

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Факультет інформатики кафедра інформатики



Магістерська робота
освітній ступінь – магістр

на тему: «ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ
ОЦІНКИ КРЕДИТОСПРОМОЖНОСТІ»

Виконала студентка 2-го року навчання,
спеціальності

122 Комп'ютерні науки

Чумаченко Олександра Романівна

Керівник Нагірна Алла Миколаївна

кандидат фізико-математичних наук, доцент

Рецензент _____

Магістерська робота захищена

з оцінкою _____

Секретар ЕК _____

«_____» _____ 2023 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
к.ф.-м.н, доц.
_____ С. С. Гороховський
(підпис)
„____” _____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на магістерську роботу

студенту 2-го курсу, факультету інформатики
Чумаченко Олександрі Романівні

Розробити Прототип інформаційної системи підтримки прийняття рішень для оцінки
кредитоспроможності

Вихідні дані:

- фінансові показники фізичної особи (відношення загального боргу до загальних доходів, відношення взятого боргу до розміру доступного ліміту);
- кредитна історія фізичної особи;

Зміст ТЧ до магістерської роботи:

Зміст
Анотація
Вступ
1 Теоретичні основи та постановка задачі
2 Проектування інформаційної системи
3 Опис розробки системи
4 Результати
Висновки
Список літератури

Дата видачі «__» _____ 2022 р. Керівник

(підпис)

Завдання отримав

(підпис)

Календарний плану виконання магістерської роботи

Тема: Інформаційна система підтримки прийняття рішень для оцінки кредитоспроможності

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу.	02.11.2022	
2.	Огляд технічної літератури за темою роботи.	15.11.2022	
3.	Пошук датасетів	25.11.2022	
3.	Підбір технологій для виконання.	30.12.2022	
4.	Реалізація моделей машинного навчання. Тренування моделей та дослідження результатів	15.01.2023	
5.	Проектування архітектури інформаційної системи.	15.02.2023	
6.	Реалізація інформаційної системи прийняття рішень.	30.03.2023	
7.	Написання пояснювальної роботи.	20.04.2023	
8.	Створення слайдів для доповіді та написання доповіді.	1.05.2023	
8.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист магістерської роботи.	15.05.2023	
10.	Корегування роботи за результатами попереднього захисту.	21.05.2023	
11.	Остаточне оформлення пояснювальної роботи та слайдів.	4.06.2023	
12.	Захист магістерської роботи (проекту)	12.06.2023	

Студент _____

Керівник _____

«_____» _____ 2022 р

Зміст

Анотація	4
Вступ	5
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА ПОСТАНОВКА ЗАДАЧІ.....	7
1.1 Основні поняття	7
1.2 Аналіз літератури	8
1.3 Постановка задачі.....	8
1.4 Висновок за главою 1	9
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	10
2.1 Вибір наборів даних.....	10
2.1.1 Набір даних «Give Me Some Credit»	10
2.1.2 Набір даних «German Credit data»	11
2.2 Функціональні вимоги	15
2.3 Проектування системи.....	18
2.3.1 Проектування взаємодії користувачів	18
2.3.2 Оцінювання кредитоспроможності.....	19
2.4 Проектування моделі даних	21
2.5 Вибір програмних засобів	22
2.6 Висновок за главою 2.....	23
РОЗДІЛ 3. ОПИС РОЗРОБКИ СИСТЕМИ.....	24
3.1 Архітектура системи	24
3.1.1 Фреймворк ABP	25
3.1.2 Domain-driven design	26
3.1.3 Рівень інфраструктури.....	28
3.1.4 Рівень домену	28
3.1.5 Прикладний рівень.....	29
3.1.6 Рівень презентації	31
3.2 ML.NET	33
3.2.1 Model Builder	34
3.2.2 Навчання моделей поза Model Builder.....	35
3.2.3 Глибинне навчання на ML.NET	36

3.2.4 Опис алгоритмів	36
3.3 Висновок за главою 3.....	39
РОЗДІЛ 4. РЕЗУЛЬТАТИ	40
4.1 Дослідження моделей	40
4.2 Програмний інтерфейс застосунку.....	42
4.3 Користувацький інтерфейс застосунку.....	47
4.4 Висновок за главою 4.....	55
Висновки	56
Список використаної літератури	57

Анотація

У даній роботі описується розробка інформаційної системи підтримки прийняття рішень для оцінки кредитоспроможності фізичної особи на основі поєднання методів кредитного скорингу та моделей машинного навчання реалізованих в середовищі .NET за допомогою технології ML.NET.

Результатом роботи є прототип системи, що забезпечує підтримку прийняття рішень шляхом передбачення моделями можливих фінансових труднощів особи та кредитоспроможності особи незалежно і враховуючи передбачення першої моделі, а також наданням інформації щодо результатів кредитного скорингу.

Ключові слова: кредитоспроможність, ML.NET, Model Builder, Abp Framework, DDD.

Вступ

У зв'язку з пандемією та військовим станом попит на користування електронними транзакцій зростає, відповідно зростає й кількість операцій з грошовими одиницями. Аналогічно враховуючи події сьогодення збільшується і актуальність кредитування. Тому є доцільним розглядати надання кредитів фізичним особам за допомогою певних технологій, що спрощують і прискорюють процес. В кожній фінансовій установі що надає послуги кредитування вже існують свої системи та правила обрахунку кредитоспроможності. Всі вони мають спиратися на певні постанови НБУ та відповідно до них враховувати при оцінюванні якісні показники (такі як вік, сімейний стан, тощо), кількісні (розмір заробітної плати, кількість заощаджень на рахунку, тощо) та звісно враховуючи кредитну історію.

Сьогодні популярним методом є кредитний скоринг, але поширення також набуває обрахування кредитоспроможності за допомогою методів машинного навчання. Унікальність системи, що розроблялась є поєднання методів класичного скорингу з обчисленням кредитоспроможності методами машинного навчання.

Метою роботи є розробка прототипу інформаційної системи підтримки прийняття рішень для оцінки кредитоспроможності фізичної особи, що надає рекомендації чи варто надавати кредит особі.

Об'єктом дослідження є автоматизація обрахунку кредитоспроможності фізичної особи.

Предметом дослідження є розробка моделей машинного для оцінювання кредитоспроможності з використанням можливостей технології ML.NET.

Результатом роботи є прототип інформаційної системи прийняття рішень, що пропонує користувачу у відповідь на внесені дані інформацію про кредитний бал клієнта, обрахованого за допомогою кредитного скорингу на базі вимог НБУ та рекомендації щодо надання кредиту, тобто ймовірність того що людина цей кредит

успішно виплатить, вирахувані на основі кредитної історії та певних показників, за допомогою моделей машинного навчання.

Прототип системи реалізовано мовою програмування C# з використанням технології Asp Framework в середовищі розробки Visual Studio 2022. Для роботи з даними використовувався Entity Framework Core та локальна система управління даними на базі SQLServer. Для розробки та тренування моделей машинного навчання була використана платформа ML.NET. Користувацький інтерфейс реалізовано за допомогою Razor Pages.

Робота складається зі вступу, чотирьох розділів та списку використаних джерел.

В першому розділі розкриваються основні поняття, що використовуються в межах роботи, проводиться аналіз літератури та визначається поставлена задача.

Другий розділ описує процес проектування системи: описуються обрані набори даних для машинного навчання, визначаються функціональні вимоги системи, проектується взаємодія клієнтів з системою та схема оцінювання кредитоспроможності, описується модель даних та вибір програмних засобів.

Третій розділ присвячено опису архітектури системи та опису обраної платформи для машинного навчання та алгоритмів для створення моделей.

В четвертому розділі роботи описуються отримані результати моделей машинного навчання, програмний інтерфейс застосунку та наведені приклади користувацького інтерфейсу.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Основні поняття

Система підтримки прийняття рішень – це комп'ютеризована система, яка, аналізуючи інформацію, здатна робити прогнози, враховуючи які будуть прийняті відповідні рішення [1].

Кредитоспроможність – це наявність у позичальника банку передумов для отримання кредиту та його здатність повернути кредит і відсотки за ним у повному обсязі та в обумовлені договором строки [2].

Поточна платоспроможність – це поточний стан розрахунків особи, що оцінюється на підставі даних про її фінансові потоки.

Кредитна історія – це набір інформації про особу, який її ідентифікує, відомостей про виконання нею кредитних зобов'язань та іншої інформації, що є відкритою [3].

Кредитний скоринг – це статистичний метод дослідження, що використовується для оцінки кредитного ризику позичальника [4].

Оцінювання кредитоспроможності фізичної особи відбувається шляхом оцінювання її кількісних та якісних характеристик, що підтверджуються документально. При обрахунку кредитоспроможності методами кредитного скорингу, кожна з характеристик зіставляється з певними балами. Відповідно до нормативних актів НБУ, фінансові установи можуть встановлювати на свій розсуд вагові коефіцієнти для оцінки кожної з характеристик та доповнювати кредитний скоринг власними показниками. Рекомендованим набором якісних характеристик прийнято вважати наявність майна, постійної роботи, сімейний стан, вік, оцінку кредитної історії, до кількісних показників належить інформація про фінансовий стан позичальника та забезпечення кредиту [5].

1.2 Аналіз літератури

Оцінка кредитоспроможності фізичних осіб часто досліджується в різних роботах. У статтях присвячених дослідженню методів обрахунку кредитоспроможності Терентьев О. М., Бідюк П. І., Гасанов А.С., Басараб А. В., Свердел К. О., Миронова О. В., Охрименко І.Б. пропонують методи удосконалення оцінювання використовуючи для цього дерева рішень, диференційований підхід, та інші методи [6-7]. Також існують роботи які присвячені використанню методів машинного навчання для виконання прогнозування кредитоспроможності. Так у роботі Гурнарсона [8] розглядається доцільність використання методів глибинного навчання при оцінці кредитного скорингу, а Девід Вест для виконання цієї задачі в своїй статі порівнює п'ять моделей нейронних мереж [9].

1.3 Постановка задачі

Розробити прототип інформаційної системи прийняття рішень для оцінювання кредитоспроможності. Система повинна надавати рекомендації щодо відкриття кредиту для фізичної особи, спираючись на кількісні і якісні показники позичальника, за допомогою кредитного скорингу та методів машинного навчання. Кінцевий результат по рішенню з відкриття кредиту надається користувачеві. Крім того, система повинна мати можливість зберігати дані, що були використані під час оцінювання та мати можливість, у випадку підтвердження відкриття кредиту, зберегти його результат – чи був успішно виплачений кредит. Вхідними даними для інформаційної системи є кількісні та якісні показники фізичної особи (майно, робота, дохід, соціальний та сімейний стан), кредитна історія та деякі фінансові показники, які надаються до системи фінансовою установою.

