

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Прикладних інформаційних технологій та електроінженерії

(назва факультету)

Комп'ютерно-інтегрованих технологій

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Дослідження моделей та засобів аналізування банківських пропозицій з використанням штучних нейронних мереж.

Виконав: студент (ка) 6 курсу, групи КТМ-61
спеціальності (напрямку підготовки) 174

Автоматизація та комп'ютерно-інтегровані технології

(шифр і назва спеціальності)

Конотон М.Є.
(підпис) (прізвище та ініціали)

Керівник Митник М.М.
(підпис) (прізвище та ініціали)

Нормоконтроль Левицький В.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Микитишин А.Г.
(підпис) (прізвище та ініціали)

Рецензент Марущак П.
(підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра комп'ютерно-інтегрованих технологій
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Микитишин

А.Г

(підпис)

(прізвище та ініціали)

« »

20__ р.

З А В Д А Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

(назва освітнього ступеня)

за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології»

(шифр і назва спеціальності)

студенту Конотоп Максим Євгенійович

(прізвище, ім'я, по батькові)

1. Тема роботи **Дослідження моделей та засобів аналізування банківських пропозицій з використанням штучних нейронних мереж.**

Керівник роботи доцент кафедри КТ Митник Микола Мирославович

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

Затверджені наказом ректора від « 29 » жовтня 2024 року № 4/7-1030

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

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

[illegible][illegible]

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

У даній магістерській кваліфікаційній роботі було визначено та проаналізовано проблему класифікації моделей аналізу банківських пропозицій за допомогою штучних нейронних мереж.

Мета створення даного дослідження є розробка моделі інформаційної системи для аналізування банківських пропозицій за допомогою ШНМ, та проведення дослідження – виявлення доцільності створення такого продукту, котрий допоможе користувачам приймати більш зважені рішення щодо майбутніх інвестицій, або отримання кредитування. Як результат було проаналізовано чотири різних моделей штучних нейронних мереж для аналізування банківських пропозицій. Було використано такі технології:

- 1) Користувацький сайт розроблений за допомогою: Blazor Server, Razor.
- 2) Серверна частина: ASP.NET Core 8, Entity Framework 8, Azure Cosmos SDK.
- 3) Модуль нейронної мережі для аналізу: ML.NET. В котрому експериментально було визначено найефективніший алгоритм для навчання нейронної мережі.
- 4) Модуль парсингу даних по пропозиціям: Використовує HtmlAgilityPack для веб-скрапінгу банківських пропозицій з ресурсу BANKIUA.

Дана робота складається з переліку скорочень, умовних позначень, символів, одиниць і термінів, вступу, 5 розділів, висновку, списку використаних джерел та додатків коду. Загальний обсяг роботи – 69 сторінок, робота ілюстрована 28 рисунками, а список проаналізованих літературних джерел містить 15 пунктів.

Ключові слова: нейронна мережа, аналіз банківських пропозицій, модульна система, парсинг, порівняння ефективності нейронних мереж.

ЗМІСТ

АНОТАЦІЯ	4
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	8
ВСТУП	9
1. АНАЛІТИЧНА ЧАСТИНА	11
1.1. Аналіз стану питання за літературними та іншими джерелами... 11	11
1.1.1. Історичний огляд розвитку досліджень у галузі застосування ШІ в банківській сфері.....	11
1.1.2. Аналіз існуючих підходів до персоналізації фінансових послуг 11	11
1.2. Актуальність виконання роботи	13
1.3. Методи вирішення поставленої задачі	14
1.4. Висновки та постановка задач на кваліфікаційну роботу магістра 14	14
2. ТЕХНОЛОГІЧНА ЧАСТИНА.....	17
2.1. Підготовка даних	17
2.1.1. Збір даних	17
2.1.2. Очищення даних	20
2.1.3. Формування навчального та тестового наборів	20
2.2. Алгоритми моделей	21
2.2.1. Алгоритм градієнтного бустингу (LightGbm).....	21
2.2.2. Алгоритм швидкі дерева (Fast tree)	22
2.2.3. Алгоритм швидкий ліс (Fast forest)	23
2.2.4. Алгоритм оптимізації з обмеженою пам'яттю (L-BFGS).....	24
2.3. Оцінка моделей.....	25
3. КОНСТРУКТОРСЬКА ЧАСТИНА.....	29
3.1. Архітектура системи	29
3.1.1. Загальна архітектура системи	29
3.1.2. Діаграма послідовностей.....	32
3.1.3. Діаграма станів	34
3.1.4. Схема бази даних.	36
3.2. Розробка програмного забезпечення.....	36
Інструкції програмісту	36
3.2.1. Встановлення IDE	37

3.2.2.	Завантаження проекту.....	39
3.2.3.	Проекти у контейнерах.....	41
3.3.	Інформаційна взаємодія	42
3.4.	Розгортання та масштабування.....	43
3.4.1.	Розуміння контейнерної архітектури	43
3.4.2.	Варіанти розгортання контейнерів в Azure	43
3.4.3.	Процес розгортання	44
3.4.5.	Переваги контейнерного підходу	45
4.	НАУКОВО-ДОСЛІДНА ЧАСТИНА	46
4.1.	Експерименти та результати	46
4.1.1.	Підготовка експериментальних даних.....	46
4.1.2.	Експеримент з алгоритмом L-BFGS.....	48
4.1.2.1.	Навчання на дата сеті з довжиною 50 стрічок.....	48
4.1.2.2.	Навчання на дата сеті з довжиною 100 стрічок.....	50
4.1.3.	Експеримент з алгоритмом Fast Forest	50
4.1.3.1.	Навчання на дата сеті з довжиною 50 стрічок.....	50
4.1.3.2.	Навчання на дата сеті з довжиною 100 стрічок.....	52
4.1.4.	Експеримент з алгоритмом Fast Tree	52
4.1.4.1.	Навчання на дата сеті з довжиною 50 стрічок.....	52
4.1.4.2.	Навчання на дата сеті з довжиною 100 стрічок.....	54
4.1.5.	Експеримент з алгоритмом LightGbm	55
4.1.5.1.	Навчання на дата сеті з довжиною 50 стрічок.....	55
4.1.5.2.	Навчання на дата сеті з довжиною 100 стрічок.....	56
4.1.6.	Порівняння результатів роботи алгоритмів.....	57
5.	СПЕЦІАЛЬНА ЧАСТИНА.....	60
5.1.	Формулювання наукової проблеми	60
5.1.1.	Мета дослідження	60
5.1.2.	Завдання дослідження.....	60
5.2.	Методологія дослідження	61
5.2.1.	Вибір алгоритмів	61
5.2.2.	Підготовка даних	61
5.2.3.	Налаштування гіперпараметрів:	62
5.2.4.	Оцінка моделей	62
5.2.5.	Порівняння моделей.....	62
5.2.6.	Обмеження дослідження	62
6.	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	

.....	63
6.1. Цифрова безпека при роботі з банківськими даними	63
6.2. Психологічні аспекти роботи з великими даними	64
6.3. Безпека роботи за комп'ютером	64
ВИСНОВКИ.....	66
ПЕРЕЛІК ПОСИЛАНЬ	67
ДОДАТОК А	69

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ШІ – Штучний Інтелект (Artificial Intelligence, AI)

ШНМ – Штучні Нейронні Мережі (Artificial Neural Networks, ANN)

L-BFGS – Полегшений алгоритм Бройдена – Флетчера – Гольдфарба – Шанно (Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm) - це алгоритм оптимізації, часто використовується для тренування нейронних мереж.

ШД – Швидкі Древа (Random Forest) - алгоритм машинного навчання, який будує множину дерев рішень і об'єднує їх результати для підвищення точності.

ШЛ – Швидкий Ліс (Random Forest) - синонім до ШД.

СКВ – Середнє Квадратичне Відхилення (Root Mean Square Error, RMSE) - міра середньої різниці між прогнозованими і фактичними значеннями.

API – Application Programming Interface (Інтерфейс Програмування Додатків) - набір правил і специфікацій, що дозволяє різним програмним продуктам взаємодіяти між собою.

HTML - HyperText Markup Language (Мова розмітки гіпертексту) - стандартна мова розмітки для створення веб-сторінок.

AKS - Azure Kubernetes Service - керована служба контейнеризації в хмарі Azure, яка дозволяє легко розгорнути і масштабувати контейнерні додатки.

IDE - Integrated Development Environment (Інтегроване середовище розробки) - програмне забезпечення, яке надає комплекс інструментів для розробки програмного забезпечення, таких як редактор коду, компілятор, налагоджувач.

ВСТУП

Швидкий розвиток фінансових технологій та зростаюча кількість банківських продуктів створюють для споживачів все більші труднощі у виборі оптимальних фінансових рішень. Різноманітність умов кредитів, депозитів, інвестиційних продуктів, а також постійні зміни на фінансовому ринку вимагають від споживачів все більше часу та зусиль для порівняння та аналізу пропозицій. Традиційні методи порівняння, такі як пошук інформації на веб-сайтах банків або звернення до фінансових консультантів, часто є трудомісткими та не завжди забезпечують повну картину.

Використання ШІ, зокрема, нейронних мереж, відкриває нові можливості для вирішення цієї проблеми. ШНМ здатні обробляти великі обсяги даних, виявляти складні залежності та надавати персоналізовані рекомендації. Це дозволить кожному споживачу отримати доступ до інструменту, який допоможе швидко та ефективно знайти найкращий фінансовий продукт, що відповідає його індивідуальним потребам та фінансовим можливостям.

Мета дослідження

Метою даного дослідження є розробка та оцінка ефективності моделей штучних нейронних мереж для аналізу банківських пропозицій. Конкретні завдання дослідження включають:

- Огляд існуючих підходів до застосування ШНМ в банківській сфері.
- Розробка моделей ШНМ для класифікації та регресії банківських продуктів.
- Оцінка точності та стійкості розроблених моделей на реальних даних.
- Порівняння ефективності різних типів ШНМ для вирішення поставлених задач.

- Розробка рекомендацій щодо практичного застосування результатів дослідження.

Об'єкт та предмет дослідження

- **Об'єкт дослідження:** Банківські пропозиції, моделі та засоби аналізування банківських пропозицій.
- **Предмет дослідження:** Існуючі та розроблена модель ШНМ, що використовуються для аналізу та класифікації банківських пропозицій.

1. АНАЛІТИЧНА ЧАСТИНА

1.1. Аналіз стану питання за літературними та іншими джерелами

1.1.1. Історичний огляд розвитку досліджень у галузі застосування ШІ в банківській сфері

Застосування штучного інтелекту в банківській сфері пройшло значний шлях від ранніх експериментів з експертними системами до сучасних рішень на основі глибокого навчання.

- **Ранні етапи (1950-ті – 1980-ті):** Перші спроби автоматизації простих операцій. Обмеження через обчислювальні ресурси та відсутність великих даних.
- **Ера експертних систем (1980-ті – 1990-ті):** Застосування експертних систем для оцінки кредитних ризиків та виявлення шахрайства. Обмеження гнучкості та нездатність до навчання.
- **Поява нейронних мереж (1990-ті – 2000-ті):** Відродження нейронних мереж. Застосування для прогнозування, виявлення аномалій та оцінки ризиків.
- **Епоха великих даних і глибокого навчання (2010-ті – дотепер):** Широке використання глибоких нейронних мереж для персоналізації послуг, автоматизації процесів та виявлення шахрайства.

Технології ШІ стають невід'ємною частиною банківської діяльності, і застосовуються для оцінки кредитних ризиків, персоналізації послуг, автоматизації процесів та виявлення шахрайства.

1.1.2. Аналіз існуючих підходів до персоналізації фінансових послуг

Багато досліджень в галузі персоналізації фінансових послуг зосереджені на використанні різних алгоритмів машинного навчання, включаючи нейронні мережі. Однак, існують певні обмеження та виклики:

- **Обмежена кількість публікацій:** Незважаючи на зростаючий інтерес, кількість публікацій, що детально описують моделі та методи аналізу банківських пропозицій саме з використанням нейронних мереж, є відносно невеликою або їх взагалі немає.
- **Орієнтованість на банки:** Існуючі ШНМ орієнтовані на аналізуванні даних про клієнт, щодо доцільності надання йому тієї чи іншої послуги. Тому актуальною є задача, рішення якої дозволить інвертувати модель взаємодії ШНМ з банком, до клієнта з ШНМ. Це надає можливість будь-якій зацікавленій особі проаналізувати доцільність отримання послуги банківської установи, із врахування потреб і можливостей користувача.
- **Різноманітність архітектур:** Існує велика різноманітність нейронних мереж (конволюційні, рекурентні, трансформаторні тощо), і вибір оптимальної архітектури для конкретної задачі залишається відкритим питанням.
- **Якість даних:** Ефективність нейронних мереж сильно залежить від якості та кількості даних. Отримання великих і репрезентативних наборів даних про банківські пропозиції може бути складним завданням, та потребувати багато ресурсів, як машинних так і людських для навчання нейронної мережі.

На сьогоднішній день, як прототип такої системи було використано платформу ML.NET[1], для створення та навчання двох ШНМ із алгоритмом LightGbm[2] яка включає в себе навчання зі вчителем для аналізування депозитів та кредитів. Ця нейронна мережа включала в себе такі характеристики:

- Чотири листка на одне дерево
- Чотири ітерації
- Десять секунд для навчання
- Чотири дерева

Як результат така мережа надавала оцінку на аналіз із точністю 40%, що є недостатнім результатом

Отже нова модель повинна

- **Забезпечити вищу прогностичну точність** у підборі найкращих пропозицій для клієнтів.
- **Мати кращу стійкість** до зміни та кількості наданих користувачем даних.

1.2. Актуальність виконання роботи

Банківська система відіграє фундаментальну роль у сучасній економіці, забезпечуючи ефективний перерозподіл фінансових ресурсів. Саме через банки відбувається рух грошей, що стимулює економічне зростання та розвиток, як на державному, так і на індивідуальному рівнях. Сьогоднішні банки пропонують широкий спектр фінансових послуг, від простих депозитів до складних інвестиційних продуктів. Однак, розуміння нюансів цих послуг, особливо таких як кредитування, інвестиційний банкінг та управління ризиками, вимагає глибоких знань.

Фінансова система є взаємопов'язаною з багатьма іншими сферами життя і, в свою чергу, зазнає їхнього впливу. Тому, щоб приймати обґрунтовані рішення в сфері інвестування, необхідно мати комплексне розуміння механізмів функціонування банківської системи.

Аналіз банківських продуктів та оцінка їхніх переваг і недоліків є складним завданням, яке потребує застосування різноманітних методів і інструментів. Одним з перспективних підходів є використання штучного інтелекту, зокрема, штучних нейронних мереж (ШНМ). ШНМ здатні аналізувати великі обсяги даних, виявляти приховані закономірності та робити прогнози. Застосування ШНМ в банківській сфері дозволяє підвищити точність прогнозування, оптимізувати процеси прийняття рішень та, як наслідок, покращити фінансове благополуччя як окремих громадян, так

і економіки в цілому.

1.3. Методи вирішення поставленої задачі

Для вирішення поставленої задачі буде використано наступні підходи:

- 1. Збір даних:** Дані про банківські пропозиції будуть зібрані з відкритих джерел та баз даних банків. Буде проведена ретельна очистка даних від пропусків та аномалій, а також нормалізація чисельних ознак.
- 2. Вибір моделі:** Для вибору моделі проаналізуємо існуючі алгоритми ШНМ, які застосовуються для задач класифікації. Є декілька алгоритмів такі як СПКП, ШД, ШЛ, ОЗОП.
- 3. Формування даних для навчання:** Для формування навчання, потрібно вручну зробити аналіз пропозицій, та позначити відповідне значення для пропозиції, котра буде використовуватись при навчанні.
- 4. Процес навчання:** Під час навчання буде використано декілька алгоритмів навчання, також будуть змінені коефіцієнти навчання, кількості ітерацій та різні оптимізаційні алгоритми такі, як середнє квадратичне відхилення, Р-квадратичне та похибка середнього квадрату.
- 5. Інтерпретація результатів:** Після отриманих результатів, результати різних алгоритмів будуть порівняні, та на основі таких параметрів, як швидкість, точність та кількість ресурсів необхідних на обчислення, буде обрано найкращий алгоритм.

1.4. Висновки та постановка задач на кваліфікаційну роботу магістра

Проведений аналіз сучасного стану досліджень у галузі застосування штучного інтелекту в банківській сфері дозволяє зробити такі висновки:

- 1. Актуальність проблеми:** Швидкий розвиток фінансових технологій та зростання кількості банківських продуктів створюють для споживачів все більші труднощі у виборі оптимальних фінансових рішень.
- 2. Перспективність застосування ШНМ:** Штучні нейронні мережі демонструють високу ефективність в аналізі великих обсягів даних та виявленні складних залежностей, що робить їх перспективним інструментом для персоналізації фінансових послуг.
- 3. Недостатня розробленість:** Незважаючи на значний потенціал, застосування ШНМ для аналізу банківських пропозицій та надання персоналізованих рекомендацій є відносно новим напрямком досліджень, і існує багато невирішених питань.

