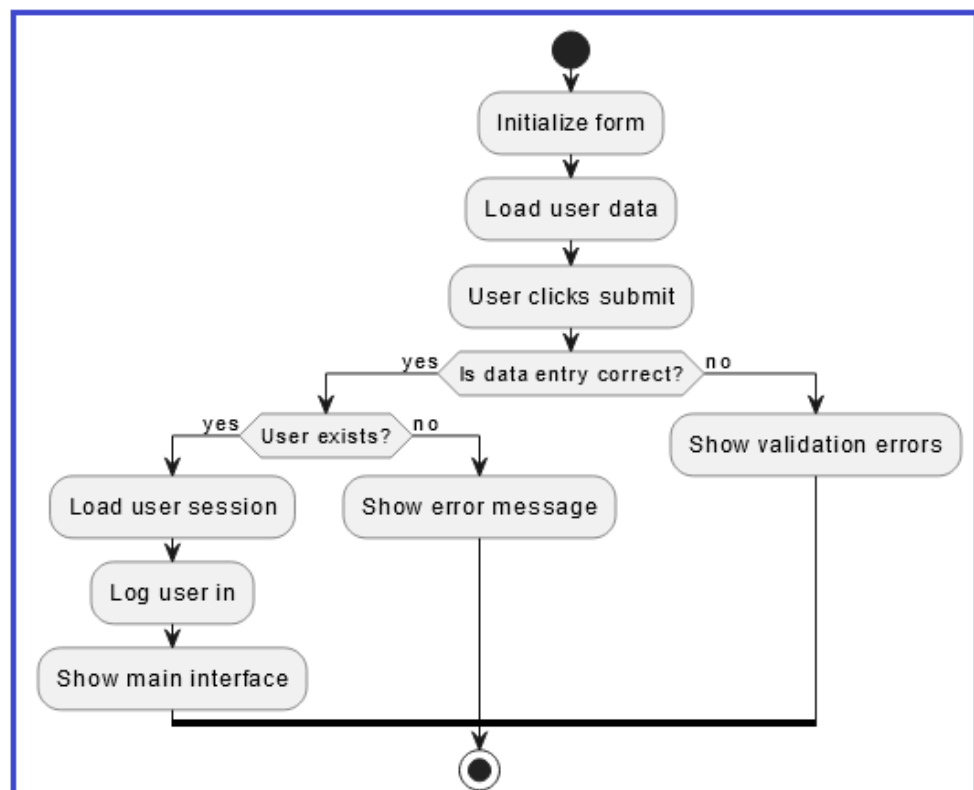


Рисунок В.5 – Процес авторизації користувача

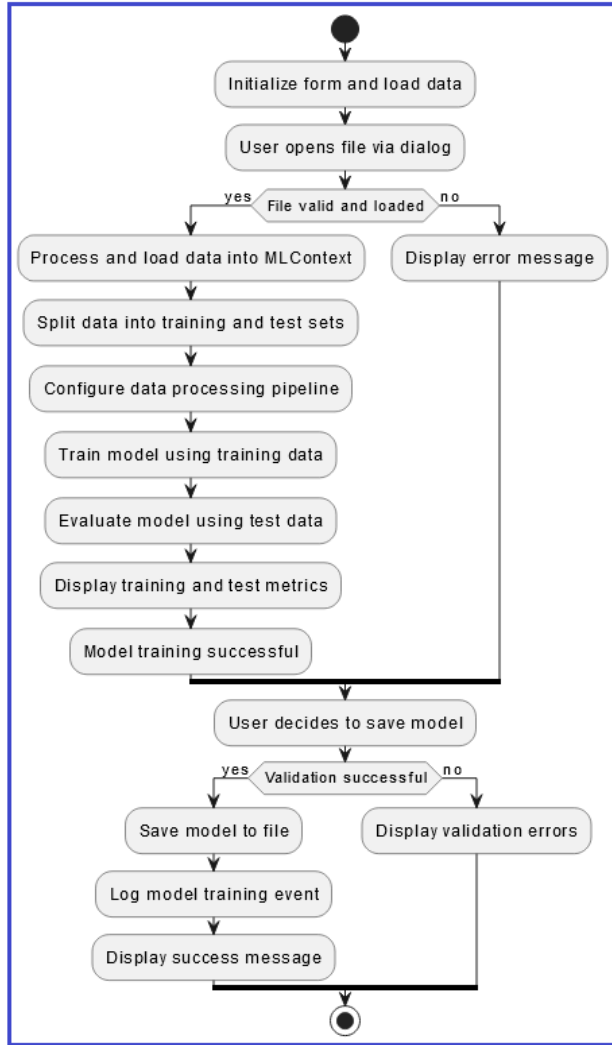


Рисунок В.6 – Процес навчання та додавання моделі в систему

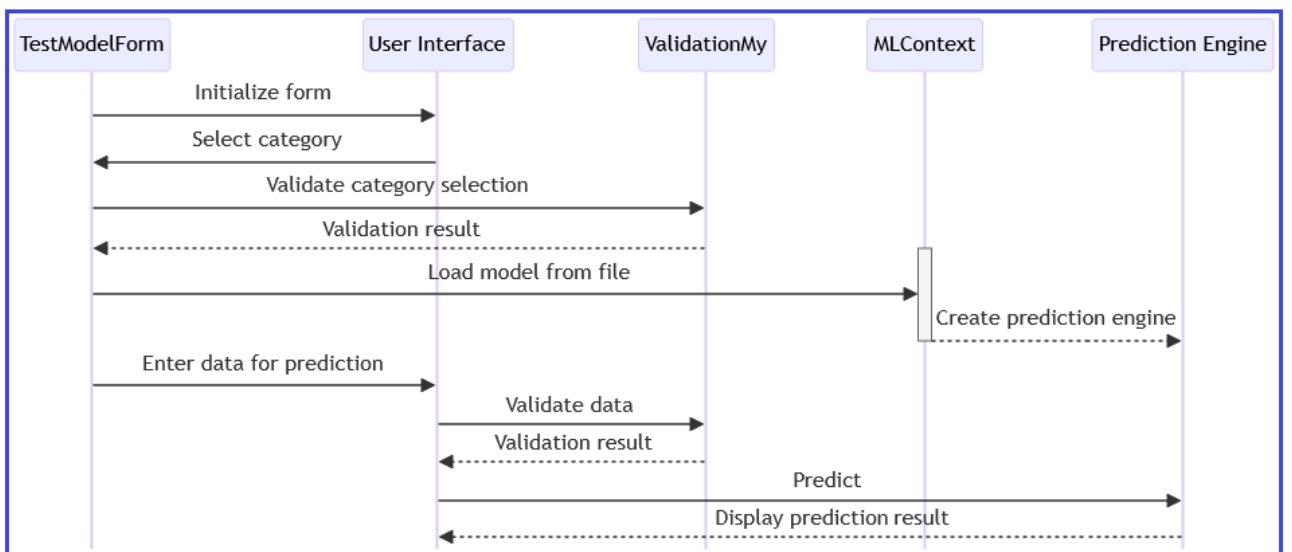


Рисунок В.7 – Процес генерації даних та прогнозування загального балу