1.4 Висновок за главою 1

В даному розділі було введено основні поняття та терміни, надано стислий опис процесу оцінки кредитоспроможності фізичної особи. Було проаналізовано роботи присвячені дослідження методів оцінювання кредитоспроможності, зокрема з використанням технологій машинного навчання та була визначена постановка задачі.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Вибір наборів даних

Проаналізувавши літературу можна дійти висновку що найчастіше для задач з розробки моделей по оцінюванні кредитоспроможності використовуються наступні набори даних: «Give Me Some Credit» та «German Credit data».

2.1.1 Набір даних «Give Me Some Credit»

Оцінювання кредитоспроможності особи необхідне для передбачення, чи зможе повернути дана особа кредит. В першу чергу на це впливає її фінансовий стан. Відповідно, важливим кроком для якісного обрахунку кредитоспроможності є передбачення фізичного стану особи в найближчому майбутньому. Основна мета даного датасету була в визначенні ймовірності того, що особа зазнає фінансових труднощів протягом наступних двох років. При його створенні було зібрано історичні дані про 250 000 позичальників. В тренувальному наборі, який буде використовуватись для роботи далі, зібрано 150 000 записів [10].

2.1.1.1 Опис атрибутів

Датасет складається з 12 колонок:

- `SeriousDlqin2yrs` – чи зазнає особа фінансових труднощів в найближчі два роки (1 або 0);
- `RevolvingUtilizationOfUnsecuredLines` – загальний баланс за кредитними картками та особистими кредитними лініями, за винятком нерухомості та заборгованості без виплат, такі кредити на авто, поділений на суму кредитних лімітів (відсоток);
- `Age` – вік позичальника;

- `NumberOfTime30-59DaysPastDueNotWorse` – кількість разів, коли позичальник прострочував оплату на 30-59 днів, але не більше за останні 2 роки (ціле число);
- `DebtRatio` – щомісячні боргові платежі, аліменти, витрати на проживання, поділені на місячний валовий дохід (відсоток);
- `MonthlyIncome` – місячний дохід (дійсне число);
- `NumberOfOpenCreditLinesAndLoans` – кількість відкритих кредитних ліній і позик (ціле число);
- `NumberOfTimes90DaysLate` – кількість разів, коли позичальник прострочував оплату на 90 днів або більше (ціле число);
- `NumberRealEstateLoansOrLines` – кількість іпотечних кредитів і кредитів на нерухомість, включаючи кредитні лінії для капіталу (ціле число);
- `NumberOfTime60-89DaysPastDueNotWorse` – кількість разів, коли позичальник прострочував сплату на 60-89 днів, але не гірше за останні 2 роки (ціле число);
- `NumberOfDependents` – кількість утриманців у сім'ї (дружина, діти тощо).

Загалом дані з датасету відповідають інформації яку можна отримати звернувшись в Українське бюро кредитних історій, окрім даних про місячний дохід, кількість утриманців, які має надати клієнт та `RevolvingUtilizationOfUnsecuredLines` і `DebtRatio`, які повинні бути обраховані банком.

2.1.2 Набір даних «German Credit data»

Датасет створений професором Dr. Hans Hofmann. Датасет містить 1000 записів. Складається з 18 атрибутів, з яких 12 є категоріями. Набір даних може бути використаним для передбачення чи варто надавати кредит особі.

2.1.2.1 Опис атрибутів

Status – статус наявного поточного рахунку. Атрибут залежить від призначення заробітної плати не менше 1 року. Значення атрибуту заковані наступним чином:

- A11 : відсутня заробітна плата не менше ніж за 1 рік,
- A12 : заробітна плата до 200 DM,
- A13 : заробітна плата вища 200 DM,
- A14 : відсутність поточного рахунку.

Duration (month) – тривалість кредиту у місяцях.

Credit history – оцінка кредитної історії. Значення атрибуту заковані наступним чином:

- A30 : без кредитів / всі кредити повернуті належним чином;
- A31 : всі кредити в цьому банку повернуті вчасно;
- A32 : наявні кредити, належно повернуті до теперішнього часу;
- A33 : затримка погашення в минулому;
- A34 : інші кредити (не в цьому банку).

Purpose – мета кредиту. Атрибут закований наступним чином:

- A40 : машина (нова);
- A41 : машина (потримана);
- A42 : меблі/обладнання;
- A43 : побутові прилади (телевізор/комп'ютер);
- A44 : побутові прилади;
- A45 : ремонт;
- A46 : навчання;
- A47 : відпустка;
- A48 : перепідготовка;
- A49 : бізнес;

- A410 : інші.

Credit amount – розмір кредиту.

Savings account – оцінка ощадного рахунку/облігацій. Атрибут заведений наступним чином:

- A61 : загальна сума менша 100 DM;
- A62 : загальна сума менша 500 DM;
- A63 : загальна сума менша 1000 DM;
- A64 : загальна сума більша 1000 DM;
- A65 : ощадні рахунки відсутні або невідомі.

Present employment since – кількість років на поточній роботі. Атрибут заведений наступним чином:

- A71 : безробітний;
- A72 : менше року;
- A73 : менше 4 років;
- A74 : менше 7 років;
- A75 : більше 7 років.

Installment rate in percentage of disposable income – розстрочка у відсотках від наявного доходу.

Personal status and sex – сімейний стан і стать, заведений наступним чином:

- A91 : чоловік, розлучений;
- A92 : жінка, розлучена/заміжня;
- A93 : чоловік, неодружений;
- A94 : чоловік, одружений/вдовець;
- A95 : жінка, незаміжня.

Other debtors / guarantors – інші поручителі, заведений наступним чином:

- A101 : відсутні;

- A102 : спільний кредит;
- A103 : поручитель.

Present residence since – кількість років проживає на одному місці.

Property – майно. Закодовано як:

- A121 : нерухомість;
- A122 : (якщо відсутня нерухомість) ощадний договір/страхування життя;
- A123 : (якщо відсутня A121/A122): машина;
- A124 : невідоме майно або відсутнє.

Age in years – вік позичальника.

Housing – поточно проживає. Закодовано наступним чином:

- A151 : орендує;
- A152 : власне;
- A153 : проживає безкоштовно.

Number of existing credits at this bank – кількість поточних кредитів в даному банку.

Job – тип роботи. Закодовано наступним чином:

- A171 : безробітний/ некваліфікований – нерезидент;
- A172 : некваліфікований – резидент;
- A173: кваліфікований працівник / посадова особа;
- A174: менеджмент/самозайнятий/висококваліфікований працівник/посадовець.

Number of people being liable to provide maintenance for – кількість утриманців.

Foreign worker – чи працює на закордонні компанії

- A201 : так;
- A202 : ні.

Фінансові показники в наборі даних в оригіналі визначались німецькою маркою (DM). Для спрощення було прийнято вважати в даній роботі ці показники в євро.

2.2 Функціональні вимоги

В системі можна виділити наступні сутності:

- Користувач – співробітник фінансової установи, що використовує систему. Відповідно до рівня дозволів може мати право створювати нових клієнтів, відкривати, підтверджувати, закривати кредити, переглядати редагувати та видаляти інформацію.
- Клієнт – фізична особа, що бажає відкрити кредит.
- Кредит – сутність, що складається з загальної інформації про розмір, ставку, валюту та термін позики, а також інформації про клієнта, результат обрахунку кредитоспроможності.

Система повинна зберігати наступну інформацію про клієнта:

- Загальна інформація (прізвище, ім'я, по-батькову, реєстраційний номер облікової книжки платника податків, дату народження);
- Інформація про адресу (країна, індекс, регіон, місто, вулиця, номер будинку, за наявності номер квартири та додаткову інформацію);
- Контактна інформація (номери телефонів);
- Персональна інформація (сімейний стан, рівень та тип освіти);
- Інформацію про місце роботи (займана посада, тип організації, дата початку роботи та дата завершення, дохід);
- Інформацію про майно (тип майна, оціночна вартість, дата початку володіння та дата завершення);
- Інформацію про фінансовий стан (загальні витрати, фінансові показники, розмір депозитів);

Оновлення інформації про клієнта (окрім загальної та інформації про адреси) повинно відбуватись шляхом додання нових значень та встановлення їх як актуальних, редагування існуючої інформації не допускається. Дана інформація надається системі від клієнта та/або від фінансової установи, що використовує дану систему.

Система повинна надавати та зберігати інформацію про кредитну історію клієнта. В розрізі даної роботи дані з кредитної історії генерується випадковим чином. Кредитна історія повинна містити в собі інформацію про кількість активних кредитів та інформацію по затриманню виплат по заборгованості.

Про кредит система має зберігати наступну інформацію:

- Загальна інформація (розмір, термін, ставка, валюта);
- Інформація про заставу (тип застави, розмір застави);
- Інформацію про додаткові гарантії (тип гарантій);
- Результат кредитного запиту;
- Результат кредиту при закритті;
- Коментарі;
- Посилання на кредитну історію;
- Посилання на інформацію про клієнта, що використовується при наданні кредиту.

Система повинна реалізовувати авторизацію користувачів та надавати наступні дозволи:

- Створення та редагування нових користувачів;
- Додавання нових клієнтів та інформації про них;
- Редагування даних клієнтів;
- Перегляд клієнтів;
- Створення запитів на відкриття кредитів;
- Створення запитів на отримання кредитної історії;

- Перегляд всіх кредитів установи;
- Додавання результату кредитного запиту;
- Закриття кредитів;
- Додавання нового курсу валют.

Інформаційна система повинна надавати користувачу наступні функції:

- Вхід до системи використовуючи власний пароль та логін;
- Реєстрацію нових користувачів (за умови наявності необхідних дозволів);
- Перегляд та створення нових клієнтів (за умови наявності необхідних дозволів);
- Внесення та редагування інформації про клієнта (за умови наявності необхідних дозволів);
- Отримання кредитної історії клієнта (за умови наявності необхідних дозволів);
- Створення нового кредитного запиту (за умови наявності необхідних дозволів);
- Обрахунок кредитоспроможності клієнта (за умови наявності необхідних дозволів);
- Створення рішення щодо кредитного запиту (за умови наявності необхідних дозволів);
- Перегляд активних/закритих кредитів певного клієнта (за умови наявності необхідних дозволів);
- Перегляд кредитних запитів певного клієнта, що очікують на підтвердження (за умови наявності необхідних дозволів);
- Перегляд усіх кредитних запитів певного клієнта (за умови наявності необхідних дозволів);
- Перегляд активних/закритих кредитів установи(за умови наявності необхідних дозволів);

- Перегляд кредитних запитів установи, що очікують на підтвердження (за умови наявності необхідних дозволів);
- Перегляд усіх кредитних запитів установи (за умови наявності необхідних дозволів);
- Перегляд курсу валют (за умови наявності необхідних дозволів);
- Додання актуального курсу валют по якому мають конвертуватись кредитні запити (за умови наявності необхідних дозволів).