Постановка задач

З урахуванням вищевикладеного, метою кваліфікаційної роботи є розробка та оцінка ефективності моделі штучного інтелекту, яка б автоматизувала процес аналізу банківських пропозицій та надавала користувачам персоналізовані рекомендації. Для досягнення цієї мети необхідно вирішити наступні завдання:

- 1. Збір та підготовка даних:** Сформувати репрезентативну вибірку даних про різноманітні банківські продукти, включаючи їх характеристики, умови та тарифи, та дані користувача.
- 2. Розробка архітектури моделі ШНМ:** Вибрати оптимальну архітектуру нейронної мережі, враховуючи специфіку задачі та наявні дані.
- 3. Навчання моделі:** Навчити модель ШНМ на підготовлених даних з метою класифікації та ранжування банківських продуктів на основі даних користувача.

4. **Оцінка ефективності моделі:** Провести експериментальну оцінку точності та релевантності рекомендацій, що надаються моделлю.
5. **Розробка інтерфейсу:** Створити зручний та інтуїтивно зрозумілий інтерфейс для взаємодії користувача з системою.
6. **Аналіз результатів та формулювання висновків:** Проаналізувати отримані результати, оцінити ефективність розробленої моделі та сформулювати рекомендації щодо її подальшого розвитку.

Очікувані результати:

- Розроблена модель ШНМ, що здатна ефективно аналізувати банківські пропозиції та надавати персоналізовані рекомендації користувачам.
- Прототип системи, який демонструє працездатність розробленої моделі.
- Наукова публікація, що відображає результати дослідження.

2. ТЕХНОЛОГІЧНА ЧАСТИНА

2.1. Підготовка даних

Підготовка даних є базовою та обов'язковою частиною, як для навчання ШНМ, так і для використання користувачем. Процедура підготовки даних передбачає такі етапи: збір даних (наприклад з існуючої бази даних); очищення даних на підставі встановлених критеріїв; формування навчальних та тестових наборів.

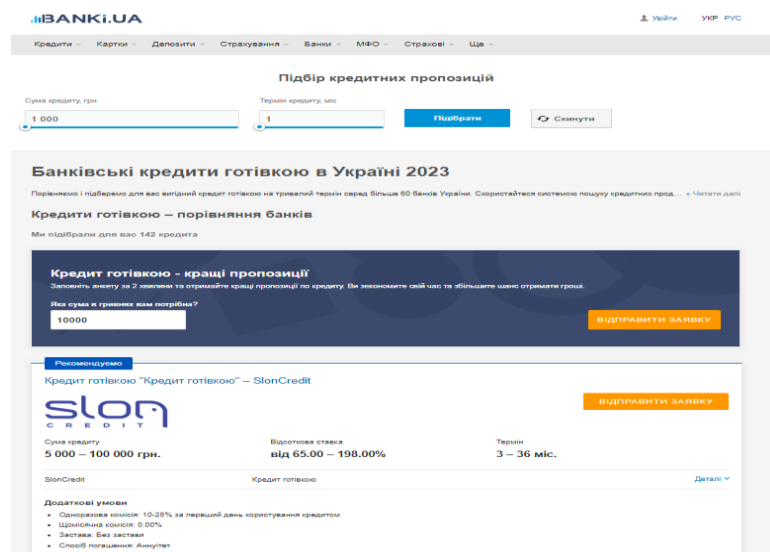
2.1.1. Збір даних

Збір даних – це перший і найважливіший етап будь-якого дослідження, включаючи дослідження моделей та засобів аналізування банківських пропозицій за допомогою штучних нейронних мереж. Це процес систематичного накопичення інформації, яка буде використана для навчання моделі та оцінювання її точності.

Збором даних займатиметься окремий компонент системи, котрий буде отримувати актуальну інформацію за пропозиціям з таких джерел:

Джерела даних:

1. Banki.ua [4]



The screenshot displays the Banki.ua website interface. At the top, there is a navigation bar with links for 'Кредити', 'Картки', 'Депозити', 'Страхування', 'Банки', 'МФО', 'Страхові', and 'Ще'. Below this, a section titled 'Підбір кредитних пропозицій' features input fields for 'Сума кредиту, грн' (set to 1 000) and 'Термін кредиту, міс' (set to 1), with buttons for 'Підбрати' and 'Скинути'. The main content area is titled 'Банківські кредити готівкою в Україні 2023' and includes a sub-header 'Кредити готівкою – порівняння банків'. A prominent blue box highlights 'Кредит готівкою - кращі пропозиції' with a 'ВІДПРАВИТИ ЗАПИТ' button. Below this, a 'Рекомендуємо' section features a card for 'Кредит готівкою "Кредит готівкою" – SlonCredit', showing a loan amount of 5 000 – 100 000 грн, an interest rate of 65.00% – 198.00%, and a term of 3 – 36 months. A 'ВІДПРАВИТИ ЗАПИТ' button is also present. The bottom of the card lists 'Додаткові умови' such as 'Санкційна комісія: 10-25% за перший день користування кредитом' and 'Цілісний екшн: 0.00%'. A 'Деталі' link is located at the bottom right of the card.

Рис.2.1.1.1. Відображення сторінки пропозицій **Banki.ua**.

Цей онлайн-ресурс призначений для комплексного аналізу та порівняння фінансових продуктів. Платформа надає користувачам зручний інструмент для самостійного дослідження ринку кредитів, депозитів та інших банківських послуг. Завдяки детальній інформації про кожну пропозицію, включаючи відсоткові ставки, комісії та додаткові умови, користувачі можуть приймати обґрунтовані рішення, обираючи найвигідніші варіанти.

Однією з відмінних рис платформи є наявність експертних оцінок, які допомагають користувачам краще зрозуміти особливості кожного продукту. Однак, деталі методики оцінювання, а також ступінь незалежності експертів не розкриваються. Хоча платформа заявляє про використання даних з офіційних джерел, зокрема Національного банку України, це питання залишається предметом для додаткового дослідження.

Додатковими функціональними можливостями платформи є наявність графіків динаміки ставок, а також відгуків користувачів, які можуть поділитися своїм досвідом взаємодії з різними банками. Це дозволяє отримати більш повну картину про ринок фінансових послуг та обрати найнадійнішого партнера.

2. Minfin.com.ua [5]

The screenshot displays the Minfin.com.ua website interface. At the top, there is a navigation bar with the Minfin logo and a 'Вхід' (Login) button. Below the navigation bar, a horizontal menu lists various financial categories: Валюта, Депозити (highlighted), Кредити, Банки, Страхування, Інвестиції, Криптовалюта, and Індекси. A secondary menu below it includes links like Бонус до депозитів, Підтвердження депозиту, Огляд ставок, and others. The main section is titled 'Депозити в банках України' (Deposits in Ukrainian banks) with a subtitle 'Порівняйте ставки по депозитам в гривні, доларі та євро. Тільки найкращі банківські вклади України.' (Compare interest rates on deposits in Ukrainian hryvnia, dollars, and euros. Only the best bank deposits in Ukraine). The search filters include 'Сума / валюта' (Amount / currency) set to '100 000' and 'гривня', and 'Термін' (Term) set to 'Будь-який' (Any). There are buttons for 'Фільтри' (Filters) and 'Підібрати' (Select). Below the filters, there are checkboxes for 'З бонусом до депозиту' (With deposit bonus), 'Онлайн заявка' (Online application), 'Державні банки' (State banks), and 'Банки з іноземним капіталом' (Banks with foreign capital). At the bottom, there is a 'Гарячі пропозиції' (Hot offers) section with logos for TBC Bank, P Bank, and others, showing their respective interest rates. A contact box on the right asks 'Є питання?' (Have a question?) and provides a phone number '+38 (044) 392-47-40' and a 'Передзвоніть мені' (Call me back) button.

Рис.2.1.2.1. Відображення сторінки Minfin.com.ua.

Minfin.com.ua, заснований у 2009 році, протягом багатьох років позиціонує себе як провідний український ресурс для порівняння фінансових продуктів та послуг. Платформа пропонує користувачам широкий спектр інформації, що охоплює кредити, депозити, страхування та інші фінансові інструменти. Завдяки зручному інтерфейсу та детальним порівняльним таблицям, користувачі можуть швидко знайти найвигідніші пропозиції на ринку.

Однак, варто зазначити, що на об'єктивність інформації, представленої на платформі, може впливати ряд факторів. Зокрема, значна залежність Minfin.com.ua від рекламних доходів створює потенційний конфлікт інтересів. Це може призводити до того, що фінансові установи, які розміщують рекламу на платформі, отримують більш сприятливе висвітлення. Крім того, наявність спеціальних пропозицій та бонусів для певних банків може створювати враження упередженості ресурсу щодо цих фінансових установ.

Хоча Minfin.com.ua є корисним інструментом для порівняння фінансових продуктів, користувачам рекомендується підходити до представленої інформації з певною критичністю. Для прийняття зваженого рішення варто додатково перевіряти інформацію на офіційних веб-сайтах банків та консультуватися з фінансовими консультантами.

Технологія збору даних

Після визначення джерел даних, потрібно автоматизувати спосіб їх отримання. Оскільки доступних API дані ресурси не пропонують, потрібно застосувати парсинг вказаних джерел.

Парсинг реалізуємо за допомогою бібліотеки **HtmlAgilityPack**, яка в свою чергу надає інструменти та засоби для організації з HTML сторінки.

Компонент системи буде запущено за допомогою HTTP API раз на добу, або у разі потреби – у будь-який час.

Після зчитування сторінки, компонент системи буде створювати у системній базі даних запис пропозиції. Якщо пропозиція існує, компонент оновить дані для цієї пропозиції.

Робота компонента буде вважатися завершеною, коли всі пропозиції з ресурсу було створено або оновлено.

Формат у якому будуть зберігатись дані – SQL. Навчання мережі буде здійснюватися на основі даних, які мають такі атрибути пропозиції: ставка, сума вкладу, термін, інфляція та атрибутів користувача : вік, заробітна плата, ціль. Після формування таких вибірок, ці дані будуть попередньо класифіковані особою зі знаннями у області фінансів або економіці.

2.1.2. Очищення даних

Оскільки якість даних не контролюється системою, можуть статися випадки коли система отримає не повну інформацію про пропозицію. Виключати такі дані, ми не можемо, тому що не буде сформована повна та незалежна база даних. Проте усі пропозиції, незалежно від ресурсів, повинні мати обов’язковий базовий набір параметрів

- Ставка
- Сума вкладу
- Термін

У разі відсутності цих даних параметрів, пропозиція буде вважатися неповною, а компонент системи відповідальний за організацію збору даних – не зберігатиме таку пропозицію у системі.

2.1.3. Формування навчального та тестового наборів

Для навчання та оцінювання моделі нейронної мережі, здатної аналізувати банківські пропозиції, сформуємо два спеціалізовані набори на основі зібраних даних. Навчальний набір, що містить деталізовану інформацію про різноманітні пропозиції (кредити, депозити), буде використаний для навчання моделі з метою виявлення закономірностей та формування точних

прогнозів. Тестовий набір, який не перетинається з навчальним, служить для порівняльної об'єктивної оцінки точності та узагальнення здатності навченої моделі правильно інтерпретувати на нові, раніше не опрацьовані дані. Для розділення вибірок застосовано підхід 80/20, де 80 % це кількість даних для навчання, а 20% кількість тестових даних .

2.2. Алгоритми моделей

2.2.1. Алгоритм градієнтного бустингу (LightGbm)

LightGBM (Light Gradient Boosting Machine) - це алгоритм градієнтного бустингу, який використовується для вирішення широкого спектра задач машинного навчання, включаючи регресію, класифікацію та ранжування. Він відрізняється високою швидкістю навчання та ефективністю для великих даних.

Ключові кроки алгоритму:

1. Ініціалізація:

- Створюється первинна модель (часто це константний предиктор).
- Встановлюються гіперпараметри алгоритму (кількість дерев, глибина дерев, швидкість навчання тощо).

2. Ітеративне навчання:

- **Обчислення градієнтів:** Для кожного зразка обчислюється градієнт функції втрат відносно передбачення поточної моделі. Градієнт вказує напрямок найбільшого зростання помилки і використовується для побудови наступного дерева.
- **Побудова дерева:**
 - **Вибір розщеплення:** Для кожного вузла дерева вибирається найкраще розщеплення, яке максимізує деякий критерій розщеплення (наприклад, інформаційний виграш).
 - **Обчислення значень в листках:** Для кожного листа обчислюється значення, яке мінімізує сумарний градієнт у цьому листі.

- **Оновлення моделі:** Нове дерево додається до ансамблю з відповідним вагою.
- **Перевірка умови зупинки:** Якщо досягнуто заданої кількості дерев або іншої умови зупинки, алгоритм завершується. Інакше, переходимо до наступної ітерації.

3. Передбачення:

- Для нового зразка передбачення здійснюється шляхом проходження його через кожне дерево ансамблю і сумування значень листів з відповідними вагами.

2.2.2. Алгоритм швидкі дерева (Fast tree)

Алгоритми Швидких Дерев (Fast Tree) – це клас алгоритмів, які використовуються в машинному навчанні для побудови моделей ансамблів дерев рішень. Вони оптимізовані для швидкої та ефективної роботи з великими наборами даних.

Ключові кроки алгоритму:

Побудова гістограм:

- Для кожної ознаки будується гістограма, яка містить інформацію про розподіл значень цієї ознаки в навчальній вибірці. Гістограми дозволяють швидко оцінити якість розбиття вузла дерева за різними значеннями ознаки.

Вибір ознаки для розбиття:

- Для кожного вузла дерева випадково вибирається підмножина ознак.
- Для кожної вибраної ознаки, використовуючи гістограми, обчислюється значення критерію розбиття (наприклад, інформаційний виграш, виграш за Джині).
- Ознака з найкращим значенням критерію розбиття використовується для розділення вузла на два під вузли.

Рекурсивна побудова дерева:

- Процес розбиття повторюється для кожного під вузла до тих пір, поки не буде досягнуто критерію зупинки (наприклад, мінімальна кількість зразків у листі, максимальна глибина дерева).

Ансамбль дерев:

- Зазвичай будується не одне, а кілька дерев рішень. Ці дерева об'єднуються в ансамбль для покращення точності прогнозування.
- Порівняння з традиційними алгоритмами побудови дерев.

2.2.3. Алгоритм швидкий ліс (Fast forest)

Швидкі ліси (Fast Forest) - це ансамбль алгоритмів, який базується на ідеї побудови множини швидких дерев рішень та об'єднання їхніх прогнозів для прийняття остаточного рішення. Це розширення концепції випадкових лісів (Random Forest), де замість традиційних дерев рішень використовуються більш ефективні швидкі дерева.

Ключові кроки алгоритму:

Генерація випадкових вибірок:

- Для кожного дерева в лісі створюється випадкова вибірка з вихідного набору даних. Ця вибірка може бути з поверненням (тобто один і той самий зразок може потрапити в вибірку кілька разів) або без повернення.
- Випадкова вибірка дозволяє кожному дереву "бачити" різні частини даних, що зменшує кореляцію між деревами в ансамблі та підвищує загальну точність.

Побудова швидкого дерева:

- Для кожної випадкової вибірки будується окреме швидке дерево за алгоритмом, описаним раніше.

- Швидкі дерева використовують такі оптимізації, як гістограми та вибірка ознак, щоб прискорити процес навчання та зменшити обчислювальну складність.

Об'єднання прогнозів:

- Для нового зразка кожне дерево в лісі дає свій прогноз.
- Для задач регресії остаточний прогноз лісу обчислюється як середнє арифметичне прогнозів окремих дерев.
- Для задач класифікації остаточний прогноз визначається шляхом голосування: вибирається той клас, який отримав найбільшу кількість голосів від окремих дерев.

2.2.4. Алгоритм оптимізації з обмеженою пам'яттю (L-BFGS)

L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) – це ітеративний метод оптимізації, який використовується для знаходження локального мінімуму або максимуму нелінійної функції без обмежень. Він є особливо ефективним для задач з великою кількістю змінних, оскільки використовує обмежену кількість пам'яті.

Кроки алгоритму L-BFGS:

Ініціалізація:

- Вибирається початкова точка x_0 .
- Визначається розмір "вікна пам'яті" m , тобто кількість попередніх кроків, які будуть зберігатися.
- Створюється апроксимація матриці Гессіана (часто як скалярна матриця).

Обчислення градієнта:

- Обчислюється градієнт функції $f(x)$ в поточній точці x_k . Градієнт вказує напрямок найбільшого зростання функції.

Побудова пошукового напрямку:

- Використовуючи апроксимацію матриці Гессіана та градієнт, обчислюється пошуковий напрямок p_k . Цей напрямок вказує, в якому напрямку слід рухатися, щоб зменшити значення функції.

Вибір довжини кроку:

- Використовуючи лінійний пошук, визначається довжина кроку α_k , яка мінімізує значення функції вздовж напрямку p_k .

Оновлення точки:

- Нова точка обчислюється як $x_{k+1} = x_k + \alpha_k p_k$.

Збереження інформації про крок:

- Зберігаються вектори $s_k = x_{k+1} - x_k$ та $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. Ці вектори містять інформацію про зміну точки та градієнта за останній крок.
- Якщо кількість збережених пар (s, y) перевищує задане значення m , найстаріша пара видаляється.

Оновлення апроксимації матриці Гессіана:

- Використовуючи збережені пари (s, y) , оновлюється апроксимація матриці Гессіана.

Перевірка умови зупинки:

- Перевіряється, чи досягнуто заданої точності або максимальної кількості ітерацій. Якщо так, алгоритм зупиняється. Інакше, повертаємося до кроку 2.

2.3. Оцінка моделей

Для оцінки моделей будемо використовувати такі критерії: швидкість збігання, вимоги до пам'яті, інтерпретованість, стійкість до викидів,

налаштування параметрів, стійкість до шуму, схильність до перетримання, масштабування, придатність до різних задач та ефективність.

Критерії оцінки:

1. **Швидкість збігання (Convergence Speed):** Показник того, наскільки швидко алгоритм досягає оптимального рішення або мінімізує функцію втрат. Швидкі алгоритми можуть швидше знайти найкращі параметри моделі, зменшуючи час тренування.
2. **Пам'ять (Memory Usage):** Вимоги до оперативної пам'яті, яку потребує алгоритм під час тренування та виконання. Високі вимоги до пам'яті можуть ускладнити використання алгоритмів для великих наборів даних.
3. **Інтерпретованість (Interpretability):** Здатність алгоритму пояснювати результати або те, як він приймає рішення. Легко інтерпретовані моделі дозволяють користувачам розуміти причини прогнозів та дії.
4. **Стійкість до викидів (Outlier Robustness):** Міра того, наскільки алгоритм може ефективно працювати у присутності викидів або аномальних даних. Алгоритми, стійкі до викидів, можуть зменшити вплив цих даних на результати.
5. **Налаштування параметрів (Parameter Tuning):** Оцінює, наскільки легко налаштувати гіперпараметри алгоритму для досягнення найкращих результатів. Деякі алгоритми мають багато параметрів, що ускладнює їх оптимізацію.
6. **Стійкість до шуму (Noise Robustness):** Здатність алгоритму працювати ефективно у присутності шумових даних. Алгоритми, стійкі до шуму, можуть зменшити вплив некоректних або непотрібних даних на результати.
7. **Перетримання (Overfitting):** Визначає, наскільки алгоритм схильний до перетримання, коли модель добре працює на тренувальному наборі даних, але погано на тестовому. Алгоритми з низькою схильністю до перетримання мають кращу узагальнюваність.

8. **Масштабування (Scalability):** Здатність алгоритму ефективно працювати з великими наборами даних. Масштабовані алгоритми можуть обробляти великі обсяги даних без значного зниження продуктивності.
9. **Придатність для різних задач (Suitability for Different Tasks):** Оцінка того, наскільки добре алгоритм підходить для різних завдань, таких як класифікація, регресія тощо.
10. **Ефективність (Efficiency):** Загальна ефективність алгоритму у виконанні своїх функцій, включаючи швидкість обробки, використання ресурсів та точність.

Критерій	LightGbm	Fast Tree	Fast Forest	L-BFGS
Швидкість збігання	Висока	Висока	Середня	Висока
Пам'ять	Вимагає багато пам'яті	Вимагає багато пам'яті	Висока вимога до пам'яті	Вимагає багато пам'яті
Інтерпретованість	Складніше інтерпретувати	Легко інтерпретується	Важко інтерпретувати	Середня
Стійкість до викидів	Висока	Висока	Висока	Середня
Налаштування параметрів	Складніше налаштування параметрів	Легке налаштування параметрів	Складніше налаштування параметрів	Середня
Стійкість до шуму	Середня	Висока	Висока	Середня
Перетримання	Низька схильність до перетримання	Поширення перетримання	Низька схильність до перетримання	Середня
Масштабування	Ефективне для великих наборів даних	Потребує багато ресурсів	Ефективне для великих наборів даних	Ефективне для великих наборів даних
Придатність до різних задач	Добре підходить для завдань класифікації і регресії	Добре підходить для завдань класифікації і регресії	Добре підходить для завдань класифікації	Добре підходить для завдань класифікації і регресії

Ефективність	Ефективне, але вимагає багато пам'яті	Ефективне, але може бути схильне до перетримання	Ефективне, але вимагає багато пам'яті	Ефективне, але може застрягати в локальних мінімумах
---------------------	---------------------------------------	--	---------------------------------------	--

Таблиця 1. Оцінка алгоритмів навчання нейронної мережі.

3. КОНСТРУКТОРСЬКА ЧАСТИНА

3.1. Архітектура системи

3.1.1. Загальна архітектура системи

Загальна концепція

Система призначена для аналізу банківських пропозицій на основі даних, зібраних з різних джерел. Вона складається з трьох основних компонентів: клієнтської частини, серверної частини та зовнішніх систем.



Рис.3. Загальна архітектура системи.

Зовнішні системи

- **Джерела даних:** Система збирає дані з різних джерел, таких як Національний банк України, BANKI.UA, Мінфін.
- **Методи отримання даних:**
 - **Парсинг HTML:** Для першої версії системи передбачено збирання даних шляхом парсингу HTML-сторінок.

- **API:** В подальшому планується використовувати API для отримання даних від ресурсів, які надають таку можливість.
- **Технології:** Для парсингу HTML буде використано бібліотеку Html Agility Pack.

Серверна частина

- **Модуль отримання даних:** Відповідає за збір даних з зовнішніх джерел за допомогою заданих методів.
- **Модуль нейронної мережі:**
 - **Аналіз:** Проводиться аналіз даних за допомогою відповідних алгоритмів. Деталі алгоритмів не наведені на схемі.
- **Модуль бізнес-логіки:** Реалізує основну логіку системи, таку як:
 - Збереження та відображення результатів аналізу.
 - Збереження та відображення пропозицій.
 - Фільтрування пропозицій.
 - Збереження даних користувачів.
 - Пошук по параметрах.
- **Технології:** Для реалізації серверної частини використовується технологія .NET та фреймворк Blazor.
- **База даних:** Для зберігання даних використовується NoSQL CosmosDb та SQL MSSQL бази даних.

Клієнтська WEB частина

- **Взаємодія з користувачем:** Користувач вводить свої дані (вік, дохід, ціль) та отримує персоналізовані рекомендації.

- **Технології:** Для створення інтерфейсу використовується фреймворк Blazor.

Загальний процес роботи системи

1. Користувач вводить свої дані на клієнтській частині.
2. Клієнтська частина відправляє запит на сервер.
3. Серверна частина збирає необхідні дані з зовнішніх джерел.
4. Зібрані дані обробляються та аналізуються.
5. Результати аналізу зберігаються в базі даних та відображаються користувачу.

Переваги такої архітектури

- **Модульність:** Кожен компонент системи відповідає за свою функцію, що полегшує розробку та підтримку.
- **Гнучкість:** Система може бути легко адаптована до нових джерел даних та зміни вимог користувачів.
- **Масштабованість:** За рахунок використання мікросервісної архітектури систему можна легко масштабувати для обробки великих обсягів даних.

3.1.2. Діаграма послідовностей

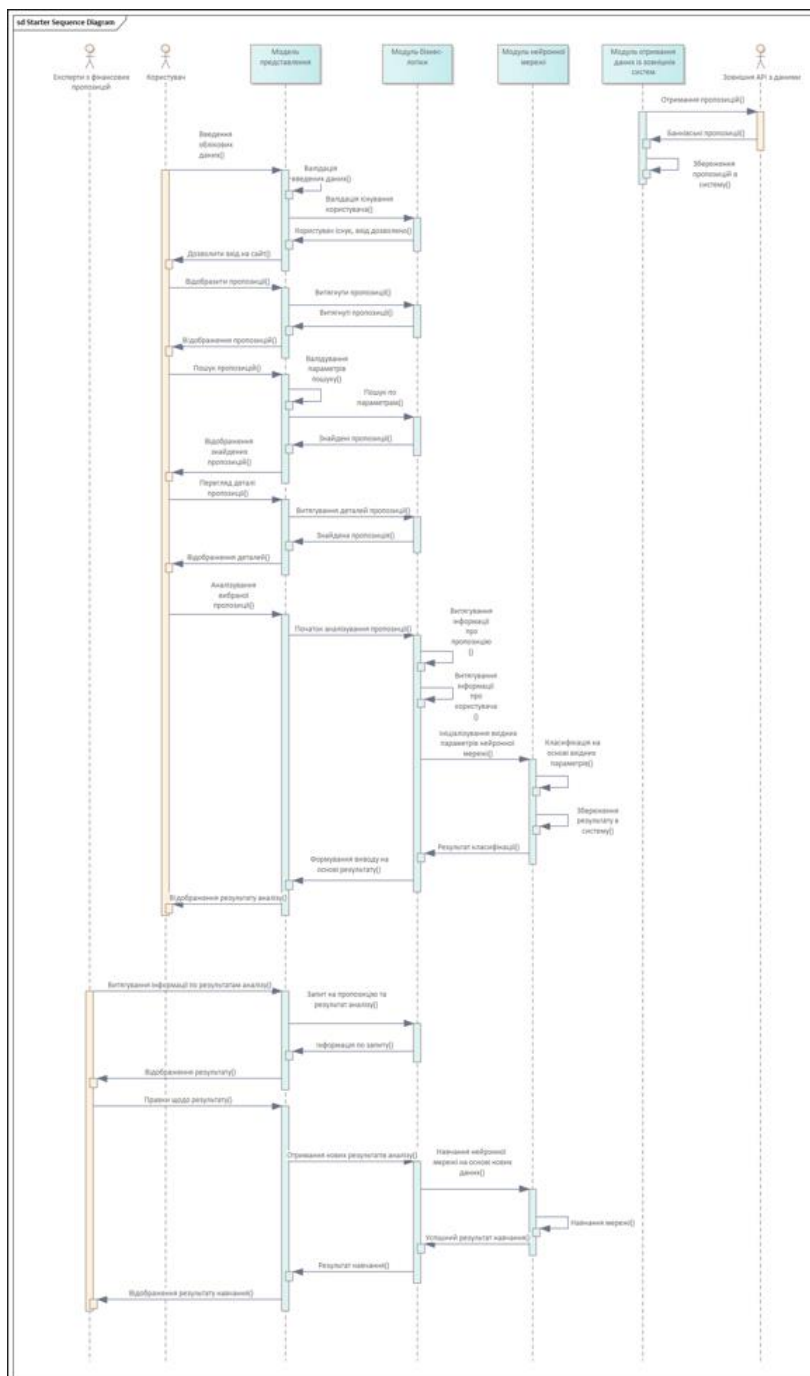


Рис.4. Загальна діаграма послідовностей.

Основний алгоритм роботи системи:

1. **Формування запити:** Користувач вводить у систему конкретні вимоги для пошуку, задаючи певні параметри.
2. **Пошук відповідностей:** Система аналізує введені користувачем дані та здійснює пошук серед наявних пропозицій, які відповідають заданим критеріям.

3. **Виведення результатів:** Система відображає користувачеві список знайдених варіантів, відсортованих за релевантністю.
4. **Оцінка результатів:** Користувач переглядає отримані результати та обирає оптимальний варіант, що найбільш відповідає його потребам.

Альтернативний сценарій: Якщо система не знаходить жодного варіанту, який би задовольняв усі вказані користувачем умови, вона інформує про відсутність відповідних результатів.

Взаємодія компонентів системи:

- **Користувач:** Ініціатор взаємодії, який вводить початкові дані для пошуку.
- **Система:** Комплекс програмних засобів, що забезпечує обробку запитів користувача та видачу результатів.
- **База даних:** Сховище інформації, в якому зберігаються всі доступні пропозиції.

Детальний опис процесу:

- **Введення даних:** Користувач вводить у спеціальне поле інтерфейсу системи конкретні параметри пошуку, такі як тип пропозиції, ціновий діапазон, додаткові характеристики тощо.
- **Обробка запиту:** Система отримує введений користувачем запит, аналізує його та формує відповідний запит до бази даних.
- **Пошук в базі даних:** Система здійснює пошук у базі даних, використовуючи ефективні алгоритми для швидкого знаходження відповідних даних.
- **Формування результатів:** Знайдені результати сортуються за релевантністю та зручно відображаються користувачеві у вигляді списку або таблиці.

3.1.3. Діаграма станів

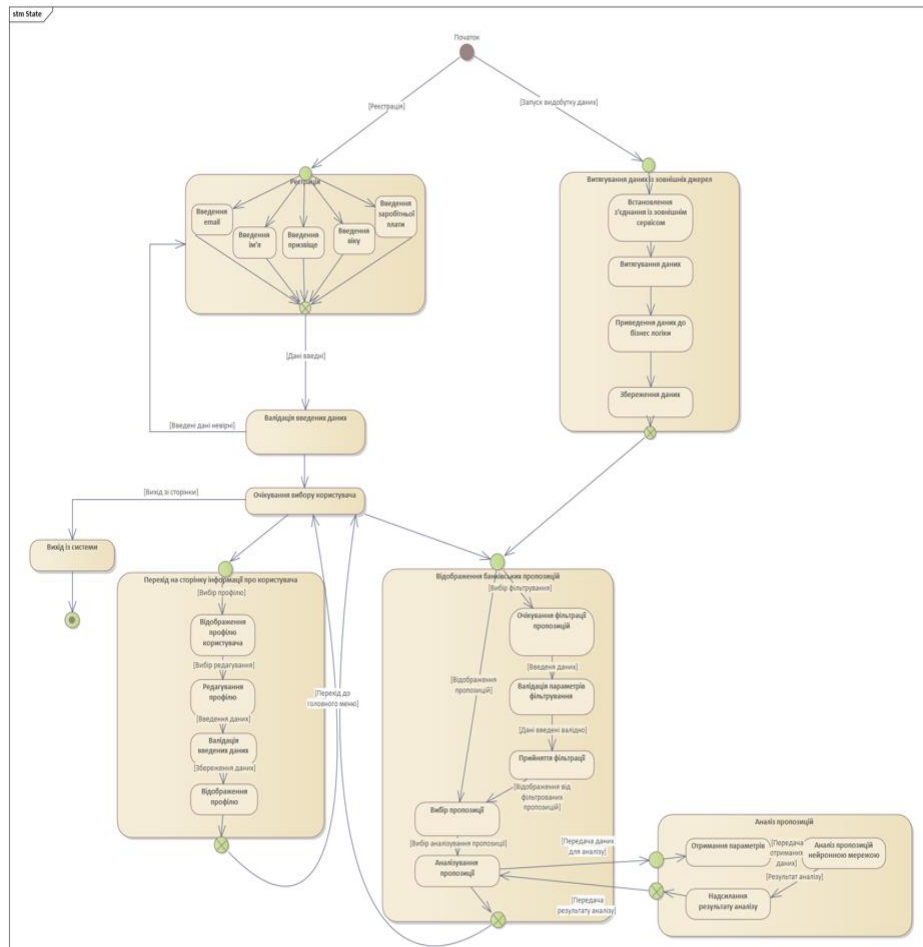


Рис.5. Загальна діаграма станів.

Функціональність системи:

Система працює в чотирьох основних режимах: очікування, збору даних, аналізу та відображення результатів.

- **Режим очікування:** Система перебуває в пасивному стані, очікуючи на дії користувача. У цей період вона може підвантажувати дані з зовнішніх джерел для подальшої обробки.
- **Режим збору даних:** Після того, як користувач вибрав певні пропозиції для аналізу, система активно збирає необхідну інформацію з зовнішніх джерел. Ці дані включають деталі про обрані пропозиції та профіль користувача.

- **Режим аналізу:** Отримані дані передаються на обробку штучній нейронній мережі, яка виконує аналіз пропозицій, порівнюючи їх з параметрами користувача та визначаючи їхню відповідність.
- **Режим відображення результатів:** Після завершення аналізу система представляє користувачеві результати у зрозумілому форматі. Результати містять оцінку вигідності кожної пропозиції та відповідні рекомендації.

Переходи між режимами:

- **Перехід з режиму очікування в режим збору даних:** Виконується після того, як користувач вибрав пропозиції для аналізу.
- **Перехід з режиму збору даних в режим аналізу:** Відбувається після того, як система збрала всі необхідні дані.
- **Перехід з режиму аналізу в режим відображення результатів:** Виконується після завершення аналізу даних.

Взаємодія з зовнішнім середовищем:

- **Вхідні дані:** Система отримує дані про банківські пропозиції та профілі користувачів з зовнішніх джерел, а також сигнали від користувача про вибір пропозицій для аналізу.
- **Вихідні дані:** Система генерує результати аналізу, які включають оцінку вигідності кожної пропозиції та рекомендації для користувача.

3.1.4. Схема бази даних.