2.3 Проектування системи

2.3.1 Проектування взаємодії користувачів

Враховуючи функціональні вимоги системи можна побачити окремі основні потоки взаємодії користувачів з системою: реєстрація клієнта, створення кредитного запиту та його подальший супровід. На рисунку 2.1 зображено діаграму взаємодії користувача з системою при створенні клієнта, або оновленні його даних.

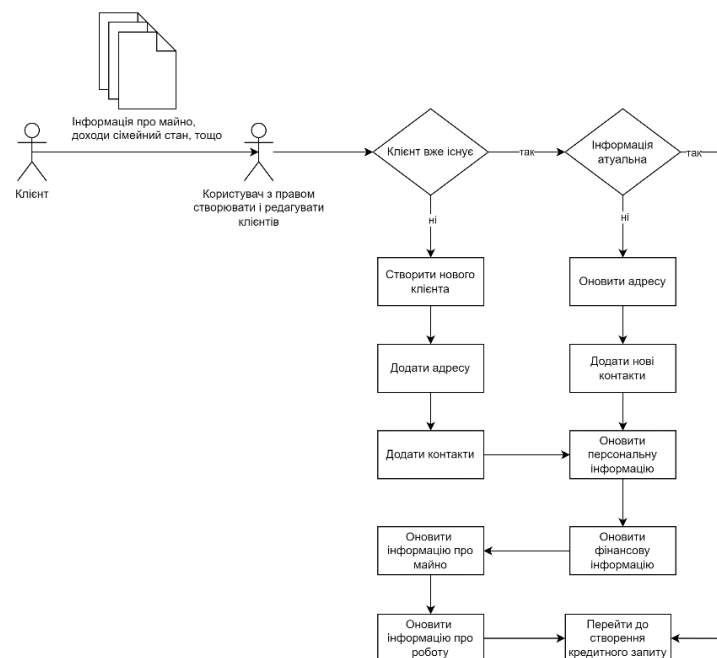


Рисунок 2.1 Діаграма взаємодії користувача при створенні клієнта

На рисунку 2.2 зображено можливу взаємодію користувача з кредитним запитом. Передбачається, що в кожному з потоків з одним кредитним засобом може працювати окремий користувач з визначеними правами доступу, що може бути корисним, наприклад, при стажуванні нових працівників.

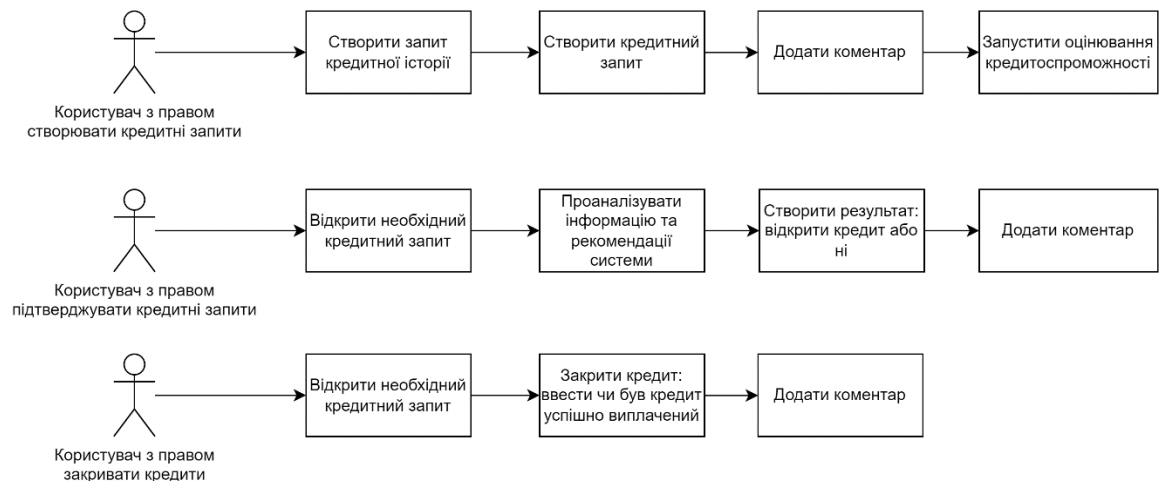


Рисунок 2.2 Діаграма взаємодії клієнта з кредитним запитом

Під час створення запиту до кредитної історії в даній роботі кредитна історія клієнта генерується випадковим чином. Розбір файлу-відповіді кредитної історії в форматі XML було реалізовано в попередній роботі [11].

В результаті оцінювання кредитоспроможності система видає рекомендації по кредитному запиту. Остаточне рішення про відкриття кредиту виконує користувач.

2.3.2 Оцінювання кредитоспроможності

Оцінка кредитоспроможності в даній роботі складається з двох незалежних модулів: модуль кредитного скорингу та модуль оцінювання за допомогою моделей машинного навчання. На рисунку 2.3 зображено процес оцінювання кредитоспроможності системою.

На вхід подається інформація про клієнта, кредитний запит та зведена інформація про кредитну історію клієнта. Після чого дані оброблюються, проводяться

обрахунки такі як обрахунок поточної платоспроможності, ставка розстрочки у відсотках від наявного доходу, нараховані відсотки, розмір щомісячних зобов'язань. Наступним кроком є співставлення даних з вхідними типами кредитного скорингу та вхідними типами моделей машинного навчання.

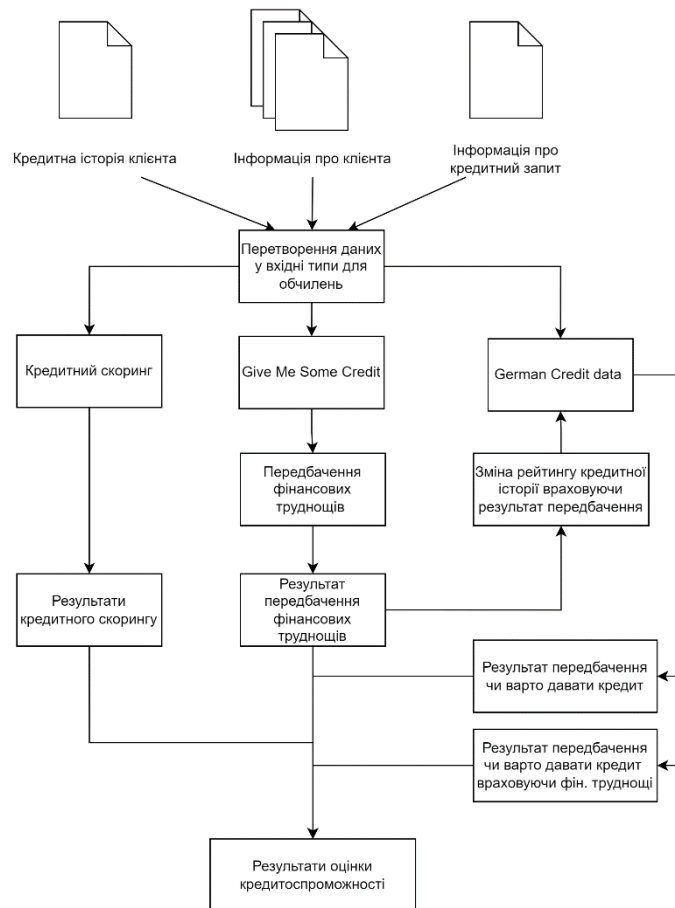


Рисунок 2.3 Процес оцінювання кредитоспроможності

Для виконання скорингу дані групуються за типом наданої інформації: персональна (якісні показники), фінансова (кількісні показники) та інформація про кредитний запит. Механізм даного скорингу був реалізований в попередній роботі [11]. В даній роботі був проведений рефакторинг коду. Результатом роботи скорингу є окремі оцінки наданої персональної, фінансової та кредитної інформації, загальний кредитний бал та рекомендований клас позичальника.

Для отримання результату за допомогою моделей машинного навчання було створено модуль, що групує вхідні дані відповідно до атрибутів обраних датасетів, та

отримує передбачення щодо можливих фінансових труднощів позичальника та передбачення щодо успішного повернення боргу позичальником. Окрім того, було помічено, що модель побудована на наборі даних “Give Me Some Credit” майже повністю використовує інформацію надану з кредитної історії клієнта. Виходячи з цього було висунуто гіпотезу про можливе покращення точності роботи другої моделі при врахуванні результатів першої. Для досягнення цієї мети було додано ще одне передбачення другою моделлю, де оцінка кредитної історії, що отримана від стороннього сервісу, буде змінюватись в залежності від прогнозу фінансових труднощів. Так якщо прогноз по клієнту передбачає можливі складнощі в фінансовому плані в наступні два роки і впевненість в цьому у прогнозу більше 60%, то рейтинг кредитної історії зменшується на 100 балів, і відповідно, якщо модель не передбачає труднощі найближчі два роки, то рейтинг збільшується на 100 балів. В результаті роботи моделей отримуємо три прогнози: фінансові труднощі, чи варто давати кредит, та чи варто давати кредит враховуючи можливі фінансові труднощі.

Результатом оцінювання кредитоспроможні є об'єднані результати кредитного скорингу та прогнози моделей.

2.4 Проектування моделі даних

Діаграма моделі даних зображена на рисунку 2.4. Також в моделі даних повинен бути присутній зв'язок між користувачем і усіма іншими сутностями зв'язком один до одного для ідентифікації користувача, який створив сутність.

Сутності персональної, фінансової інформації, інформації про майно та роботу винесені в окремі від клієнту сутності та мають окремий зв'язок з сутністю кредиту для зберігання повної інформації кредитного запиту та надання можливості при користуванні системою в майбутньому створити власний набір даних для навчання моделей машинного навчання.

Також передбачається заборона видалення інформації про клієнта, якщо вони використовуються в будь-якому кредитному запиті.