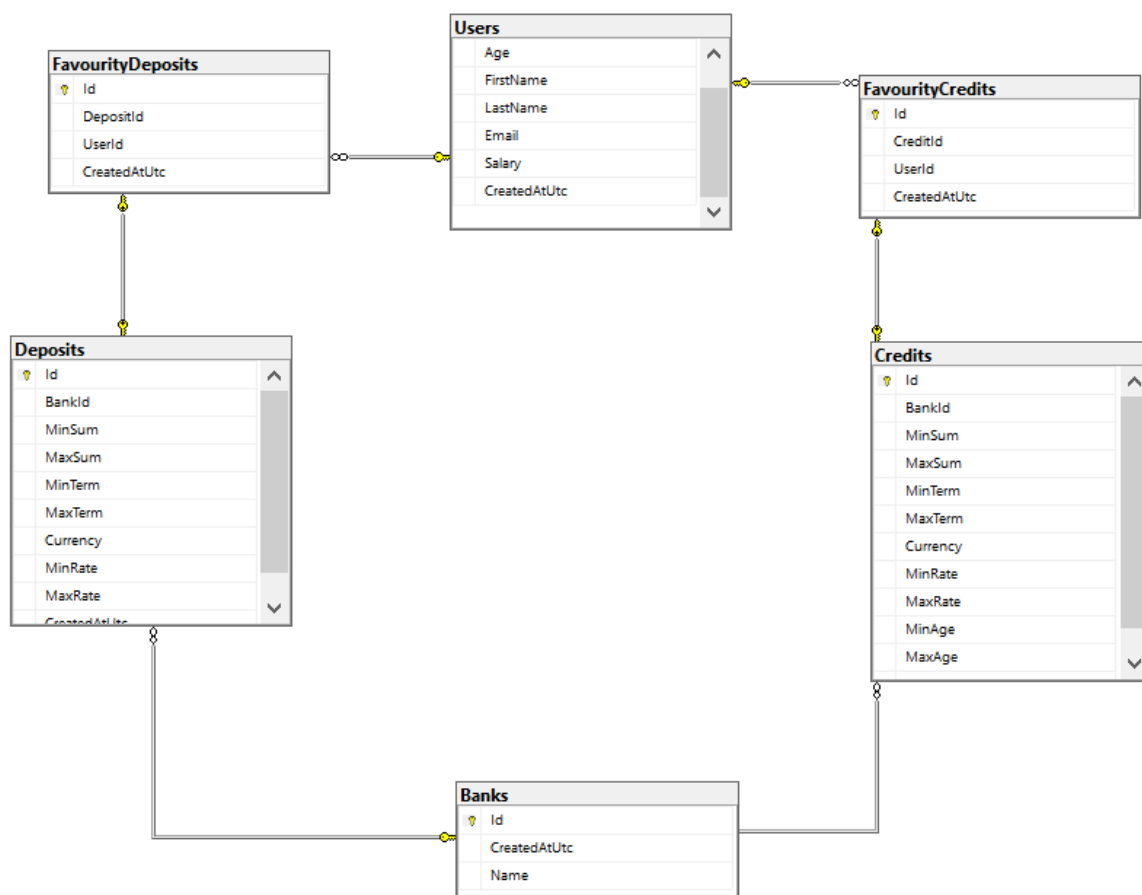


Рис.6. Схема бази даних на рівні SQL.

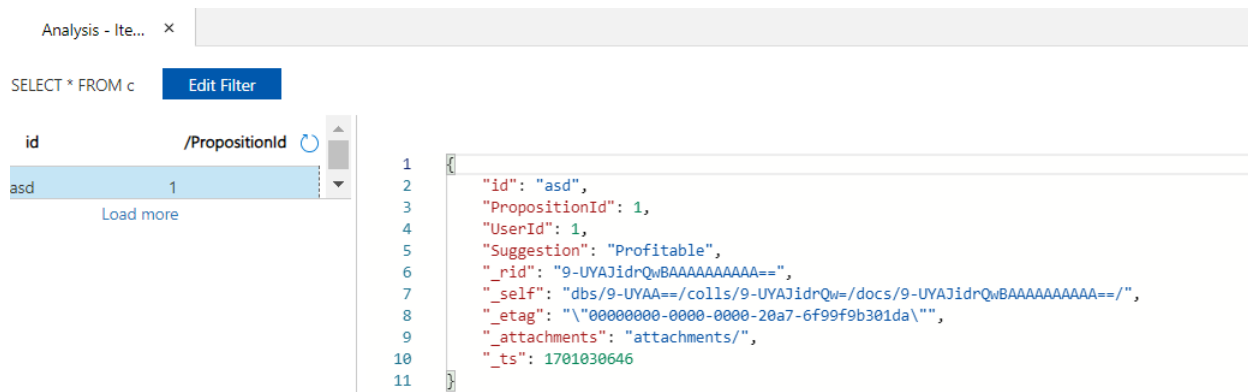


Рис.7. Схема бази даних на рівні NoSQL.

3.2. Розробка програмного забезпечення

Інструкції програмісту

Підготовка середовище виконання – це обов’язковий крок для налаштування та розробки системи. Включає в себе завантаження потрібного

ПЗ та бібліотек. Дана інструкція описує покрокове налаштування середовища на локальному комп'ютері.

3.2.1. Встановлення IDE

Щоб створювати, тестувати та вдосконалювати наш програмний продукт, нам потрібне спеціальне програмне середовище. Одним з найпопулярніших і потужних інструментів для цього є Microsoft Visual Studio. Ми рекомендуємо використовувати безкоштовну версію Community, яку можна завантажити з офіційного сайту за посиланням: <https://visualstudio.microsoft.com/>. Visual Studio надає всі необхідні інструменти для розробки, включаючи редактор коду, відладчик та інші корисні функції.

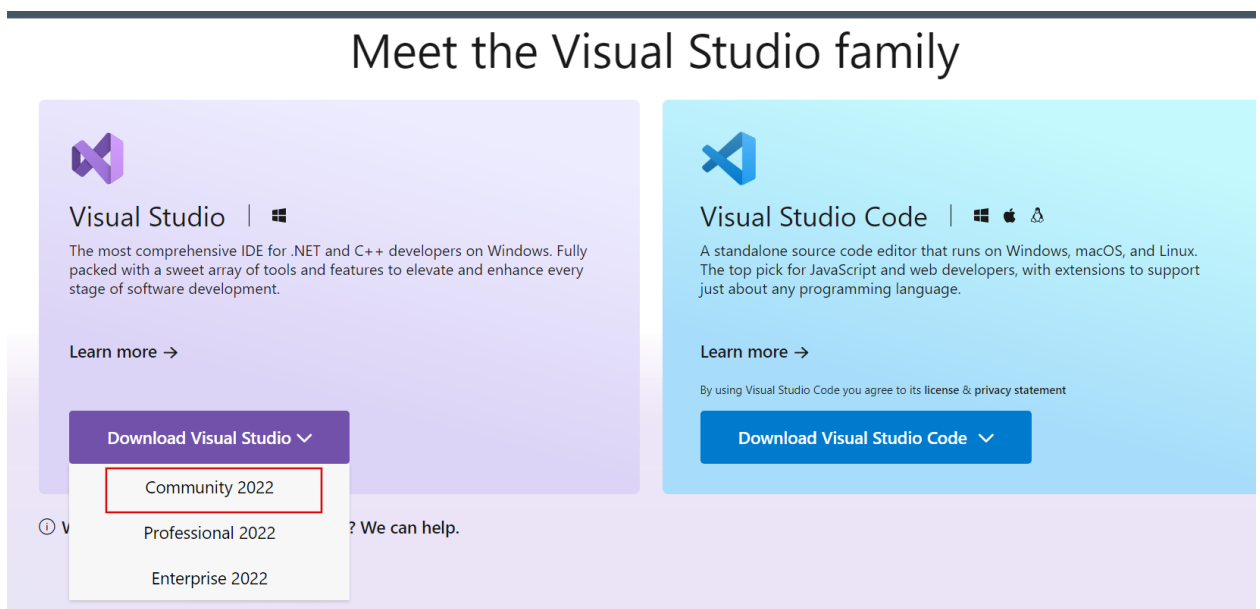


Рис.8. Завантаження Visual Studio.

Для ефективної роботи з базами даних NoSQL нам знадобиться спеціальний інструмент – емулятор Azure Cosmos DB. Цей емулятор дозволяє створювати локальні бази даних, які імітують роботу реальної бази даних Azure Cosmos DB. Це дуже зручно для розробки та тестування додатків.

Завантажити емулятор можна з офіційної документації Microsoft за посиланням: <https://learn.microsoft.com/en-us/azure/cosmos-db/how-to-develop-emulator?tabs=windows%2Ccsharp&pivots=api-nosql>

💡 Tip

The **emulator release notes** article lists all the available versions and the feature updates that were made in each release.

1. Download the **Azure Cosmos DB emulator**.
2. Run the installer on your local machine with **administrative privileges**.
3. The emulator automatically installs the appropriate developer certificates and configures firewall rules on your local machine.

Рис.9. Завантаження Azure Cosmos DB

Окрім NoSQL баз даних, нам знадобиться також реляційна база даних. Для цього ми використаємо Microsoft SQL Server (MSSQL). MSSQL є однією з найпопулярніших реляційних баз даних і пропонує широкий спектр можливостей. Щоб почати роботу з MSSQL, завантажте її з офіційного сайту Microsoft за посиланням: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Or, download a free specialized edition



Developer

SQL Server 2022 Developer is a full-featured free edition, licensed for use as a development and test database in a non-production environment.

[Download now](#)



Express

SQL Server 2022 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.

[Download now](#)

Рис.10. Завантаження MSSQL

Тепер нам необхідно встановити платформу .NET. Існує декілька способів це зробити, але ми розглянемо ручне встановлення з офіційного сайту. Цей метод дає більше контролю над процесом і дозволяє вибрати саме ті компоненти, які нам потрібні.

Перейдіть на офіційний сайт .NET і завантажте дистрибутив, що відповідає вашій операційній системі. Під час інсталяції слідуйте інструкціям на екрані.

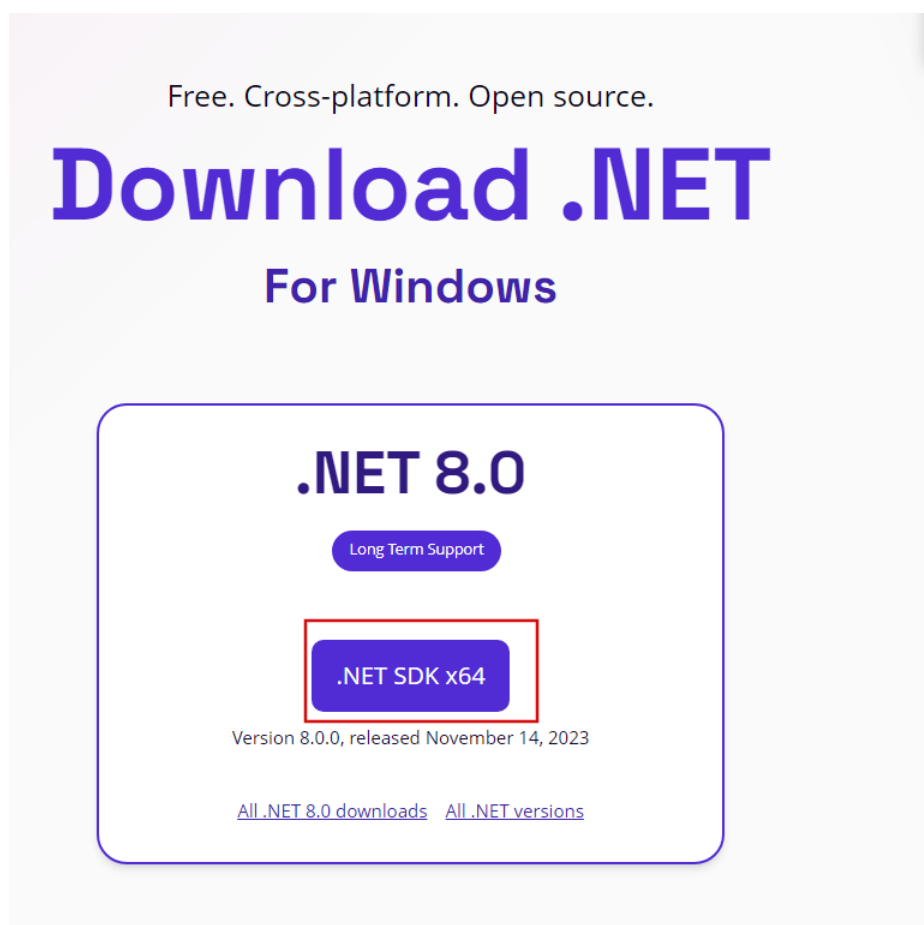


Рис.11. Встановлення.NET

3.2.2. Завантаження проекту

Щоб ефективно керувати розробкою нашого проекту, ми використовуємо систему контролю версій Git. Вона дозволяє відстежувати всі зміни в коді, співпрацювати з іншими розробниками та створювати резервні копії проекту. Для роботи з Git нам потрібно встановити клієнт Git на наш комп'ютер. Завантажте останню версію Git з офіційного сайту і встановіть її. Після встановлення ви зможете клонувати наш проект з GitHub.



Рис.11. Встановлення Git.

Після встановлення Git, створимо локальну папку для нашого проекту. Далі клонуємо весь проект з GitHub за посиланням <https://github.com/kanilasz/BankPropositionsPredictor.git>. Це дозволить нам працювати з кодом локально і синхронізувати зміни з репозиторієм на GitHub.

Для планування та відстеження задач ми використовуємо дошку Kanban на GitHub: <https://github.com/users/kanilasz/projects/1>. Тут ви можете побачити, які завдання вже виконані, які в процесі, і які ще потрібно виконати. Початок роботи над проектами

Тепер, коли середовище налаштовано, можна запускати проект. Наш проект складається з трьох основних модулів: інтерфейс користувача (UI), модуль для вилучення даних та модуль ШНМ для аналізування даних. Щоб запустити UI, відкрийте рішення у Visual Studio і запустіть відповідний проект.

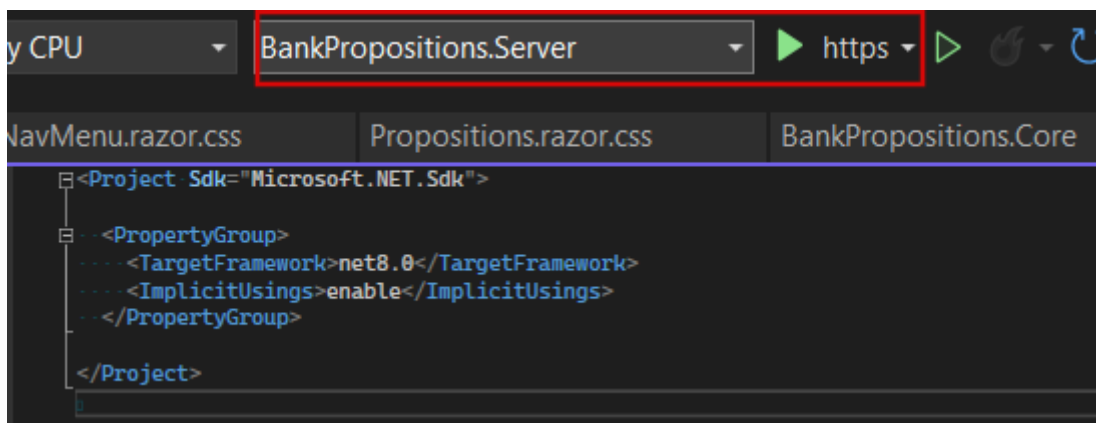


Рис.12. Запуск клієнтської сторони

Для витягування та обробки даних потрібно запустити інший проект під назвою BankPropositions.Parser.Та викликати єдину команду GET.

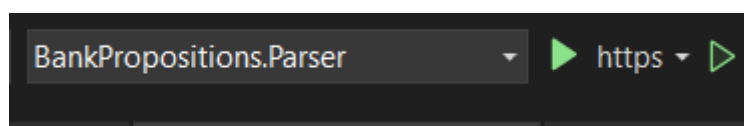


Рис.13. Запуск витягування даних.

3.2.3. Проекти у контейнерах

Кожен компонент системи працює в ізольованому контейнері. Це означає, що зміни в одному контейнері не впливають на інші. Це підвищує надійність і дозволяє швидко розгортати нові версії компонентів. Для порівняння, якщо б використовувалась віртуальна машина для розгортання то – це цілий комп'ютер в комп'ютері. У порівняння до цього контейнер – це більш легкий і ефективний спосіб ізоляції, який передбачає спрощене налаштування інфраструктури для роботи. Контейнери ділять одне ядро операційної системи, що дозволяє використовувати ресурси більш ефективно.

Оскільки система підтримується у контейнерах, тому потрібно встановити Docker Desktop з офіційного сайту: <https://www.docker.com/products/docker-desktop/>



Рис.14. Завантаження Docker Desktop

3.3. Інформаційна взаємодія

Інформаційна взаємодія в контексті описаної системи є критично важливим процесом, який забезпечує надходження актуальних даних з різних джерел для подальшого аналізу. Діаграма чітко ілюструє, як система взаємодіє з зовнішнім світом, та які компоненти відповідають за цей процес.

Основні аспекти інтеграції

- **Вибір джерел даних:** Система використовує BANKI.UA для отримання пропозицій. Вибір саме цього джерел даних обумовлений їхньою релевантністю для аналізу банківських пропозицій.
- **Метод взаємодії:**
 - **Веб-парсинг:** Для вилучення даних з BANKI.UA використовується веб-скрапінг HTML. Цей метод дозволяє отримувати дані з веб-сторінок, які не надаються через API.
- **Протоколи передачі даних:** Використовуються стандартний протокол для передачі даних по мережі - HTTPS.
- **Формат даних:** Отримані дані на початках будуть у форматі JSON.
- **Частота оновлення даних:** Передбачено періодичне оновлення даних за розкладом або по події.