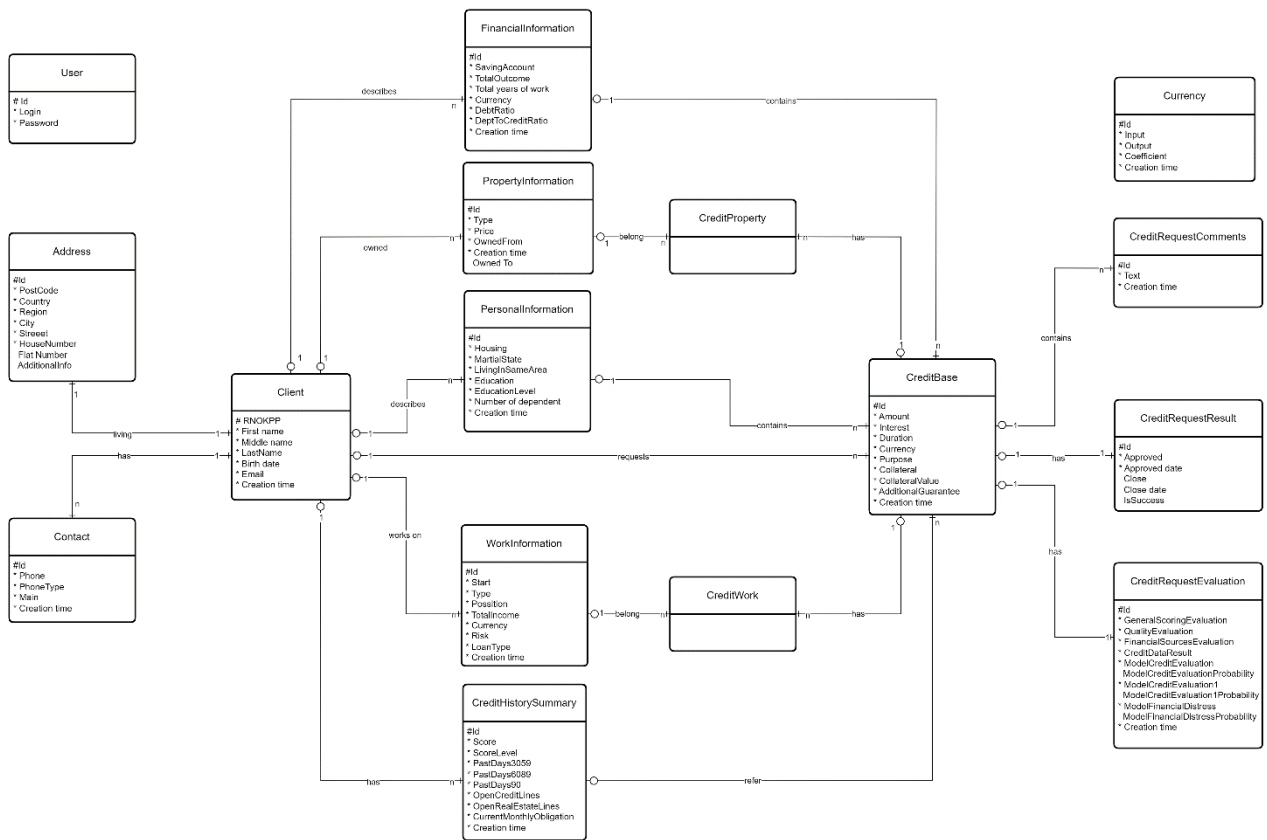


Рисунок 2.4 Схема моделі даних

2.5 Вибір програмних засобів

Для реалізації визначених вимог було обрано наступні технології.

Для реалізації інформаційної системи було вибрано безкоштовну платформу з відкритим кодом .Net 7. Для моделей машинного навчання обрано технологію ML.Net, що дозволяє створювати та тренувати моделі в середовищі .Net. Для створення серверної та клієнтської частин використовується фреймворк Asp, що дозволяє значно прискорити розробку. Для роботи з базою даних використовувався Entity Framework Core. Сховищем даних виступає SqlServer, безпосередньо в роботі використовувалось локальне сховище. Інтерфейс реалізовано за допомогою Razor Pages.

2.6 Висновок за главою 2

В даному розділі було описано набори даних для навчання моделей машинного навчання. Були описані функціональні вимоги системи, спроектована взаємодія користувача з системою, визначений процес оцінювання кредитоспроможності. Також була створена модель даних і було обрано технології для реалізації поставлених функціональних вимог.

РОЗДІЛ 3. ОПИС РОЗРОБКИ СИСТЕМИ

3.1 Архітектура системи

Фреймворк АВР притримується принципів domain-driven design (DDD) і пропонує чотири рівня для реалізації цього підходу. По суті він розділяє рівень презентації на два окремі: презентації (presentation) та прикладний рівень (application). Перший відповідає за інтерфейс для клієнта, а другий стає прошарком між інтерфейсом та рівнем домену. При чому прикладний рівень керує бізнес об'єктами для виконання певних завдань. Рівень інфраструктури в даному випадку надає загальні технічні можливості за допомогою бібліотек сторонніх розробників [12].

На рисунку 3.1 зображені залежності проектів одне від одного.

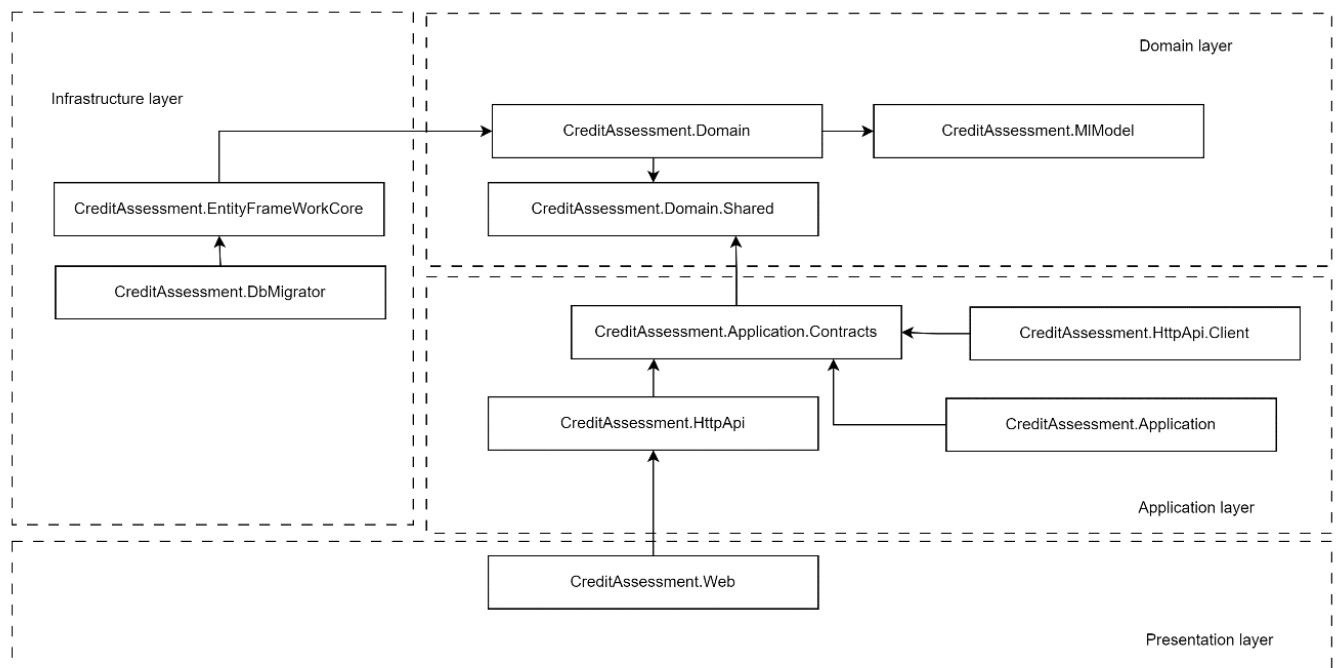


Рисунок 3.1 Діаграма залежностей проектів

Загальна спрощена структура класів серверної частини зображена на рисунку 3.2.

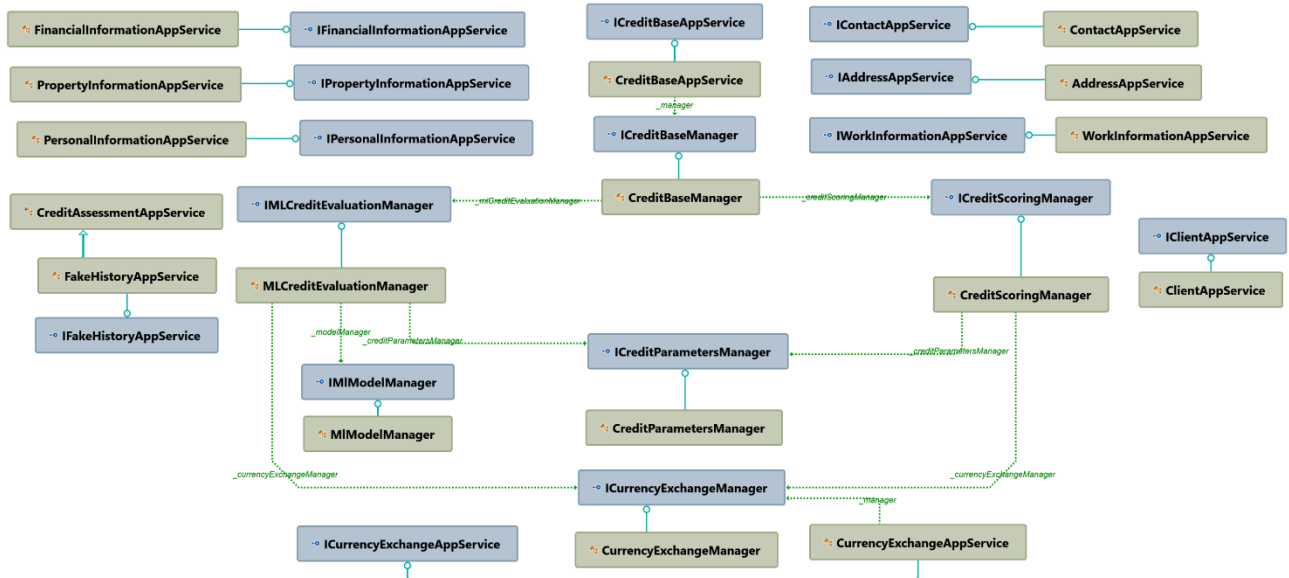


Рисунок 3.2 Спрощена діаграма класів застосунку

3.1.1 Фреймворк АВР

ABP Framework — це повна інфраструктура для створення сучасних веб-додатків із дотриманням найкращих практик і правил розробки програмного забезпечення. Він пропонує архітектуру для створення корпоративних програмних рішень на основі платформ .NET і ASP.NET Core [12].

Фреймворк забезпечує швидкий старт розробки надаючи стартовий шаблон проекту. При використанні стартового шаблону створюється проект, в якому побудовано архітектуру за принципами найкращих практик підходу domain-driven design, вже реалізовано авторизацію та автентифікацію користувачів, створено мінімальний користувацький інтерфейс, доступна локалізація.

При створенні стартового проекту доступні три варіанти архітектури:

- Багатошарова архітектура – рекомендована для застосунків з довгим терміном життя, кодова база яких буде розширятись;
- Одношарова архітектура – для побудови невеликих застосунків;
- Модуль застосунку – архітектура для окремого модулю, який може використовуватись в інших застосунках.

Без додаткових налаштувань фреймворк підтримує три провайдери баз даних: Entity Framework Core, MongoDB та Dapper.

В проєкті одразу налаштовуються AutoMapper для легкої конвертації об'єктів одного типу в об'єкти іншого та Autofac для впровадження залежностей та забезпечення більшої гнучкості коду.

Для інтерфейсу користувача в стартовому шаблоні доступні фреймворки Angular, Blazor, Blazor Pages та Razor Pages.

Для генерації стартового шаблону використовується команда:

```
dotnet tool install -g Volo.Abp.Cli
```

```
abp new [Project name]
```

3.1.2 Doman-driven design

Domain-driven design (DDD) – це концепція, представлена Еріком Евансом у 2004 році в його книзі Domain-Driven Design: Tackling Complexity in Heart of Software. DDD – це підхід який допомагає з вирішенням проблем побудови архітектури програмного забезпечення для складної предметної області. В такому підході фокус при розробці припадає на об'єкти предметної області та їх взаємодію. В архітектурі систем побудованих за даною концепцією виокремлюють три рівня: рівень домену, рівень інфраструктури та рівень презентації.

Рівень домену відповідає за відображення бізнес-логіки. На цьому рівні знаходиться стан, моделі, що відображають бізнес-ситуацію. Хоча технічні деталі

зберігання стану делеговані рівню інфраструктури, тут відбувається контроль всіх необхідних процесів. На цьому рівні реалізована бізнес-логіка застосунку. Важливим також є те, що, в ідеалі, всі сутності визначені на цьому рівні повинні бути вільними від прямих посилань на будь-яку інфраструктуру для доступу до даних.

Рівень презентації (application layer) – визначає завдання які має виконувати програмне забезпечення та спрямовує об'єкти до рівню домену де ці завдання вирішуються. Як правило, цей рівень залишається невеликим. Сутності цього рівня не мають жодної інформації щодо бізнес-логіки.

Рівень інфраструктури відповідає за збереження даних в сховищах. Він не містить в собі жодної бізнес логіки і його задача це надати доступ до об'єктів в сховищі [13].

Мета підходу полягає в тому, щоб логіка рівня домену, її інваріанти, модель даних і відповідні бізнес-правила були повністю незалежні від рівня презентації та рівня інфраструктури. При чому рівні інфраструктури і презентації залежать від рівню домену. Схему залежностей рівнів підходу зображено на рисунку 3.3.

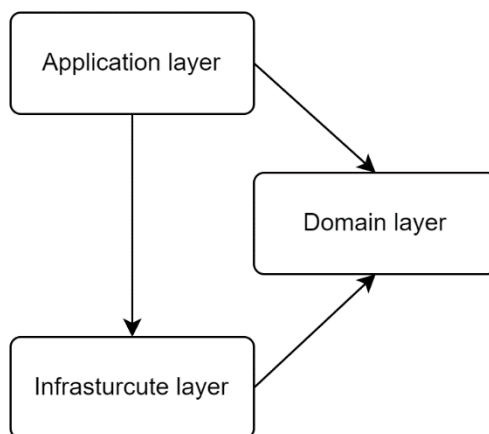


Рисунок 3.3 Схема залежностей рівнів

3.1.3 Рівень інфраструктури

На цьому рівні знаходяться проекти `CreditAssessment.DbMigrator` та `CreditAssessment.EntityFrameworkCore`. `CreditAssessment.DbMigrator` - це консольний застосунок, що спрощує виконання міграції бази даних у різних середовищах. Застосунок при необхідності створює базу даних, застосовує незавершені міграції та за необхідністю додає початкові дані [12]. `CreditAssessment.EntityFrameworkCore` – містить в собі `DbContext` в якому визначена структура бази даних.

3.1.4 Рівень домену

На рівні домену можуть використовуватися наступні об'єкти: репозиторії (`Repository`), сервіси (`DomainService`), прості сутності (`Entity`) та сутності які складаються з декількох об'єктів (`AggregatedRoot`), специфікації (`Specification`). В даній роботі було використано сутності обох типів та сервіси. За замовчуванням цей рівень складається з двох проектів: `CreditAssessment.Domain` та `CreditAssessment.Domain.Shared`. В першому проекті зберігаються всі об'єкти рівня домену, другий проект містить в собі сутності які належать до домену, але можуть бути використані за межами рівня. `CreditAssessment.Domain.Shared` мусить бути незалежним від усіх інших проектів рішення, і всі інші проекти повинні залежати від нього [12]. Також в цьому проекті можуть визначатися сутності для локалізації.

Додатково в даній роботі до доменного рівня належить проект `CreditAssessment.MLModel` в якому було реалізовано моделі для визначення кредитоспроможності.

Спрощена структура (без класів моделей та класів створених фреймворком) класів доменного рівня можна побачити на рисунку 3.4.

В проекті `CreditAssessment.Domain` реалізовано основну логіку системи. Тут знаходяться моделі сутностей предметної області, та менеджери, які реалізують

поставлені задачі. CreditParameterManager обчислює значення параметрів, які використовуються в оцінки кредитоспроможності.

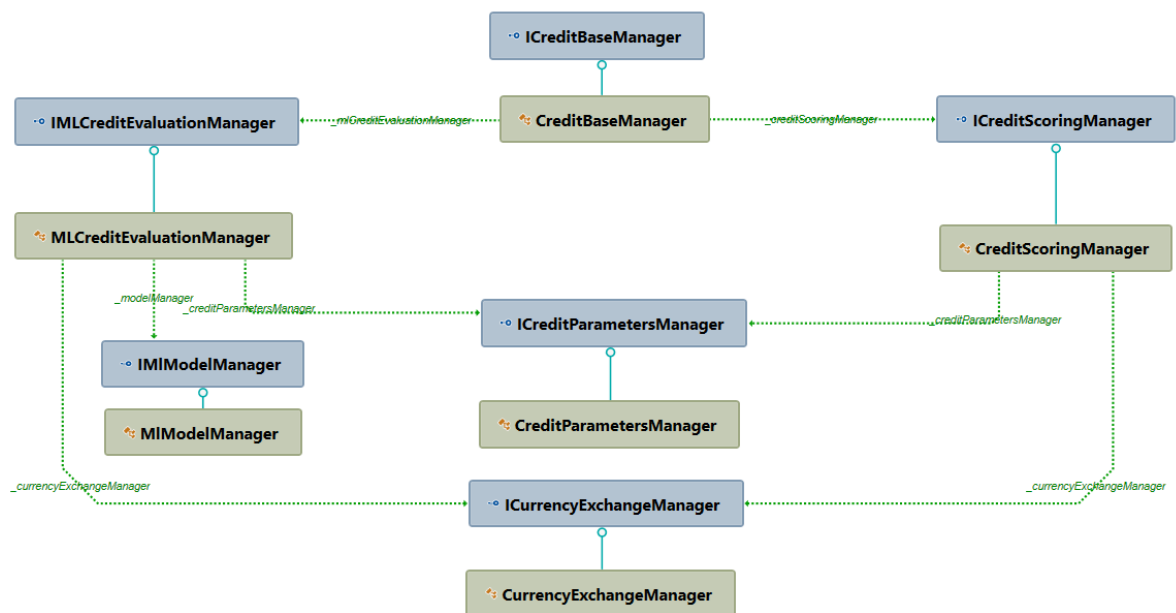


Рисунок 3.4 Діаграма класів рівня домену

CreditScoringManager реалізує методи для виконання кредитного скорингу і є удосконаленою версією попередньої роботи [11]. MLCreditEvaluationManager реалізує підготовку даних та звернення до моделей машинного навчання. CurrencyExchangeManager надає методи для конвертації валюти. CreditBaseManager забезпечує методами для роботи з сутністю кредиту та пов'язаними з ним сутностями, а також надає методи для обчислення кредитоспроможності.

3.1.5 Прикладний рівень

Рівень складається з чотирьох проектів. Проект CreditAssessment.Application.Contracts містить в собі інтерфейси, DTO (Data Transfer Object – шаблон проектування для передачі даних) та дозволи. Він окремо існує для розділення реалізації та інтерфейсів рівня презентації. Це забезпечує можливість

надати клієнтам інтерфейси як контрактний пакет [12]. Проект CreditAssessment.Application містить у собі реалізацію визначених інтерфейсів. Проект CreditAssessment.HttpApi використовується для створення власних контролерів. У більшості випадків фреймворк самостійно визначає контролери спираючись на визначені сервіси в Application. В даній роботі всі контролери було згенеровано автоматично. CreditAssessment.HttpApi.Client створено для визначення клієнтських проксі. В даній роботі він також не використовувався.

Структуру класів прикладного рівня можна побачити на рисунку 3.5.