Реалізація механізмів інформаційної взаємодії з зовнішніми системами забезпечує:

- **Актуальність даних:** Забезпечує використання найсвіжіших даних для аналізу.
- **Розширення можливостей системи:** Дозволяє отримувати додаткову інформацію, яка не може бути отримана з внутрішніх джерел.

- **Автоматизація процесів:** Автоматизує збір даних, що зменшує трудомісткість ручних операцій.

Можливі проблеми та їх вирішення

- **Зміни у структурі даних:** Зміни у форматі або структурі даних, що надаються зовнішніми системами, можуть вимагати модифікації модуля отримання даних.
- **Обмеження доступу:** Можливі обмеження на доступ до даних або обсяг запитів.
- **Помилки передачі даних:** Можуть виникати помилки при передачі даних по мережі.

3.4. Розгортання та масштабування

3.4.1. Розуміння контейнерної архітектури

Використання контейнерної архітектури для розгортання системи має ряд переваг, таких як:

- **Ізоляція компонентів:** Кожен модуль системи (отримання даних, обробка, представлення) може бути упакований в окремий контейнер, що забезпечує їхню незалежність та спрощує управління.
- **Швидке розгортання:** Контейнери можуть бути швидко розгорнуті та масштабовані.
- **Портативність:** Контейнери можна розгорнути в будь-якому середовищі, що підтримує контейнери (локально, в хмарі тощо).

3.4.2. Варіанти розгортання контейнерів в Azure

- **Azure Container Instances (ACI):** Це найпростіший варіант для запуску окремих контейнерів без управління кластером.

- **Azure Kubernetes Service (AKS):** Потужна платформа для управління контейнерними кластерами на основі Kubernetes. Забезпечує високу доступність, масштабованість та надійність.
- **Azure Container Apps:** Сервіс для запуску контейнеризованих додатків без управління інфраструктурою.

3.4.3. Процес розгортання

1. **Створення Docker-образів:** Для кожного модуля системи створюється Docker-образ, який містить всі необхідні залежності та код.
2. **Публікація образів в Azure Container Registry:** Отримані образи публікуються в реєстрі контейнерів Azure для зберігання та розповсюдження.
3. **Розгортання в Azure:**
 - **ACI:** Створити інстанцію контейнера для кожного модуля.
 - **AKS:** Створити Kubernetes-маніфести для опису розгортання, сервісів та інших ресурсів. Запустити ці маніфести в кластері AKS.
 - **Azure Container Apps:** Створити контейнерну програму та вказати необхідні образи та конфігурації.

3.4.4. Масштабування

- **Автоматичне масштабування:**
 - **ACI:** На основі CPU, пам'яті або кількості запитів.
 - **AKS:** Використовувати Horizontal Pod Autoscaler (HPA) для автоматичного масштабування реплік.
 - **Azure Container Apps:** На основі правил масштабування, заснованих на метриках.
- **Ручне масштабування:** Додавання або видалення екземплярів контейнерів вручну.

3.4.5. Переваги контейнерного підходу

- **Швидке розгортання та масштабування:** Можливість швидко додавати або видаляти ресурси відповідно до навантаження.
- **Ізоляція компонентів:** Кожен модуль працює в ізольованому середовищі, що знижує ризик впливу збоїв одного компонента на інші.
- **Висока доступність:** Можливість розгортання декількох реплік контейнерів для забезпечення високої доступності.
- **Гнучкість:** Можливість використання різних технологій і фреймворків всередині контейнерів.

4. НАУКОВО-ДОСЛІДНА ЧАСТИНА

Мета даного дослідження полягає у виявленні оптимального алгоритму машинного навчання для класифікації банківських пропозицій, у конкретному випадку - це депозитів. Буде проведено порівняльний аналіз ефективності алгоритмів L-BFGS, Fast Forest, Fast Tree та LightGbm на заданому наборі даних. Отримані результати дозволять розробити рекомендації щодо вибору алгоритму для конкретної задачі класифікації.

4.1. Експерименти та результати

4.1.1. Підготовка експериментальних даних

Для проведення експериментів була сформована база даних пропозицій та характеристик користувачів. Ці дані було класифіковано вручну та будуть використані для навчання нейронної мережі. Детально оглянути ці дані можливо у Додатку А у форматі Excel. База даних містить 48 записів.

Опишемо, як було класифіковано ці дані, та їх атрибути:

- **Ручна класифікація:** Всі пропозиції були попередньо класифіковані вручну на 5 груп результатів аналізу: невідомий, не вигідний, сумнівний, середній, потребує додаткового аналізу, вигідний. Це дозволило створити необхідний набір даних з відомими мітками класів, для навчання моделей.
- **Атрибути даних:**
 - **Характеристики пропозиції:**
 - **Мінімальна сума:** Це найменша сума коштів, яку необхідно вкласти, щоб скористатися даною пропозицією. Це важливий параметр для клієнтів з різними фінансовими можливостями.

- **Максимальна сума:** Максимальна сума, яку можна вкласти за цією пропозицією. Це обмеження може бути встановлено банком.
- **Валюта:** Валюта, в якій здійснюється вклад (наприклад, гривня, долар США, євро).
- **Мінімальний відсоток:** Найнижчий відсоток, який клієнт може отримати за цим вкладом.
- **Максимальний відсоток:** Найвищий відсоток, який клієнт може отримати за цим вкладом. Зазвичай, максимальний відсоток залежить від суми вкладу та строку.
- **Максимальний термін вкладу:** Максимальний термін, на який можна відкрити вклад за цією пропозицією.
- **Мінімальний термін вкладу:** Мінімальний термін, на який необхідно відкрити вклад за цією пропозицією.

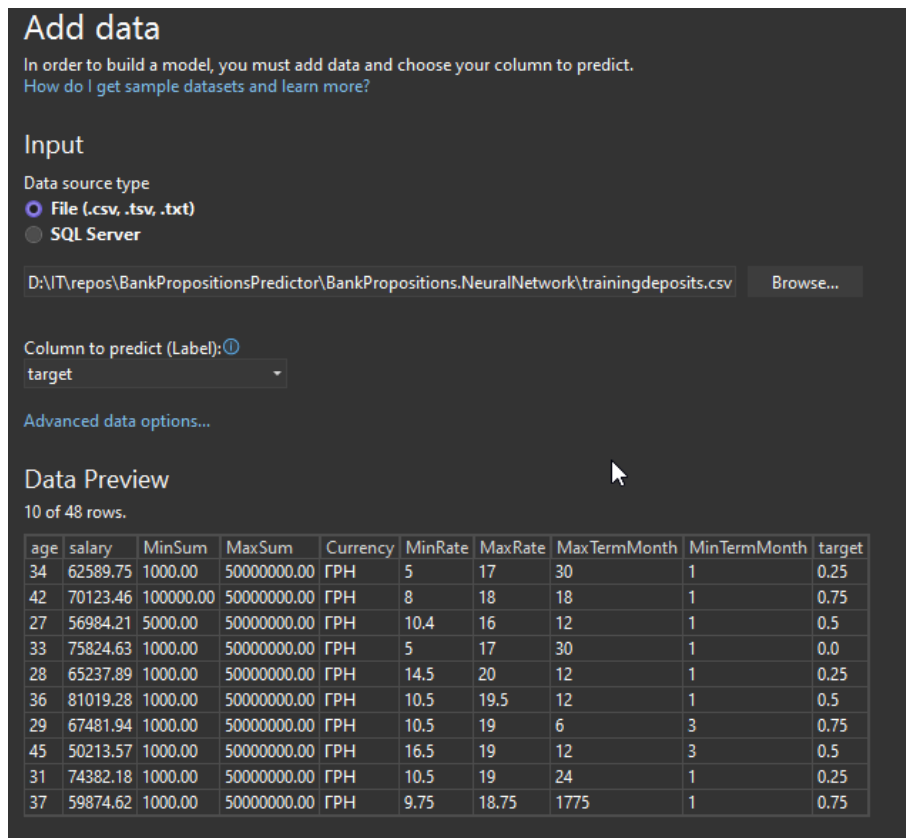
○ **Характеристики користувача:**

- **Вік:** Цей атрибут дозволяє розділити клієнтів на вікові групи і аналізувати, які фінансові продукти найбільш популярні серед різних поколінь. Наприклад, молоді люди можуть бути більш зацікавлені в кредитах на навчання, тоді як люди старшого віку - в пенсійних продуктах.
- **Зарплата:** Цей атрибут відображає платоспроможність клієнта. Він дозволяє сегментувати клієнтів за рівнем доходу і пропонувати їм відповідні фінансові продукти. Наприклад, клієнтам з високим доходом можна пропонувати інвестиційні продукти, а з меншим короткострокові пропозиції з вигідним відсотком.

4.1.2. Експеримент з алгоритмом L-BFGS

4.1.2.1. Навчання на дата сеті з довжиною 50 стрічок

Для початку експерименту з алгоритмом L-BFGS, здійснимо конфігурацію ШНМ. Вказуємо шлях до даних та цільового атрибуту, який надає кінцевий результат аналізу пропозицію.



Add data

In order to build a model, you must add data and choose your column to predict.
[How do I get sample datasets and learn more?](#)

Input

Data source type

☒ File (.csv, .tsv, .txt)

☐ SQL Server

D:\IT\repos\BankPropositionsPredictor\BankPropositions.NeuralNetwork\trainingdeposits.csv [Browse...](#)

Column to predict (Label): [?](#)

target

[Advanced data options...](#)

Data Preview

10 of 48 rows.

	age	salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTermMonth	MinTermMonth	target
34	62589.75	1000.00	50000000.00	GRN	5	17	30	1	0.25	
42	70123.46	100000.00	50000000.00	GRN	8	18	18	1	0.75	
27	56984.21	5000.00	50000000.00	GRN	10.4	16	12	1	0.5	
33	75824.63	1000.00	50000000.00	GRN	5	17	30	1	0.0	
28	65237.89	1000.00	50000000.00	GRN	14.5	20	12	1	0.25	
36	81019.28	1000.00	50000000.00	GRN	10.5	19.5	12	1	0.5	
29	67481.94	1000.00	50000000.00	GRN	10.5	19	6	3	0.75	
45	50213.57	1000.00	50000000.00	GRN	16.5	19	12	3	0.5	
31	74382.18	1000.00	50000000.00	GRN	10.5	19	24	1	0.25	
37	59874.62	1000.00	50000000.00	GRN	9.75	18.75	1775	1	0.75	

Рис.4.3.2.1.1. Налаштування даних для навчання.

Наступним кроком налаштовуємо тренування мережі. Встановлюємо потрібний алгоритм навчання, кількість секунд навчання, алгоритм оптимізації та запускаємо навчання.

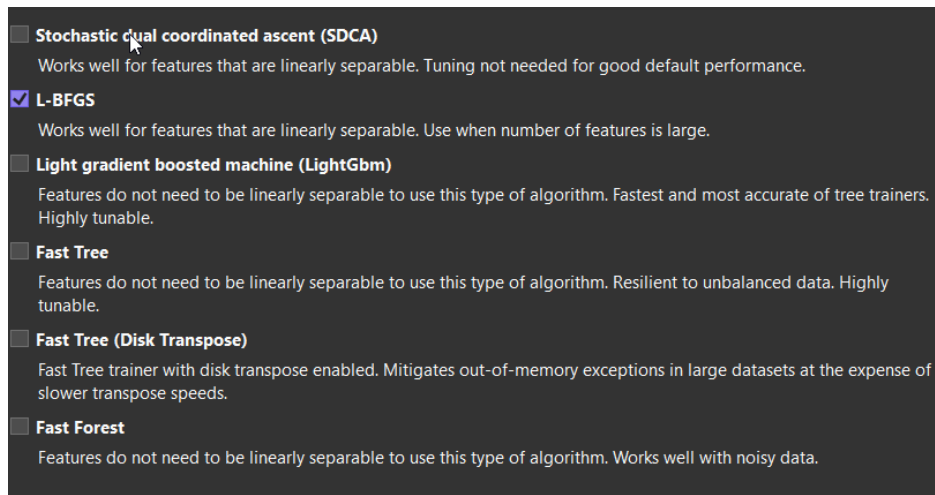


Рис.4.3.2.1.2. Налаштування алгоритму мережі.

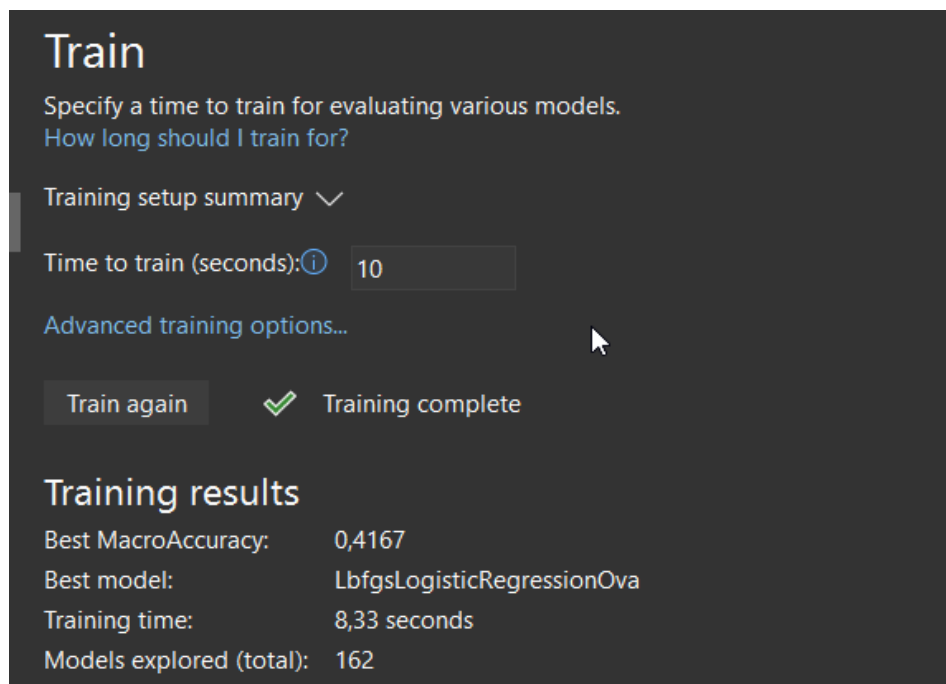


Рис. Рис.4.3.2.1.3 Навчання мережі.

Використання дата сету з довжиною 50 дало найкращу точність: 0.416. Здійснимо аналіз нашою мережею на даних котрі не використовувались під час навчання. А саме на:

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm	MinTerm
		m					Month	Month
42	70123.46	100000.00	5000000.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 25% що класифікується як сумнівний. Проте для даної пропозиції вручну

було класифіковано результат, який потребує додаткового аналізу.

4.1.2.2. Навчання на дата сеті з довжиною 100 стрічок

Повторимо навчання

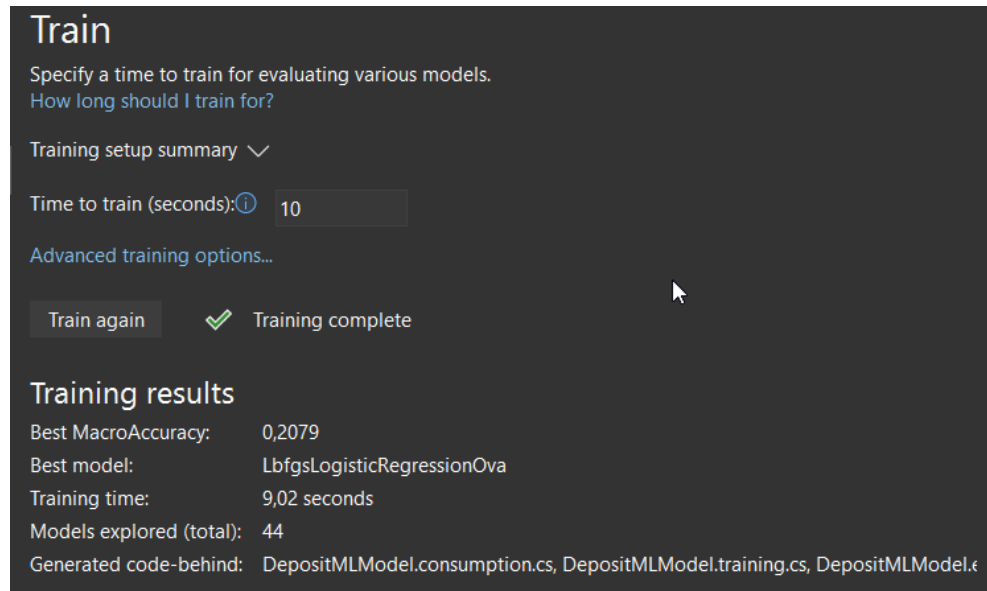


Рис.4.3.2.2.1 Навчання мережі.

Використання дата сету з довжиною 100 дало точність: 0.5.

Проаналізуємо тестову пропозицію.