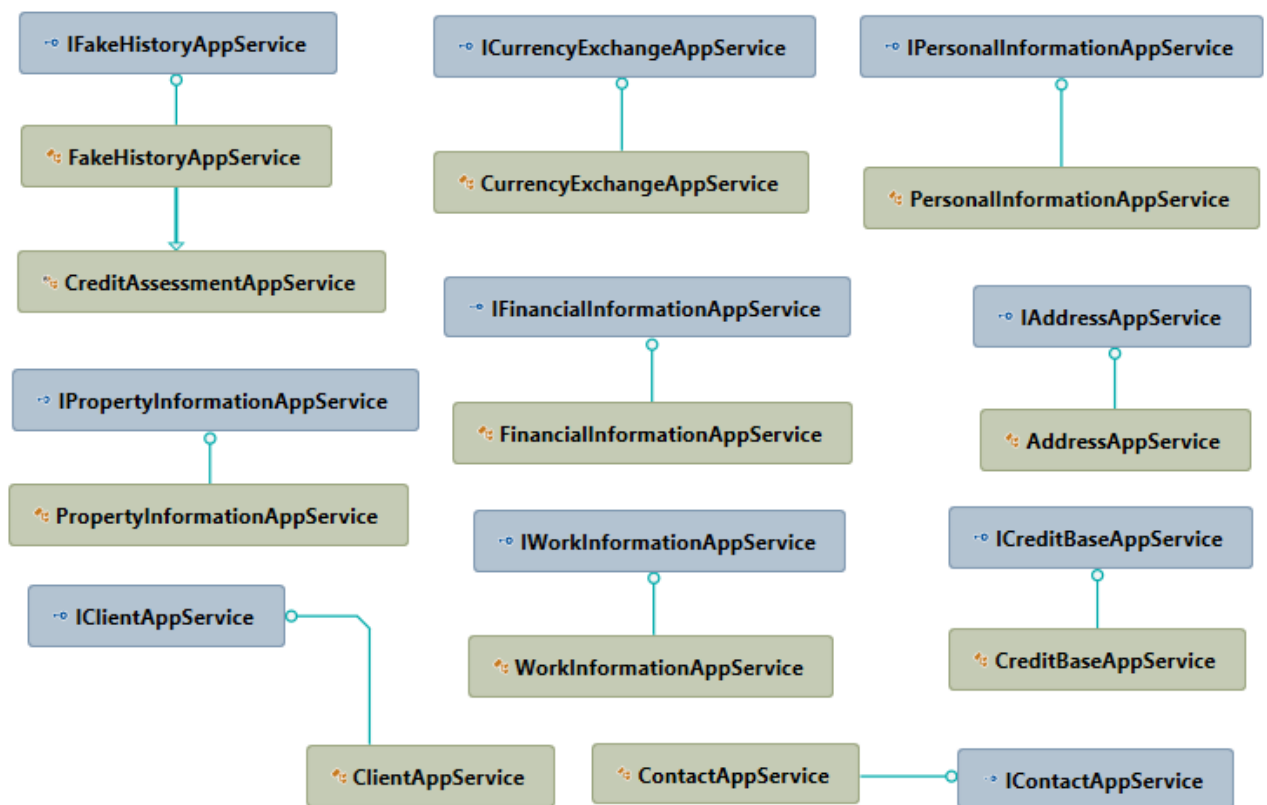


Рисунок 3.5 Діаграма класів прикладного рівня

В роботі було створено 10 сервісів. AddressAppService, ClientAppService, ContactAppService, FinancialInformationAppService, PersonalInformationAppService, PropertyInformationAppService, WorkInformationAppService, CurrencyExchangeAppService реалізують основні функції управління даними (CRUD)

відповідних сутностей та мають додаткові методи, як наприклад отримати перелік сутностей пов'язаних з клієнтом. CreditBaseAppService, окрім реалізації CRUD, забезпечує методи для управління пов'язаними даними такими як коментарі, результати. Всі перелічені сервіси наслідують від абстрактного класу CrudAppService<TEntity, TDto, TKey, TPagedRequestInputDto, CreateDto> в якому і реалізовано основні функції управління даними. Для сутності CreditBase використовується створення, визначене в відповідному менеджері, де забезпечується автоматичне додавання усіх залежностей. FakeHistoryAppService – реалізує методи для генерації кредитної історії.

3.1.6 Рівень презентації

Цей рівень представляє собою один проект CreditAssessment.Web. Він реалізує інтерфейс системи та містить в собі структуру елементів інтерфейсу. В даній роботі в цьому проекті визначались сторінки, модулі та меню сайту. В результаті було визначено три головних пункти меню: Кредити, Клієнти і Валюти. Діаграма класів сторінок проекту для Кредиту зображена на рисунку 3.6.



Рисунок 3.6 Структура класів-сторінок розділу кредити

На сторінках Credit, Credit/Close, Credit/Active та Credit/Waiting знаходяться відповідно таблиці усіх створених кредитних запитів, закритих кредитів, активних та тих, що очікують на підтвердження.

На рисунку 3.7 зображено структуру сторінок розділу клієнта. Розділ клієнта на основній сторінці Client містить таблицю з усіма клієнтами та модали для створення та редагування загальної інформації про клієнта. Сторінка Client/View містить детальну інформацію про конкретного клієнта та можливість її редагування. Сторінка Client/CreditView містить інформацію про кредитну історію клієнта та всі його попередні кредитні запити, а також можливість створити новий кредитний запит.

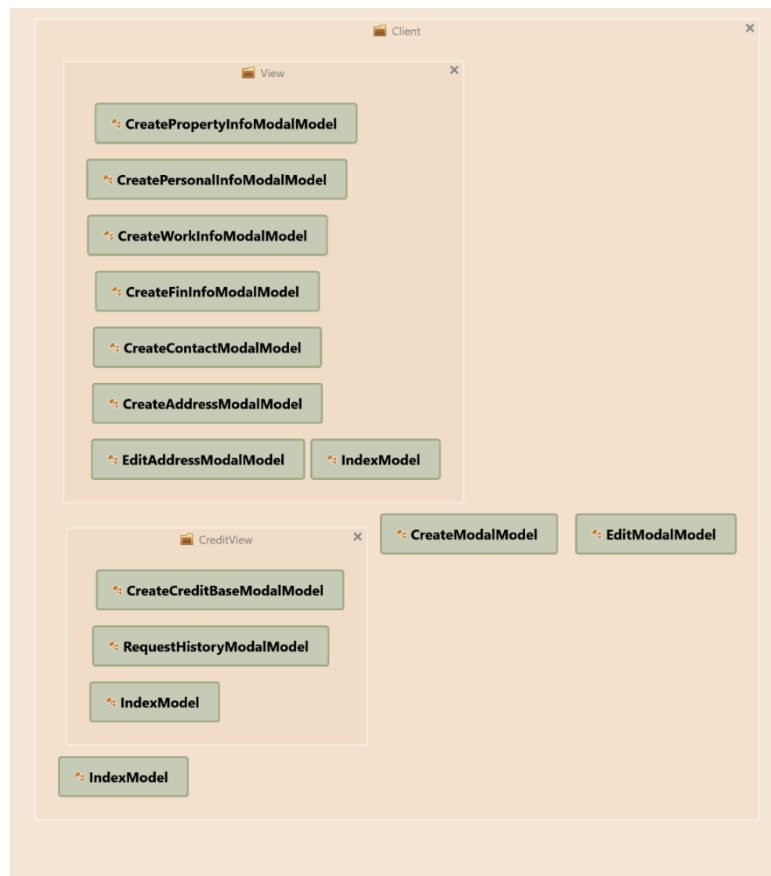


Рисунок 3.7 Структура класів-сторінок розділу Клієнти

На рисунку 3.8 зображена структура розділу інформації про валютні курси. Тут відображається таблиця усіх доданих курсів та є можливість додати новий.

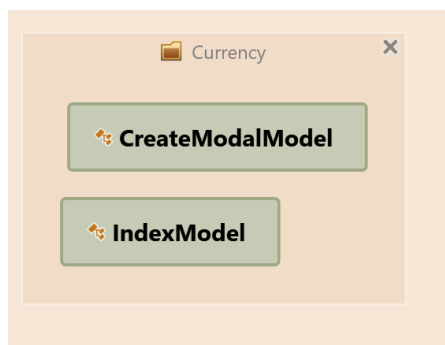


Рисунок 3.8 Структура класів-сторінок розділу Валюта

3.2 ML.NET

ML.NET - це відносно нова платформа, що дає можливість використовувати машинне навчання в застосунках .Net. Вперше платформа була випущена 7 травня 2018 року. Остання стабільна версія з'явилась в лютому 2023 року. Платформа забезпечує виконання автоматичних прогнозів, використовуючи дані, доступні в застосунку [14]. Рекомендованим застосуванням платформи наразі є:

- Класифікація та/або кластеризація
- Регресія
- Визначення аномалій
- Рекомендації
- Класифікація зображень
- Класифікація тексту

Для роботи з моделями фреймворк пропонує набори методів для завантаження даних, їх обробку, розбиття даних, різного роду трансформації та набори класів для проведення тренування.

3.2.1 Model Builder

ML.NET Model Builder — це інтуїтивно зрозуміле графічне розширення Visual Studio для створення, навчання та розгортання спеціальних моделей машинного навчання.

Model Builder використовує автоматичне машинне навчання (AutoML), щоб досліджувати різні алгоритми та налаштування машинного навчання, щоб допомогти знайти той алгоритм, який найкраще відповідає сценарію [14].

Model Builder створює конфігураційний файл `mbconfig`. Після вибору набору даних, сценарію та тренування моделі розширення генерує наступні файли:

- `Model.consumption.cs` — файл, що містить схеми `ModelInput` та `ModelOutput`, функцію `Predict`, що згенерована для використання моделі.
- `Model.training.cs` — файл містить конвеєр (pipeline) для тренування (перетворення даних, алгоритм, гіперпараметри алгоритму), вибраний Model Builder для навчання моделі. Цей конвеєр може бути використаний для повторного навчання моделі.
- `Model.zip`: це серіалізований zip-файл, що представляє навчену модель ML.NET [14].

В Model Builder доступні наступні сценарії:

- Бінарна класифікація
- Мультикласова класифікація
- Класифікація зображень
- Класифікація тексту
- Регресія
- Рекомендації
- Передбачення (forecasting)

3.2.2 Навчання моделей поза Model Builder

Для тренування моделей необхідно виконати наступні кроки:

- Створити новий об'єкт класу `MLContext`
- Зібрати та завантажити набір даних для тестування в об'єкт `IDataView`. Також `ML.NET` дає можливість розбити вхідний датасет на набір для тренування та тестовий набір.
- Визначити конвеєр для визначення функцій для вилучення особливостей (features) та застосування вибраного алгоритму машинного навчання.
- Запустити тренування викликом функції `Fit()` на конвеєрі.
- Оцінити модель.
- На цьому етапі модель можна зберегти в бінарний формат для подальшого використання в застосунках. Збережену модель можна буде завантажити назад в `ITransformer` і викликати функцію `Predict()` для отримання передбачення [14].

Цикл побудови та використання моделей зображено на рисунку 3.9.