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123.46	100000.00	5000000.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 0.25% на 55% та 0.75 на 45%, що класифікується як сумнівний, проте наближений до необхідного варіанту.

4.1.3. Експеримент з алгоритмом Fast Forest

4.1.3.1. Навчання на дата сеті з довжиною 50 стрічок

Здійснюємо конфігурацію даних для навчання так само, як і для алгоритму L-BFGS. Налаштовуємо алгоритм мережі.

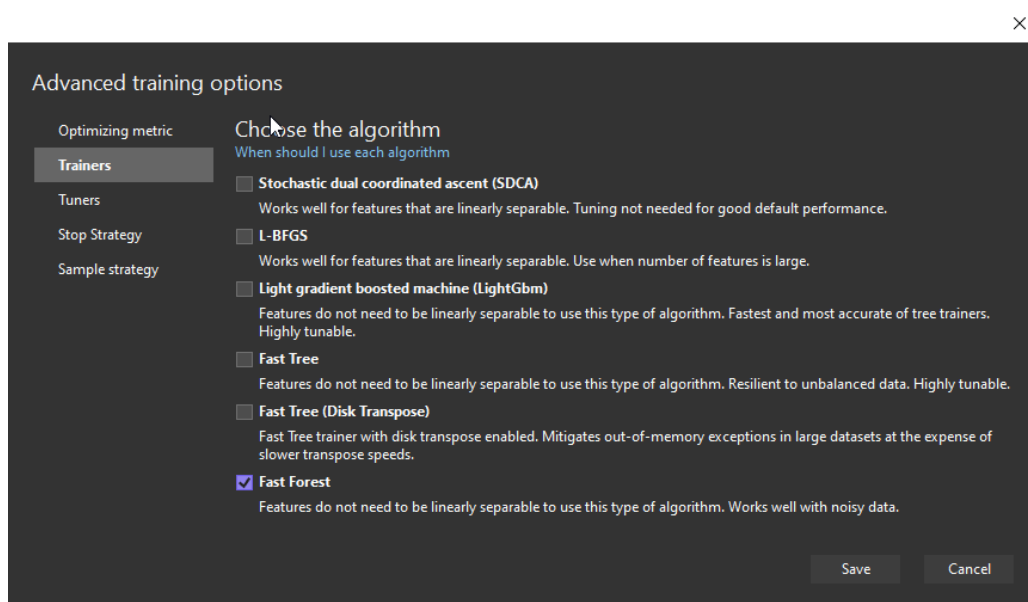


Рис. 4.3.3.1 Налаштування алгоритму мережі.

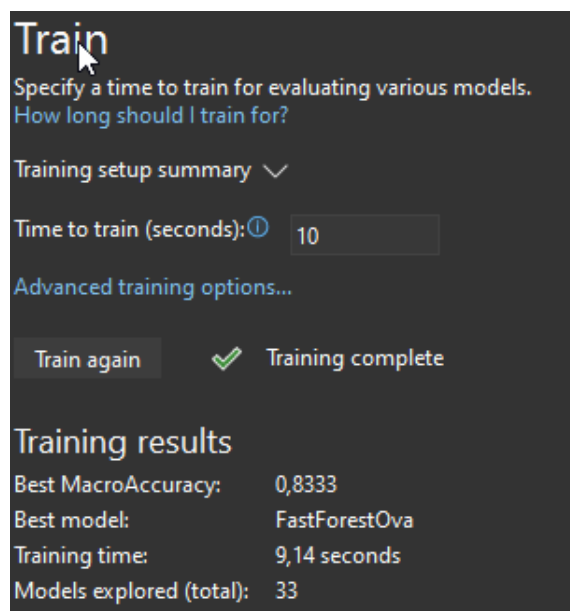


Рис. 4.3.3.2. Навчання мережі.

Використання дата сету з довжиною 50 дало найкращу точність: 0.83. Здійснимо аналіз нашою мережею на даних котрі не використовувались під час навчання. А саме на:

Age	Salary	MinSum m	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123 .46	100000 .00	5000000 0.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 0 що класифікується як не вигідний. Проте для даної пропозиції вручну було

класифіковано результат, який - потребує додаткового аналізу.

4.1.3.2. Навчання на дата сеті з довжиною 100 стрічок

Повторимо навчання

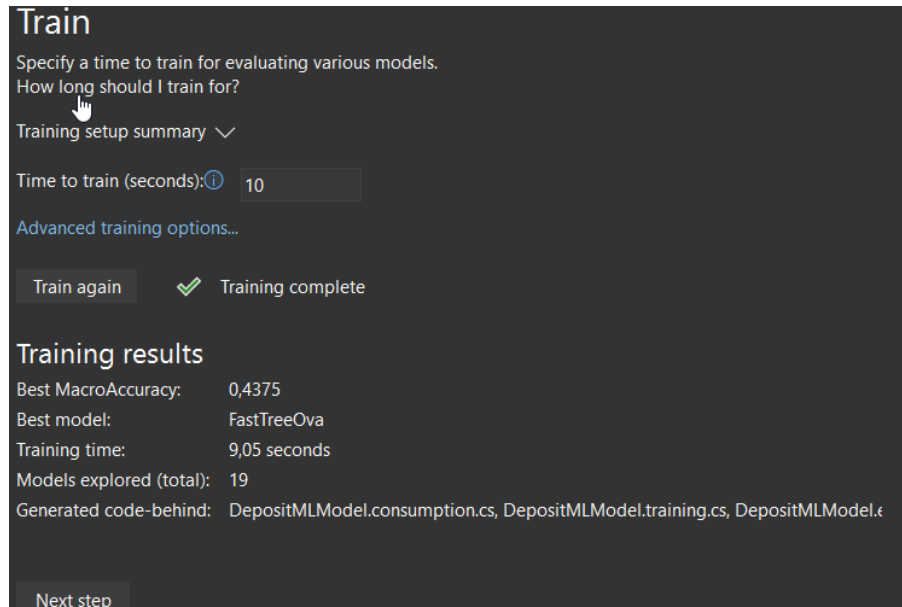


Рис. 4.3.3.2.1. Навчання мережі.

Використання дата сету з довжиною 100 дало точність: 0.5

Проаналізуємо тестову пропозицію.

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123.46	100000.00	5000000.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 0.75 на 75% що класифікується як такий, що відповідає результату визначеному вручну.

4.1.4. Експеримент з алгоритмом Fast Tree

4.1.4.1. Навчання на дата сеті з довжиною 50 стрічок

Здійснюємо конфігурацію даних для навчання так само, як і для алгоритму L-BFGS. Налаштовуємо алгоритм мережі.

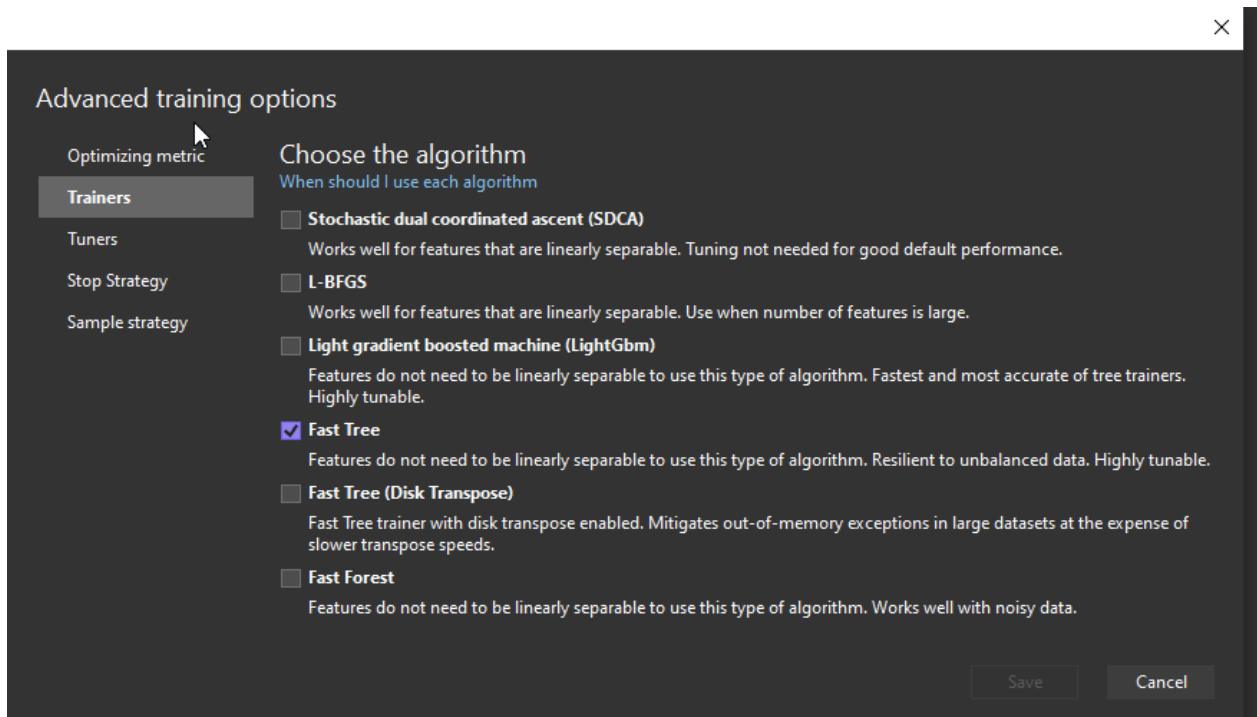


Рис. 4.3.4.1.1 Налаштування алгоритму мережі.

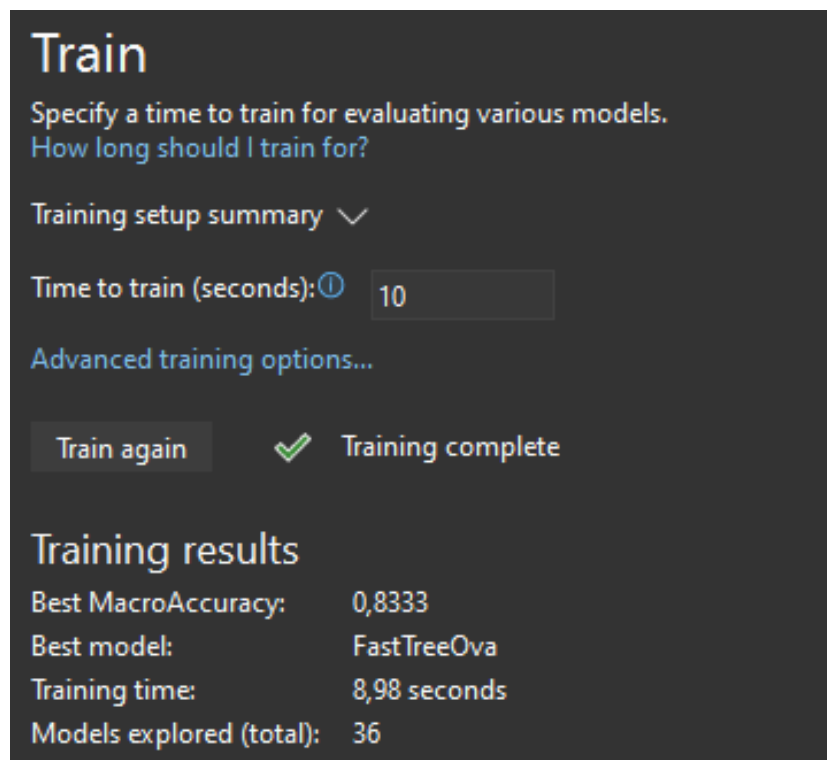


Рис. 4.3.4.1.1. Навчання мережі.

Використання дата сету з довжиною 50 дало найкращу точність: 0.833, що є набагато кращим результатом у порівнянні з L-BFGS. Здійснимо аналіз нашою мережею на даних котрі не використовувались під час навчання. А саме на:

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123 .46	100000 .00	5000000 0.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 0.5 що класифікується як середнє. Проте для даної пропозиції вручну було класифіковано результат, який - потребує додаткового аналізу.

4.1.4.2. Навчання на дата сеті з довжиною 100 стрічок

Повторимо навчання

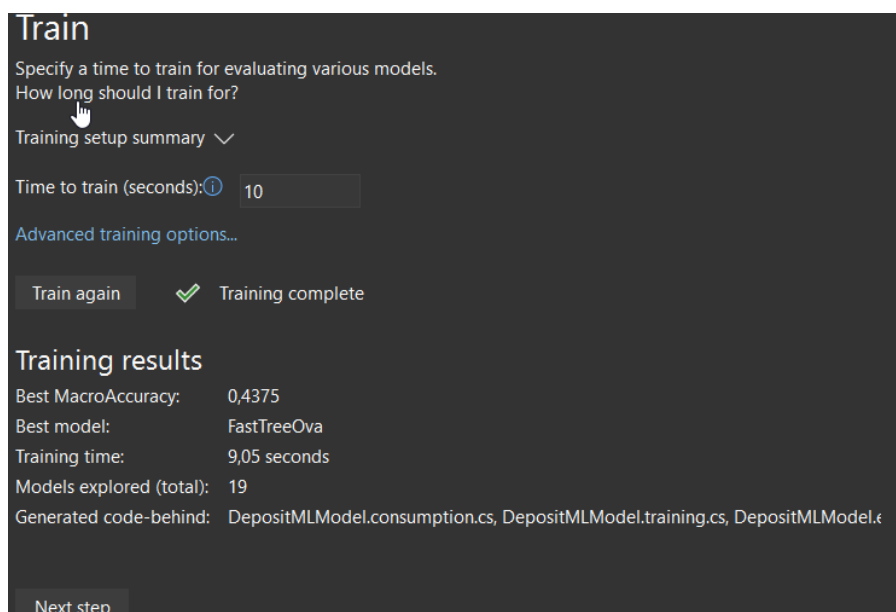


Рис. 4.3.4.2.1 Навчання мережі.

Використання дата сету з довжиною 100 дало точність: 0.4375

Проаналізуємо тестову пропозицію.

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123 .46	100000 .00	5000000 0.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 0 на 16% що класифікується як не вигідний. Проте для даної пропозиції вручну

було класифіковано результат, який потребує додаткового аналізу.

4.1.5. Експеримент з алгоритмом LightGbm

4.1.5.1. Навчання на дата сеті з довжиною 50 стрічок

Конфігурація даних для навчання ідентична, як і для всіх алгоритмів. Налаштовуємо алгоритм мережі.

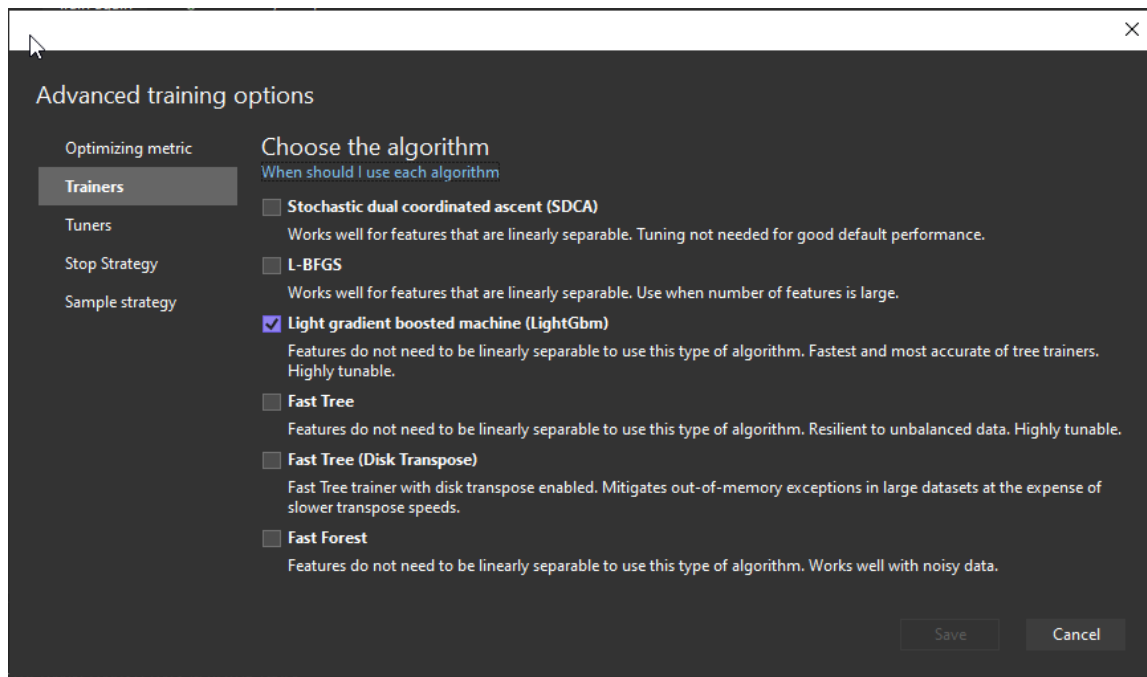


Рис. 4.3.5.1.1 Налаштування алгоритму мережі.

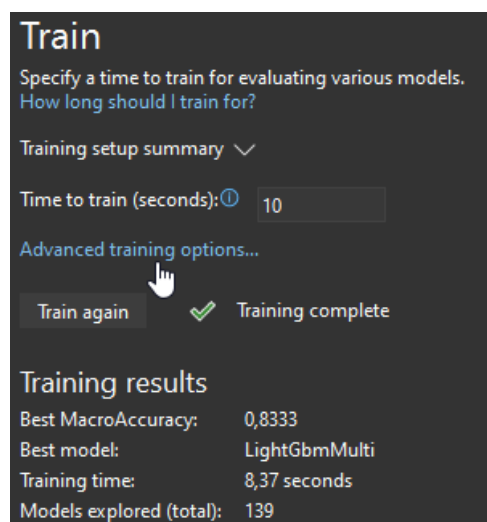


Рис. 4.3.5.1.2. Навчання мережі.

Використання дата сету з довжиною 50 дало найкращу точність: 0.83. Здійснимо аналіз нашою мережею на даних котрі не використовувались під час навчання. А саме на:

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123 .46	100000 .00	5000000 0.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 0.75% що класифікується як такий, що потребує додаткового аналізу. Це відповідає значенню класифікованому вручну.

4.1.5.2. Навчання на дата сеті з довжиною 100 стрічок

Повторимо навчання

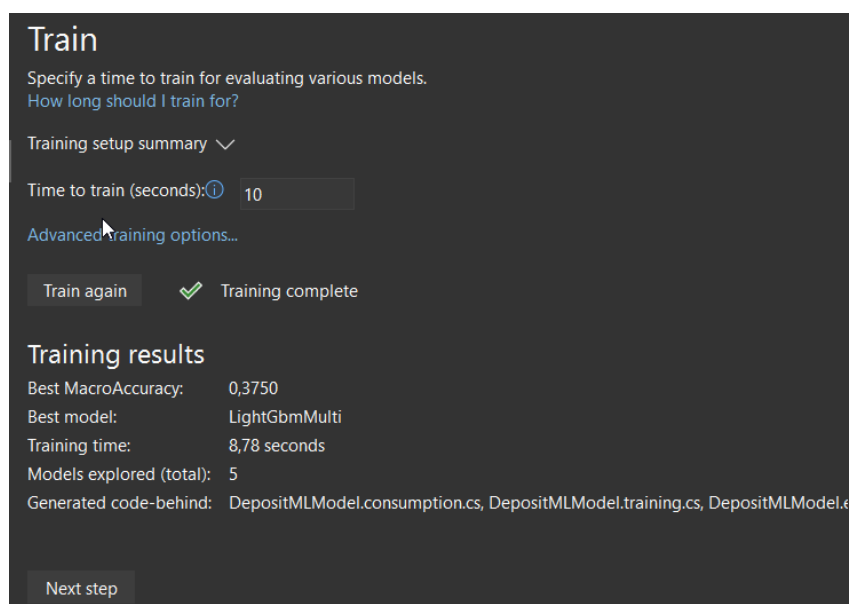


Рис. 4.3.5.2.1 Навчання мережі.

Використання дата сету з довжиною 100 дало точність: 0.375

Проаналізуємо тестову пропозицію.

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123 .46	100000 .00	5000000 0.00	ГРН	8	18	18	1

Результат аналізу класифікував дану пропозицію і дані користувача, як 0.75 на 92% що є найточнішим серед усіх.

4.1.6. Порівняння результатів роботи алгоритмів

Нагадаємо яка була тестова пропозиція та який має бути результат аналізу.

Age	Salary	MinSum	MaxSum	Currency	MinRate	MaxRate	MaxTerm Month	MinTerm Month
42	70123.46	100000.00	5000000.00	ГРН	8	18	18	1

Результат: потребує додаткового аналізу.

Алгоритм	50 стрічок		100 стрічок	
	Точність	Результат аналізу	Точність	Результат аналізу
L-BFGS	0,416	Сумнівний	0,5	Сумнівний
Fast tree	0,83	Середнє	0,4375	Невигідно
Fast forest	0,83	Невигідно	0,4375	Потребує додаткового аналізу
LightGbm	0,83	Потребує додаткового аналізу	0,375	Потребує додаткового аналізу

Ця таблиця демонструє результати роботи кількох алгоритмів машинного навчання при аналізі банківської пропозиції для користувача з заданими характеристиками.

- **Алгоритми:** L-BFGS, Fast tree, Fast forest, LightGBM.
- **Метрика точності:** Використовується для оцінки якості роботи алгоритму. Чим вище значення, тим точніше алгоритм передбачив відповідний результат.
- **Результат аналізу:** Категорична оцінка, яка відображає загальну оцінку результату (сумнівний, середній, невигідний, потребує додаткового аналізу).

Ключові спостереження:

- **Низька точність L-BFGS:** Алгоритм L-BFGS показав найнижчу точність у обох випадках (50 та 100 стрічок), що вказує на його недостатню ефективність для даної задачі.
- **Висока точність деревних алгоритмів:** Алгоритми Fast tree, Fast forest та LightGBM показали значно вищу точність, особливо при використанні 50 стрічок. Це свідчить про те, що для даної задачі краще підходять ансамблі дерев рішень.
- **Неоднозначні результати для великої кількості стрічок:** При використанні 100 стрічок точність алгоритмів знизилася, особливо для Fast tree та Fast forest. Це може бути пов'язано з перенавчанням моделей або недостатньою кількістю даних для навчання.
- **Недостатність категоричної оцінки:** Категорична оцінка "Потребує додаткового аналізу" використовується для більшості результатів, що ускладнює порівняння та вибір найкращого алгоритму.

Висновки:

1. **Ефективність деревних алгоритмів:** Для даної задачі найбільш ефективними виявилися алгоритми на основі дерев рішень (Fast tree, Fast forest, LightGBM). Вони показали значно кращу точність порівняно з L-BFGS.
2. **Необхідність подальшого аналізу:** Для більш об'єктивного порівняння алгоритмів необхідно провести більш детальний аналіз, включаючи:
 - **Використання інших метрик:** Крім точності, можна розглянути такі метрики, як повнота, F1-мера.
 - **Аналіз важливості ознак:** Визначити, які ознаки найбільше впливають на результат прогнозування.
 - **Додаткові експерименти:** Змінювати гіперпараметри алгоритмів, використовувати різні методи перехресної перевірки.
3. **Проблема перенавчання:** Для алгоритмів Fast tree та Fast forest спостерігається зниження точності при збільшенні кількості даних, що може свідчити про перенавчання. Для вирішення цієї проблеми можна

використовувати регуляризацію, ранню зупинку або зменшити складність моделей.

5. СПЕЦІАЛЬНА ЧАСТИНА

5.1. Формулювання наукової проблеми

Основною проблемою дослідження є визначення оптимального алгоритму машинного навчання для класифікації банківських депозитів з метою підвищення точності прогнозування та зменшення ризиків. Необхідно порівняти ефективність різних алгоритмів (L-BFGS, Fast Forest, Fast Tree, LightGbm) та визначити, який з них найбільш підходить для вирішення даної задачі. Крім того, важливо дослідити вплив різних гіперпараметрів алгоритмів та методів попередньої обробки даних на точність класифікації.

5.1.1. Мета дослідження

Порівняльний аналіз ефективності алгоритмів L-BFGS, Fast Forest, Fast Tree та LightGbm для класифікації банківських депозитів.

- Оцінка впливу різних методів попередньої обробки даних на точність класифікації.
- Виявлення найбільш значимих ознак, що впливають на класифікацію депозитів.

5.1.2. Завдання дослідження

1. **Збір та підготовка даних:** Сформувати набір даних, що містить інформацію про банківські депозити, провести очищення даних, нормалізацію та кодування категоріальних змінних, розділити дані на навчальну та тестову вибірки.
2. **Вибір та налаштування алгоритмів:** Реалізувати алгоритми L-BFGS, Fast Forest, Fast Tree та LightGbm, підібрати оптимальні гіперпараметри для кожного алгоритму.
3. **Оцінка якості моделей:** Оцінити точність моделей за допомогою метрик (точність, повнота, F1-мера, AUC-ROC), побудувати матриці плутанини, проаналізувати важливість ознак.
4. **Порівняння результатів:** Порівняти ефективність різних алгоритмів,

проаналізувати вплив різних гіперпараметрів та методів попередньої обробки даних.

- 5. Формулювання висновків:** Сформулювати висновки щодо ефективності різних алгоритмів, визначити оптимальний алгоритм для даної задачі та надати рекомендації для практичного застосування.

Очікувані результати

- Розроблена модель класифікації банківських депозитів з високою точністю.
- Визначені оптимальні алгоритми для даної задачі.
- Оцінка впливу різних факторів на результати класифікації.
- Рекомендації щодо застосування отриманих результатів на практиці.

5.2. Методологія дослідження

Метою дослідження було визначення оптимального алгоритму машинного навчання для класифікації банківських депозитів з метою підвищення точності прогнозування та зменшення ризиків.

Для досягнення поставленої мети було використано наступну методологію:

5.2.1. Вибір алгоритмів

Були обрані чотири алгоритми машинного навчання, які часто використовуються для задач класифікації: L-BFGS, Fast Forest, Fast Tree та LightGbm. Вибір цих алгоритмів був обумовлений їхнім широким застосуванням у подібних задачах.

5.2.2. Підготовка даних

Після збору даних було проведено їх очищення та нормалізацію. Дані були розділені на навчальну та тестову вибірки у співвідношенні 80% до 20%.

5.2.3. Налаштування гіперпараметрів:

Для кожного алгоритму було проведено підбір оптимальних гіперпараметрів за допомогою методу `esio-cfo`. Цей метод дозволив знайти найкраще поєднання параметрів, яке забезпечує максимальну точність моделі на наданій вибірці.

5.2.4. Оцінка моделей

Для оцінки якості моделей були використані наступні метрики: точність, повнота та результат класифікації на тестових даних.

5.2.5. Порівняння моделей

Було проведено порівняння моделей за значеннями метрик точності, повноти, F1-мери та AUC-ROC. Для візуалізації результатів були побудовані графіки ROC-кривих.

5.2.6. Обмеження дослідження

- **Обмежений набір даних:** Використання внутрішньої бази даних банку може обмежувати узагальнення результатів на інші банки.
- **Вибір алгоритмів:** Було розглянуто обмежене число алгоритмів, і можуть існувати інші алгоритми, які можуть показати кращі результати.

6. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1. Цифрова безпека при роботі з банківськими даними

Робота з великими обсягами банківських даних, особливо за допомогою штучних нейронних мереж, передбачає високий рівень відповідальності за їхню безпеку. Потенційні загрози включають:

- **Несанкціонований доступ:** Хакерські атаки, злом облікових записів можуть призвести до розголошення конфіденційної інформації клієнтів.
- **Викрадення даних:** Зловмисники можуть отримати доступ до бази даних і викрасти персональну інформацію клієнтів, що може призвести до фінансових втрат та порушення репутації банку.
- **Внутрішні загрози:** Дії невдоволених співробітників або помилки персоналу можуть призвести до витоку даних.

Для мінімізації цих ризиків необхідно вживати комплекс заходів:

- **Шифрування даних:** Забезпечення конфіденційності даних шляхом використання сучасних алгоритмів шифрування.
- **Системи виявлення вторгнень:** Регулярний моніторинг мережі для виявлення підозрілої активності.
- **Створення резервних копій:** Регулярне створення резервних копій даних для відновлення інформації у разі втрати.
- **Обмеження доступу:** Надання співробітникам мінімально необхідних прав доступу до даних.
- **Свідомість співробітників:** Проведення регулярних тренінгів з кібербезпеки для підвищення обізнаності співробітників про потенційні загрози.

6.2. Психологічні аспекти роботи з великими даними

Робота з великими обсягами даних може спричинити значний психологічний тиск на дослідника. Серед основних ризиків можна виділити:

- **Інформаційне перевантаження:** Великий обсяг даних, які необхідно обробити та проаналізувати, може призвести до перевтоми та зниження ефективності роботи.
- **Стрес від відповідальності:** Відповідальність за точність результатів дослідження може спричинити підвищений рівень стресу.
- **Вигорання:** Тривала інтенсивна робота з даними може призвести до емоційного виснаження.

Для зменшення негативного впливу цих факторів необхідно:

- **Оптимізація робочого процесу:** Використання автоматизованих інструментів для обробки даних та візуалізації результатів.
- **Регулярні перерви:** Планування регулярних перерв для відпочинку та відновлення сил.
- **Збалансований режим роботи та відпочинку:** Створення здорового балансу між роботою та особистим життям.
- **Підтримка команди:** Створення сприятливого робочого середовища та взаємодопомога серед колег.

6.3. Безпека роботи за комп'ютером

Тривала робота за комп'ютером може призвести до розвитку професійних захворювань. Для зменшення ризику необхідно дотримуватися таких правил:

- **Ергономіка робочого місця:** Правильне розташування монітора, клавіатури та миші.

- **Регулярна фізична активність:** Виконання фізичних вправ під час роботи.
- **Контроль освітлення:** Забезпечення достатнього рівня освітлення робочого місця.

ВИСНОВКИ

Метою даної магістерської роботи було дослідити моделі та засоби для аналізування банківських пропозицій на основі штучних нейронних мереж. Результатом дослідження став експериментально визначена найефективніша модель для навчання нейронної мережі, яка здатна аналізувати різноманітні банківські пропозиції та надавати користувачеві рекомендації для прийняття обґрунтованих фінансових рішень.

У процесі дослідження були виконані наступні завдання:

- **Аналітична частина:** Проведено детальний аналіз стану питання за літературними та іншими джерелами. Визначено актуальність роботи, методів вирішення та постановка задачі.
- **Технологічна частина:** Було визначено технологію підготовки даних, а саме їх збір, очищення, та формування навчального та тестового наборів. Проаналізовано та оцінено чотири алгоритми навчання нейронної мережі.
- **Конструкторська частина:** Розроблено програмне забезпечення системи: загальна архітектура, діаграма послідовностей та станів, визначено схеми бази даних. Було сформовано інструкцію програмісту для роботи над системою. Визначено інформаційну взаємодію системи із зовнішніми джерелами та спосіб розгортання та масштабування системи.
- **Науково-дослідна частина:** Сформульовано наукову проблему з детальним описом мети, гіпотези та завдання дослідження. Сформовано методологію дослідження і як результат було проведено експерименти для оцінки ефективності вибраних алгоритмів навчання та аналіз результатів цих експериментів для визначення найефективнішого алгоритму.

ПЕРЕЛІК ПОСИЛАНЬ

1. ML.NET [Електронний ресурс]. Режим доступу до ресурсу: <https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet>
2. LightGBM [Електронний ресурс]. Режим доступу до ресурсу: <https://lightgbm.readthedocs.io/en/latest/index.html>
3. .NET Aspire [Електронний ресурс]. Режим доступу до ресурсу: <https://devblogs.microsoft.com/dotnet/introducing-dotnet-aspire-simplifying-cloud-native-development-with-dotnet-8/>
4. BankiUA [Електронний ресурс]. Режим доступу до ресурсу: <https://banki.ua/deposit>
5. MinfinUA [Електронний ресурс]. Режим доступу до ресурсу: <https://minfin.com.ua/ua/currency/>
6. Вимоги щодо безпечності контрольно-вимірювального та лабораторного електричного устаткування. Частина 2-020. Додаткові вимоги до лабораторних центрифуг (EN 61010-2-020:1994, IDT): ДСТУ EN 61010-2-020:2005. [Чинний від 2007-01-01]. Київ: Держспоживстандарт України, 2007. IV, 18 с. (Національний стандарт України).
7. Безпека інформації. Вимоги до управління ризиками інформаційною безпекою (ISO/IEC 27001:2013, IDT): ДСТУ ISO/IEC 27001:2014. [Чинний від 2014-03-01]. Київ: Держспоживстандарт України, 2014. IV, 30 с. (Національний стандарт України).
8. Курс лекцій з дисципліни “Системний аналіз видавничих процесів” для студентів інженерного рівня підготовки по спеціальності “Видавництво та поліграфія” (шифр 186) ”/ Укл. доц. Ковівчак Я. В., доц. Лотошинська Н. Д. – Львів-2019. – 32 с.
9. Іванов О.В., Петров С.О. Модель та засоби інформаційної системи аналізу банківських пропозицій з використанням штучних нейронних мереж. Київ: Київ. ун-т, 2023. 120 с.
10. Павлов О.В., Гаврилюк В.В. Інформаційні технології в банківській справі. Львів: ЛНУ, 2021. 240 с.
11. Коваленко В.О., Ігнатенко О.В. Штучні нейронні мережі: основи теорії та застосування. Харків: Основа, 2022. 256 с.
12. Мочурад Л., Цьона І., Побудова діаграм моделювання роботи інформаційної системи розподілу навантаження викладачів // Impactofmodernityonscienceandpractice. Abstractsof XII internationalscientificandpracticalconference. Edmonton, Canada 2020. pp. 105-108.
13. Шаховська Н. Б., Литвин В. В. Проектування інформаційних систем: навчальний посібник / Н. Б. Шаховська, В. В. Литвин. -Львів: 'Магнолія-2006", 2011. - 380 с.
14. Міщенко, В. В. (2023). Модель та засоби інформаційної системи аналізу банківських пропозицій з використанням штучних нейронних мереж. /Київ: Видавництво "К. І.С."
15. Ковальчук, О. М. (2023). Аналіз банківських пропозицій з використанням штучних нейронних мереж. Київ: Видавництво Національного банку України.
16. Левицький В.В. Методичні вказівки для виконання курсового проекту з дисципліни

- «Ідентифікація та моделювання технологічних об'єктів» для студентів спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» / уклад. В. В. Левицький. // ТНТУ. – 2024. – С. 49.
17. Микитишин А.Г. Методичні рекомендації з виконання, оформлення та захисту кваліфікаційних робіт магістрів спеціальності 174 – «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» / ТНТУ ім. І. Пулюя; уклад. А.Г. Микитишин, М.М. Митник, В.В. Левицький, Р.І. Королук – Тернопіль: ТНТУ, 2024. – 82 с.
18. Микитишин А.Г. Методичні рекомендації з виконання, оформлення та захисту кваліфікаційних робіт магістрів спеціальності 151 – «Автоматизація та комп'ютерно-інтегровані технології» / ТНТУ ім. І. Пулюя; уклад. А.Г. Микитишин, М.М. Митник, В.В. Левицький, Р.І. Королук – Тернопіль: ТНТУ, 2022. – 84 с.
19. Левицький В.В. Методичні вказівки до виконання курсового проєкту з дисципліни «Основи автоматизованого проєктування складних об'єктів та систем» для здобувачів освітнього рівня магістр за спеціальністю 174"Автоматизація, комп'ютерно-інтегровані технології та робототехніка" / Уклад. В.В. Левицький, І.С. Дідич – Тернопіль: ТНТУ, 2023. - 44 с.