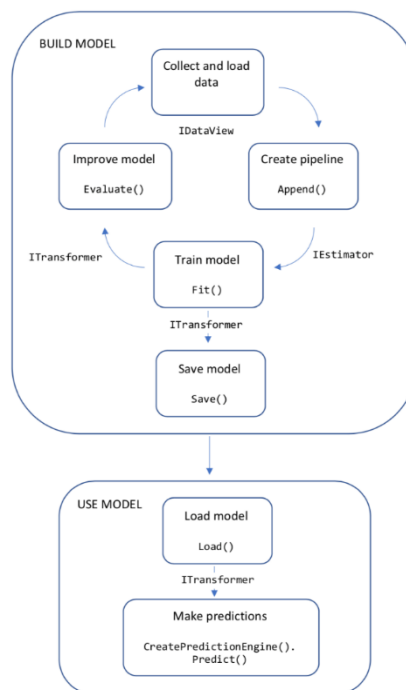


Рисунок 3.9 Цикл побудови та використання моделей [15]

3.2.3 Глибинне навчання на ML.NET

В поточній версії платформи недоступне глибинне навчання. Проте є можливість використовувати моделі, що були попередньо натреновані, наприклад, за допомогою TensorFlow. В даній роботі використовувались лише моделі натренувати які можна з використанням виключно ML.NET.

3.2.4 Опис алгоритмів

3.2.4.1 FastTree

FastTree — це ефективна реалізація алгоритму посилення градієнта MART. Збільшення градієнта — це техніка машинного навчання для задач регресії. Вона будує кожне дерево регресії поетапно, використовуючи попередньо визначену функцію втрат для вимірювання помилки на кожному кроці та виправляє її на наступному. Таким чином, ця модель передбачення насправді є набором слабших моделей прогнозування.

MART вивчає поєднання дерев регресії, яке є деревом рішень зі скалярними значеннями в його листках. Дерево рішень (або регресії) — це бінарна деревоподібна блок-схема, де на кожному внутрішньому вузлі вирішується, який із двох дочірніх вузлів продовжувати, на основі одного зі значень ознак із вхідних даних. У кожному кінцевому вузлі повертається значення. У внутрішніх вузлах рішення базується на перевірці $x \leq v$, де x — значення ознаки у вхідній вибірці, а v — одне з можливих значень цієї ознаки.

Поєднання дерев створюється шляхом обчислення на кожному кроці дерева регресії, яке апроксимує градієнт функції втрат, і додавання його до попереднього дерева з коефіцієнтами, які мінімізують втрати нового дерева. Результатом ансамблю, створеного MART для даного екземпляра, є сума виходів дерева.

У разі проблеми двійкової класифікації вихідні дані перетворюються на ймовірність за допомогою певної форми калібрування [16].

3.2.4.2 FastForest

FastForest – це алгоритм основою якого є дерева рішень. Дерева рішень це не параметричні моделі, які виконують набір простих перевірок на вхідних даних. Прийняття рішення відбувається шляхом співставлення вхідних даних до виходів, що знайдені на тренувальних наборах і які є подібними до вхідних даних. Прийняття рішення відбувається на кожному вузлу бінарного дерева спираючись на міру подібності, яка співставляє кожен елемент рекурсивно через гілки дерева до поки не буде досягнуто потрібний листок і результат не буде повернуто.

Переваги дерев рішень:

- Вони ефективні як для обчислень, так і для використання пам'яті під час навчання та прогнозування.
- Вони можуть представляти нелінійні межі рішень.

Fast forest – це імплементація лісу випадкових дерев (Random Forest). Модель складається з набору дерев рішень. Кожне дерево в лісі видає гаусівський розподіл за допомогою передбачення. Агрегація виконується над набором дерев, щоб знайти розподіл, найближчий до комбінованого розподілу для всіх дерев у моделі.

Як правило, такі моделі забезпечують вищу точність ніж окремі дерева рішень [17].

3.2.4.3 L-BFGS

Матриця Гессе – це симетрична квадратична форма, що описує поведінку функції другого порядку. Для функції f , яка двічі диференціюється в точці $x \in R^n$ матриця Гессе буде дорівнювати:

$$H(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j \quad (3.1)$$

$$\text{де } a_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}.$$

Квазіньютонівський метод – це метод оптимізації, що збирає інформацію про кривизну цільової функції

L-BFGS — це квазіньютонівський метод, який замінює дорогі витрати на обчислення матриці Гессе апроксимацією, але все ще має швидку швидкість збіжності, як метод Ньютона, де обчислюється повна матриця Гессе (3.1). Апроксимація L-BFGS використовує лише обмежену кількість історичних станів для обчислення напрямку наступного кроку, вона особливо підходить для проблем із вектором ознак високої розмірності [18].

Основними гіперпараметрами даного методу є визначення регуляризації. Регуляризація — це метод, який може зробити некоректну задачу більш піддатливою, накладаючи обмеження, які надають інформацію для доповнення даних. Регуляризації L1 та L2 мають різні ефекти та використання, які доповнюють певні аспекти. Під час роботи з даними великої розмірності L1 регуляризація зменшує невеликі ваги нерелевантних функцій до 0, тому під час прогнозування на них не витрачатиметься ресурс. Регуляризація L2 є кращою для даних, які не є розрідженими, і вона значною мірою прибирає ваги з великими значеннями [18].

3.2.4.4 SDCA

Ця модель базується на методі стохастичного двокоординатного сходження (SDCA) – найсучаснішій техніці оптимізації опуклих цільових функцій. Метод SDCA поєднує в собі кілька найкращих властивостей, таких як можливість виконувати потокове навчання (без розміщення всього набору даних у вашій пам'яті) та досягнення прийняттого результату за допомогою кількох сканувань усього набору даних, не витрачаючи обчислень на нулі в розріджених наборах даних. Основними гіперпараметрами даного методу є визначення регуляризації [19].

3.3 Висновок за главою 3

В цьому розділі була описана архітектура системи створена на базі АВР Framework, була описана взаємодія класів та залежності між проектами рішення. Було досліджено можливості технології ML.NET та описано алгоритми моделей, які будуть досліджуватись.

РОЗДІЛ 4. РЕЗУЛЬТАТИ

4.1 Дослідження моделей

Для вибору найкращої моделі та гіперпараметрів для неї було проведено навчання моделей, що реалізують алгоритми FastTree, Lbfgs, FastForest та SDCA. В якості метрики використовувалась метрика AUC, що рекомендується для оцінювання подібних задач [8]. AUC – area under curve – вимірює площу під кривою, створеної шляхом зіставлення істинно позитивного показника з хибнопозитивним.

Результати оцінювання отриманих моделей для прогнозування фінансових труднощів наведено на рисунку 4.1.

FastTree				Lbfgs			
Кількість дерев	Кількість листіків	AUC train	AUC test	L1	L2	AUC train	AUC test
4	63	0,8672616	0,865117	0,2791819F	0,03125F	0,6973014	0,703354
4	163	0,8704745	0,863943	0,2791819F	0,02125F	0,6957757	0,701937
4	80	0,8679302	0,864229	0,2791819F	0,01125F	0,6972638	0,703316
4	33	0,8648763	0,863912	0,2791819F	0,00125F	0,69733	0,70339
6	63	0,8689732	0,863313	0,3791819F	0,03125F	0,6972774	0,70336
6	163	0,875405	0,858439	0,3791819F	0,02125F	0,6972732	0,703356
6	33	0,866051	0,864595	0,3791819F	0,01125F	0,697471	0,703577
6	80	0,8706656	0,863619	0,3791819F	0,00125F	0,6968849	0,702328
8	63	0,8715389	0,862119	0,1791819F	0,03125F	0,6494302	0,652572
8	163	0,8798835	0,851812	0,1791819F	0,02125F	0,6494317	0,652573
8	33	0,8677688	0,863778	0,1791819F	0,01125F	0,649423	0,652571
8	80	0,8732585	0,861822	0,1791819F	0,00125F	0,6494231	0,65257
16	33	0,871678	0,859706	0,0791819F	0,03125F	0,6494203	0,65257
16	163	0,8917315	0,837011	0,0791819F	0,02125F	0,6494255	0,652572
16	80	0,8798316	0,855383	0,0791819F	0,01125F	0,6494205	0,65257
16	63	0,8768959	0,857482	0,0791819F	0,00125F	0,6494272	0,652571
FastForest				SDCA			
Кількість дерев	Кількість листіків	AUC train	AUC test	L1	L2	AUC train	AUC test
4	4	0,8084756	0,812246	0,03125F	0,06536928F	0,642614	0,649697
8	4	0,8123074	0,816635	0,03125F	0,16536928F	0,641752	0,650115
16	4	0,8138678	0,818036	0,03125F	0,00536928F	0,5944308	0,597474
32	4	0,8117571	0,815254	0,03125F	0,26536928F	0,6427362	0,651201
4	12	0,8360737	0,833672	0,13125F	0,06536928F	0,6431645	0,651753
8	12	0,8361157	0,834143	0,13125F	0,16536928F	0,641539	0,65009
16	12	0,8464105	0,842454	0,13125F	0,00536928F	0,6319587	0,634341
32	12	0,8467716	0,842761	0,13125F	0,26536928F	0,6343247	0,643611
4	48	0,8528933	0,847743	0,23125F	0,06536928F	0,6418179	0,650187
8	48	0,8559139	0,850486	0,23125F	0,16536928F	0,6343247	0,643611
16	48	0,8565661	0,851411	0,23125F	0,00536928F	0,5554897	0,562739
32	48	0,8568537	0,851815	0,23125F	0,26536928F	0,6296696	0,637747
4	24	0,8481127	0,843115	0,00125F	0,06536928F	0,6407649	0,646553
8	24	0,848443	0,842989	0,00125F	0,16536928F	0,6424754	0,649039
16	24	0,8510462	0,84621	0,00125F	0,00536928F	0,6105177	0,616804
32	24	0,8518248	0,846912	0,00125F	0,26536928F	0,6425262	0,649616

Рисунок 4.1 Таблиця результатів для датасету "Give Me Some Credit"

Результати оцінювання отриманих моделей для прогнозування чи поверне клієнт кредит наведено на рисунку 4.2.