ДОДАТОК А

Дані для навчання розміром 50

age;salary;MinSum;MaxSum;Currency;MinRate;MaxRate;CreatedAtUtc;MaxTermMonth;MinTermMonth;target

34;62589.75;1000.00;50000000.00;ГРН;5;17;2023-12-03 15:51:27.4240860;30;1;0.25

42;70123.46;100000.00;50000000.00;ГРН;8;18;2023-12-03 15:51:27.5479942;18;1;0.75

27;56984.21;5000.00;50000000.00;ГРН;10.4;16;2023-12-03 15:51:27.5737766;12;1;0.5

33;75824.63;1000.00;50000000.00;ГРН;5;17;2023-12-03 15:51:27.5939881;30;1;0.0

28;65237.89;1000.00;50000000.00;ГРН;14.5;20;2023-12-03 15:51:27.6157374;12;1;0.25

36;81019.28;1000.00;50000000.00;ГРН;10.5;19.5;2023-12-03 15:51:27.6369547;12;1;0.5

29;67481.94;1000.00;50000000.00;ГРН;10.5;19;2023-12-03 15:51:27.6621887;6;3;0.75

45;50213.57;1000.00;50000000.00;ГРН;16.5;19;2023-12-03 15:51:27.6967531;12;3;0.5

31;74382.18;1000.00;50000000.00;ГРН;10.5;19;2023-12-03 15:51:27.7346165;24;1;0.25

37;59874.62;1000.00;50000000.00;ГРН;9.75;18.75;2023-12-03 15:51:27.7591753;1775;1;0.75

39;59963.24;1000.00;50000000.00;ГРН;8;18.75;2023-12-03 15:51:27.7935546;12;1;0.5

54;83329.68;1000.00;50000000.00;ГРН;17;18.5;2023-12-03 15:51:27.8149662;13;3;0.25

32;74321.57;1000.00;50000000.00;ГРН;5;17;2023-12-03 15:51:27.4240860;30;1;0.5

41;67482.36;100000.00;50000000.00;ГРН;8;18;2023-12-03 15:51:27.5479942;18;1;0.75

29;59137.68;5000.00;50000000.00;ГРН;10.4;16;2023-12-03 15:51:27.5737766;12;1;0.25

36;75293.42;1000.00;50000000.00;ГРН;5;17;2023-12-03 15:51:27.5939881;30;1;0.0

30;68628.15;1000.00;50000000.00;ГРН;14.5;20;2023-12-03 15:51:27.6157374;12;1;0.25

38;81129.24;1000.00;50000000.00;ГРН;10.5;19.5;2023-12-03 15:51:27.6369547;12;1;0.5

30;66748.92;1000.00;50000000.00;ГРН;10.5;19;2023-12-03 15:51:27.6621887;6;3;0.75

47;49921.67;1000.00;50000000.00;ГРН;18.5;19;2023-12-03 15:51:27.6967531;12;3;0.5

34;74738.15;1000.00;50000000.00;ГРН;10.5;19;2023-12-03 15:51:27.7346165;24;1;0.25

38;59963.24;1000.00;50000000.00;ГРН;9.75;18.75;2023-12-03 15:51:27.7591753;1775;1;0.75

39;59832.57;1000.00;50000000.00;ГРН;8;18.75;2023-12-03 15:51:27.7935546;12;1;0.5

54;83329.68;1000.00;50000000.00;ГРН;17;18.5;2023-12-03 15:51:27.8149662;13;3;0.25

41;62948.36;5000.00;50000000.00;ГРН;8.4;16;2023-12-03 15:51:27.8392750;30;1;0.5

28;74129.18;1000.00;50000000.00;ГРН;14;19;2023-12-03 15:51:27.8695833;12;1;0.25

35;60823.57;1000.00;50000000.00;ГРН;9.75;17;2023-12-03 15:51:27.8967487;24;1;0.75

32;68823.46;1000.00;50000000.00;ГРН;5;17;2023-12-03 15:51:27.9296865;30;1;0.0

33;75481.94;1000.00;50000000.00;ГРН;10.5;19.5;2023-12-03 15:51:27.9641703;12;1;0.25

37;81249.36;1000.00;50000000.00;ГРН;10;19;2023-12-03 15:51:27.9918858;12;1;0.5

29;67238.15;1000.00;50000000.00;ГРН;10.5;18.5;2023-12-03 15:51:28.0215812;6;3;0.75

48;50924.63;1000.00;50000000.00;ГРН;18.5;19;2023-12-03 15:51:28.0522941;12;3;0.5

31;74984.72;1000.00;50000000.00;ГРН;10.5;19;2023-12-03 15:51:28.0831612;24;1;0.25

40;58849.24;1000.00;50000000.00;ГРН;9.75;18.75;2023-12-03 15:51:28.1181157;1775;1;0.75

36;59782.36;1000.00;50000000.00;ГРН;8;18.75;2023-12-03 15:51:28.1464363;12;1;0.5

53;83539.68;1000.00;50000000.00;ГРН;17;18.5;2023-12-03 15:51:28.1768232;13;3;0.25

31;69742.18;1000.00;50000000.00;ГРН;5;17;2023-12-03 15:51:28.2043925;30;1;0.5

39;63492.36;1000.00;50000000.00;ГРН;8;18;2023-12-03 15:51:28.2356328;18;1;0.75

27;55938.15;1000.00;50000000.00;ГРН;10.4;16;2023-12-03 15:51:28.2681459;12;1;0.25

34;76019.28;1000.00;50000000.00;ГРН;5;17;2023-12-03 15:51:28.2978068;30;1;0.0

30;64481.94;1000.00;50000000.00;ГРН;14;19;2023-12-03 15:51:28.3292537;12;1;0.25

38;81829.24;1000.00;50000000.00;ГРН;10.5;19.5;2023-12-03 15:51:28.3615655;12;1;0.5

30;67848.92;1000.00;50000000.00;ГРН;10.5;18.5;2023-12-03 15:51:28.3904035;6;3;0.75

47;51291.67;1000.00;50000000.00;ГРН;18.5;19;2023-12-03 15:51:28.4216822;12;3;0.5

35;75828.36;1000.00;50000000.00;ГРН;10.5;19;2023-12-03 15:51:28.4523154;24;1;0.25

39;60832.57;1000.00;50000000.00;ГРН;9.75;18.75;2023-12-03 15:51:28.4832913;1775;1;0.75

39;60663.24;1000.00;50000000.00;ГРН;8;18.75;2023-12-03 15:51:28.5146774;12;1;0.5

54;83029.68;1000.00;50000000.00;ГРН;17;18.5;2023-12-03 15:51:28.5466472;13;3;0.25

Дані для навчання розміром 100

age;salary;MinSum;MaxSum;Currency;MinRate;MaxRate;MaxTermMonth;MinTermMonth;target

34;62589.75;1000.00;50000000.00;ГРН;5;17;30;1;0.25
42;70123.46;100000.00;50000000.00;ГРН;8;18;18;1;0.75
27;56984.21;5000.00;50000000.00;ГРН;10;4;16;12;1;0.5
33;75824.63;1000.00;50000000.00;ГРН;5;17;30;1;0.0
28;65237.89;1000.00;50000000.00;ГРН;14.5;20;12;1;0.25
36;81019.28;1000.00;50000000.00;ГРН;10.5;19.5;12;1;0.5
29;67481.94;1000.00;50000000.00;ГРН;10.5;19;6;3;0.75
45;50213.57;1000.00;50000000.00;ГРН;16.5;19;12;3;0.0
31;74382.18;1000.00;50000000.00;ГРН;10.5;19;24;1;0.25
37;59874.62;1000.00;50000000.00;ГРН;9.75;18.75;1775;1;0.75
39;59963.24;1000.00;50000000.00;ГРН;8;18.75;12;1;0.5
54;83329.68;1000.00;50000000.00;ГРН;17;18.5;13;3;0.0
32;74321.57;1000.00;50000000.00;ГРН;5;17;30;1;0.5
41;67482.36;100000.00;50000000.00;ГРН;8;18;18;1;0.75
29;59137.68;5000.00;50000000.00;ГРН;10.4;16;12;1;0.25
36;75293.42;1000.00;50000000.00;ГРН;5;17;30;1;0.0
30;68628.15;1000.00;50000000.00;ГРН;14.5;20;12;1;0.5
38;81129.24;1000.00;50000000.00;ГРН;10.5;19.5;12;1;0.25
30;66748.92;1000.00;50000000.00;ГРН;10.5;19;6;3;0.75
47;49921.67;1000.00;50000000.00;ГРН;18.5;19;12;3;0.5
34;74738.15;1000.00;50000000.00;ГРН;10.5;19;24;1;0.25
38;59963.24;1000.00;50000000.00;ГРН;9.75;18.75;1775;1;0.75
39;59832.57;1000.00;50000000.00;ГРН;8;18.75;12;1;0.0
54;83329.68;1000.00;50000000.00;ГРН;17;18.5;13;3;0.25
41;62948.36;5000.00;50000000.00;ГРН;8.4;16;30;1;0.5
28;74129.18;1000.00;50000000.00;ГРН;14;19;12;1;0.25
35;60823.57;1000.00;50000000.00;ГРН;9.75;17;24;1;0.75
32;68823.46;1000.00;50000000.00;ГРН;5;17;30;1;0.0
33;75481.94;1000.00;50000000.00;ГРН;10.5;19.5;12;1;0.25
37;81249.36;1000.00;50000000.00;ГРН;10;19;12;1;0.5
29;67238.15;1000.00;50000000.00;ГРН;10.5;18.5;6;3;0.75
48;50924.63;1000.00;50000000.00;ГРН;18.5;19;12;3;0.0
31;74984.72;1000.00;50000000.00;ГРН;10.5;19;24;1;0.25
40;58849.24;1000.00;50000000.00;ГРН;9.75;18.75;1775;1;0.75
36;59782.36;1000.00;50000000.00;ГРН;8;18.75;12;1;0.5
53;83539.68;1000.00;50000000.00;ГРН;17;18.5;13;3;0.0
31;69742.18;1000.00;50000000.00;ГРН;5;17;30;1;0.25
39;63492.36;1000.00;50000000.00;ГРН;8;18;18;1;0.75
27;55938.15;1000.00;50000000.00;ГРН;10.4;16;12;1;0.5
34;76019.28;1000.00;50000000.00;ГРН;5;17;30;1;0.0
30;64481.94;1000.00;50000000.00;ГРН;14;19;12;1;0.25
38;81829.24;1000.00;50000000.00;ГРН;10.5;19.5;12;1;0.5
30;67848.92;1000.00;50000000.00;ГРН;10.5;18.5;6;3;0.75
47;51291.67;1000.00;50000000.00;ГРН;18.5;19;12;3;0.5
35;75828.36;1000.00;50000000.00;ГРН;10.5;19;24;1;0.0
39;60832.57;1000.00;50000000.00;ГРН;9.75;18.75;1775;1;0.75
39;60663.24;1000.00;50000000.00;ГРН;8;18.75;12;1;0.5
54;83029.68;1000.00;50000000.00;ГРН;17;18.5;13;3;0.25
28;69325.78;900.00;40000000.00;ГРН;7;15;20;2;0.4
32;54378.92;700.00;35000000.00;ГРН;6;12;15;1;0.3
44;78219.34;2000.00;45000000.00;ГРН;8;18;24;3;0.6
50;62389.47;1100.00;50000000.00;ГРН;9;20;18;2;0.5
29;68912.84;1300.00;47000000.00;ГРН;6.5;17.5;22;1;0.35
35;73923.18;1200.00;39000000.00;ГРН;7.5;19;20;2;0.45
47;51982.63;1500.00;42000000.00;ГРН;9.5;20;25;3;0.5
33;78237.41;1000.00;41000000.00;ГРН;8;19;19;2;0.0
41;52938.47;1200.00;38000000.00;ГРН;7;15;16;1;0.25
38;78549.82;800.00;47000000.00;ГРН;7.5;20;21;2;0.5
36;61029.38;1400.00;42000000.00;ГРН;9;19;23;3;0.75
50;53628.97;1300.00;49000000.00;ГРН;8.5;18.5;17;2;0.0
32;74921.83;1500.00;40000000.00;ГРН;7;17;19;1;0.25
43;69019.48;1100.00;46000000.00;ГРН;8;19;22;2;0.5
27;78347.91;1200.00;43000000.00;ГРН;6.5;17.5;18;1;0.75
35;61829.25;900.00;39000000.00;ГРН;7;16;24;3;0.0
29;65938.47;1000.00;42000000.00;ГРН;8.5;20;25;2;0.25
47;50983.84;1100.00;47000000.00;ГРН;9;19;20;1;0.5
38;73349.19;1400.00;48000000.00;ГРН;6;18.5;23;2;0.75
33;78341.86;900.00;46000000.00;ГРН;7.5;20;19;3;0.0
40;58943.52;1500.00;45000000.00;ГРН;8;17;18;1;0.25
48;52389.23;1200.00;49000000.00;ГРН;8.5;20;21;2;0.5
44;75293.68;1300.00;47000000.00;ГРН;9.5;18;17;1;0.75
35;61029.58;1100.00;44000000.00;ГРН;8;18.5;22;2;0.0
30;68912.45;1200.00;45000000.00;ГРН;7;17;19;1;0.25
32;73929.18;1300.00;46000000.00;ГРН;7.5;16.5;20;3;0.5
33;52489.72;900.00;48000000.00;ГРН;8;17.5;21;2;0.75
39;69318.34;1500.00;47000000.00;ГРН;9;19;18;1;0.0
29;64932.58;1100.00;41000000.00;ГРН;7.5;18;23;2;0.25
38;79429.72;1400.00;49000000.00;ГРН;8;20;17;1;0.5
41;53189.56;1200.00;44000000.00;ГРН;9.5;18.5;22;3;0.75
45;50938.47;1300.00;45000000.00;ГРН;8.5;17;19;2;0.0
47;68129.38;1100.00;46000000.00;ГРН;7;20;25;1;0.25
50;53928.65;900.00;47000000.00;ГРН;7.5;18.5;18;2;0.5
44;78341.73;1200.00;48000000.00;ГРН;8;17.5;19;3;0.75
40;60589.72;1400.00;49000000.00;ГРН;9;20;21;1;0.0
36;78319.45;1300.00;45000000.00;ГРН;8.5;19;23;2;0.25
39;64921.84;1100.00;46000000.00;ГРН;7;18;24;1;0.5
30;73218.29;900.00;42000000.00;ГРН;7.5;17.5;22;2;0.75
48;51939.47;1500.00;47000000.00;ГРН;9.5;18;20;3;0.0
50;78324.56;1200.00;48000000.00;ГРН;8;17;21;1;0.25
35;53218.29;1300.00;43000000.00;ГРН;8.5;18.5;19;2;0.5
42;51029.75;1100.00;45000000.00;ГРН;7;20;23;3;0.75
38;78349.72;900.00;49000000.00;ГРН;9;17.5;18;1;0.0
41;60329.18;1400.00;44000000.00;ГРН;8.5;19;25;2;0.25
45;67912.38;1200.00;47000000.00;ГРН;7.5;20;24;1;0.5
50;54389.56;1300.00;46000000.00;ГРН;9.5;18.5;18;2;0.75
36;78923.49;1100.00;43000000.00;ГРН;8;19;23;3;0.0
30;74123.91;1500.00;42000000.00;ГРН;7.5;17;21;1;0.25
31;53928.71;1200.00;45000000.00;ГРН;8.5;18.5;20;2;0.5
39;79123.86;1300.00;47000000.00;ГРН;9;19;22;3;0.75
44;79023.18;900.00;48000000.00;ГРН;8;17.5;18;1;0.0