FastTree				Lbfgs			
Кількість дерев	Кількість листіків	AUC train	AUC test	L1	L2	AUC train	AUC test
100	10	0,995457	0,748184	1F	1F	0,833997	0,751211
200	10	1	0,756053	4.355228F	0.9916154F	0,815643	0,751816
500	10	1	0,75	3.2133F	1F	0,820059	0,754237
800	10	1	0,743341	5F	1F	0,815913	0,74092
1000	10	1	0,745157	6F	1F	0,811909	0,74092
1500	10	1	0,753027	7F	1F	0,799836	0,743947
200	5	0,985839	0,751211	0.2F	1F	0,821367	0,753632
200	20	1	0,772397	0.4F	1F	0,828428	0,735472
200	30	1	0,747579	0.1F	0.2F	0,819646	0,746368
200	40	1	0,743341	3.2133F	0.1F	0,825474	0,746368
1500	5	1	0,734867	3.2133F	0.2F	0,821964	0,758475
1500	20	1	0,746368	3.2133F	0.3F	0,824522	0,746368
1500	30	1	0,736683	3.2133F	0.4F	0,824539	0,745157
1500	40	1	0,74092	3.2133F	0.5F	0,806764	0,751816
100	5	0,948478	0,739709	3.2133F	0.6F	0,826902	0,750605
100	15	0,999788	0,742736	3.2133F	0.7F	0,820134	0,750605
FastForest				SDCA			
Кількість дерев	Кількість листіків	AUC train	AUC test	L1	L2	AUC train	AUC test
803	40	0,94011	0,730024	1F	0.1F	0,560292	0,674637
803	35	0,936542	0,73063	1F	0.2F	0,560292	0,674637
803	30	0,928518	0,73063	1F	0.3F	0,560292	0,674637
803	25	0,919261	0,730024	1F	0.01F	0,598634	0,561743
1500	40	0,940815	0,736077	0.5F	0.1F	0,597039	0,575061
100	40	0,932527	0,726392	0.5F	0.2F	0,560292	0,674637
750	40	0,940161	0,731235	0.5F	0.3F	0,560292	0,674637
500	40	0,939008	0,73063	0.5F	0.01F	0,598519	0,561743
1500	35	0,936622	0,736077	0.2F	0.1F	0,598393	0,567191
100	35	0,931157	0,73063	0.2F	0.2F	0,59416	0,588378
750	35	0,936749	0,726998	0.2F	0.3F	0,599517	0,556901
500	35	0,935624	0,730024	0.2F	0.01F	0,609715	0,679782
1500	55	0,943075	0,73063	2F	0.1F	0,42982	0,59201
150	35	0,931862	0,733051	2F	0.2F	0,42982	0,59201
150	45	0,937684	0,722155	2F	0.3F	0,42982	0,59201
150	55	0,938745	0,731235	2F	0.01F	0,560292	0,674637

Рисунок 4.2 Таблиця результатів для датасету "German credit datat"

Найкращі значення моделей визначались на основі результату тестового датасету. Ці значення на обох рисунках виділено жовтим кольором. Для обох моделей найкращий результат показав алгоритм FastTree. Для моделі оцінювання фінансового стану в найближчі два роки найкращими гіперпараметрами виявились: кількість дерев: 4, кількість листків: 63. Для моделі побудованої на німецькому наборі даних найкращими гіперпараметрами є кількість дерев 200, кількість листків: 20.

4.2 Програмний інтерфейс застосунку

В результаті розробки було створено застосунок, який надає програмний інтерфейс, що складається з контролерів згенерованих фреймворком та запрограмованих контролерів відповідно до функціональних вимог.

Контролер адреси:

- GET /api/app/address/by-client – метод для отримання адреса за ідентифікатором клієнта;
- GET /api/app/address – метод для отримання списку адрес;
- GET /api/app/address/{id} – метод для отримання адреси за її ідентифікатором;
- PUT /api/app/address/{id} – метод для оновлення адреси;
- POST /api/app/address – метод для створення адреси;
- DELETE /api/app/address/{id} – метод для видалення адреси.

Контролер клієнта:

- GET /api/app/client – метод для отримання списку клієнтів;
- GET /api/app/client/{id} – метод для отримання загальної інформації по клієнту за її ідентифікатором;
- PUT /api/app/client/{id} – метод для редагування загальної інформації по клієнту;
- POST /api/app/client – метод для створення загальної інформації по клієнту;
- DELETE /api/app/client/{id} – метод для видалення загальної інформації по клієнту.

Контролер контактної інформації:

- GET /api/app/contact/by-client – метод для отримання списку контактної інформації по клієнту;

- GET /api/app/contact – метод для отримання списку контактів;
- GET /api/app/contact/{id} – метод для отримання контакту за його ідентифікатором;
- PUT /api/app/contact/{id} – метод для оновлення контактної інформації;
- POST /api/app/contact – метод для створення нової контактної інформації;
- POST /api/app/contact/set-as-main – метод для визначення контакту основним для клієнта;
- DELETE /api/app/contact/{id} – метод для видалення контактної інформації клієнта.

Контролер базової сутності кредитного запиту:

- GET /api/app/credit-base/result-by-credit-id/{id} – метод для отримання результату по кредитному запиту за ідентифікатором базової сутності кредитного запиту;
- GET /api/app/credit-base/evaluation-by-credit-id/{id} – метод для отримання результату обрахунку кредитоспроможності за ідентифікатором базової сутності кредитного запиту;
- GET /api/app/credit-base/comments-by-credit-id – метод для отримання списку коментарів за ідентифікатором базової сутності кредитного запиту;
- GET /api/app/credit-base/history-for-credit/{id} – метод для отримання кредитної історії пов'язаної з кредитним запитом;
- GET /api/app/credit-base/history-for-credit-by-client – метод для отримання кредитної історії за клієнтом;
- GET /api/app/credit-base/credits-by-client – метод для списку отримання всіх кредитних запитів клієнта;
- GET /api/app/credit-base/waiting-credits-by-client – метод для отримання списку кредитних запитів клієнта, що очікують на підтвердження;
- GET /api/app/credit-base/active-credits-by-client – метод для отримання списку відкритих кредитів клієнта;

- GET /api/app/credit-base/close-credits-by-client – метод для отримання списку закритих кредитів клієнта;
- GET /api/app/credit-base/waiting-credits – метод для отримання списку кредитних запитів установи, що очікують на підтвердження;
- GET /api/app/credit-base/active-credits – метод для отримання списку відкритих кредитів установи;
- GET /api/app/credit-base/close-credits – метод для отримання списку закритих кредитів установи;
- GET /api/app/credit-base – метод для отримання списку усіх кредитних запитів установи;
- GET /api/app/credit-base/{id} – метод для отримання кредитного запиту за його ідентифікатором;
- PUT /api/app/credit-base/{id} – метод для оновлення інформації про кредитний запит на основі ідентифікатора;
- PUT /api/app/credit-base/result-to-close – метод для закриття кредиту;
- POST /api/app/credit-base – метод для створення базової сутності кредитного запиту;
- POST /api/app/credit-base/credit-result – метод для підтвердження або відхилення відкриття кредиту;
- POST /api/app/credit-base/credit-comment – метод для додання коментарів до кредитного запиту;
- POST /api/app/credit-base/evaluate-credit/{id} – метод для отримання оцінки кредитоспроможності;
- POST /api/app/credit-base/credit-base-with-dependencies – метод для створення базової сутності кредитного запиту з усією інформацією яка доступна від клієнта;
- DELETE /api/app/credit-base/{id} – метод для видалення кредитного запиту.

Контролер валют:

- GET /api/app/currency-exchange/{id} – метод для отримання курсу валют за ідентифікатором;
- PUT /api/app/currency-exchange/{id} – метод для оновлення пари курсу валют;
- POST /api/app/currency-exchange/ - метод для створення нової пари курсу валют;
- POST /api/app/currency-exchange/currency-exchange – метод для додання нової пари курсу валют та автоматичного створення оберненої до неї;
- DELETE /api/app/currency-exchange/{id} – метод для видалення пари курсу валют;

Контролер кредитної історії:

- POST /api/app/fake-history/generate-fake-history/{id} – метод для отримання згенерованої кредитної історії за ідентифікатором клієнта;

Контролер фінансової інформації:

- GET /api/app/financial-information/by-client – метод для отримання списку фінансової інформації за ідентифікатором клієнта;
- GET /api/app/financial-information – метод для отримання списку фінансової інформації;
- GET /api/app/financial-information/{id} – метод для отримання фінансової інформації за її ідентифікатором;
- PUT /api/app/financial-information/{id} – метод для оновлення фінансової інформації;
- POST /api/app/financial-information – метод для створення фінансової інформації;
- DELETE /api/app/financial-information/{id} – метод для видалення фінансової інформації;

Контролер персональної інформації:

- GET /api/app/personal-information/by-client – метод для отримання персональної інформації за ідентифікатором клієнта;
- GET /api/app/personal-information – метод для отримання списку персональної інформації;
- GET /api/app/personal-information/{id} – метод для отримання персональної інформації за її ідентифікатором;
- PUT /api/app/personal-information/{id} – метод для оновлення персональної інформації;
- POST /api/app/personal-information – метод для створення персональної інформації;
- DELETE /api/app/personal-information/{id} – метод для видалення персональної інформації;

Контролер інформації про майно:

- GET /api/app/property-information/by-client – метод для отримання списку інформації про майно, яким володіє клієнт;
- GET /api/app/property-information/by-credit – метод для отримання списку інформації про майно асоційоване з кредитним запитом;
- GET /api/app/property-information – метод для отримання списку інформації про майно;
- GET /api/app/property-information/{id} – метод для отримання інформації про майно за його ідентифікатором;
- PUT /api/app/property-information/{id} – метод для оновлення інформації про майно;
- POST /api/app/property-information – метод для створення інформації про майно;
- POST /api/app/property-information/{id}/set-as-not-active – метод для позначення інформації про майно як неактуальним для клієнта;
- DELETE /api/app/property-information/{id} – метод для видалення інформації про майно;

Контролер інформації про роботу:

- GET /api/app/work-information/by-client – метод для отримання інформації про роботу клієнта;
- GET /api/app/work-information/by-credit – метод для отримання інформації про роботу асоційовану з кредитним запитом;
- GET /api/app/work-information – метод для отримання списку інформації про роботу;
- GET /api/app/work-information/{id} – метод для отримання інформації про роботу за ідентифікатором;
- PUT /api/app/work-information/{id} – метод для оновлення інформації про роботу;
- POST /api/app/work-information – метод для створення інформації про роботу;
- POST /api/app/work-information/{id}/set-as-not-active – метод для позначення інформації про роботу, як неактуальну;
- DELETE /api/app/work-information/{id} – метод для видалення інформації про роботу.

4.3 Користувацький інтерфейс застосунку

Веб-застосунок складається з чотирьох основних розділів: розділ адміністрації (згенерований безпосередньо фреймворком), розділ клієнта, розділ кредитів та розділ курсу обміну валют та сторінки входу. Далі будуть представлені скріншоти кожного з розділів вебсайту.

На рисунку 4.3 зображено сторінку входу в систему.

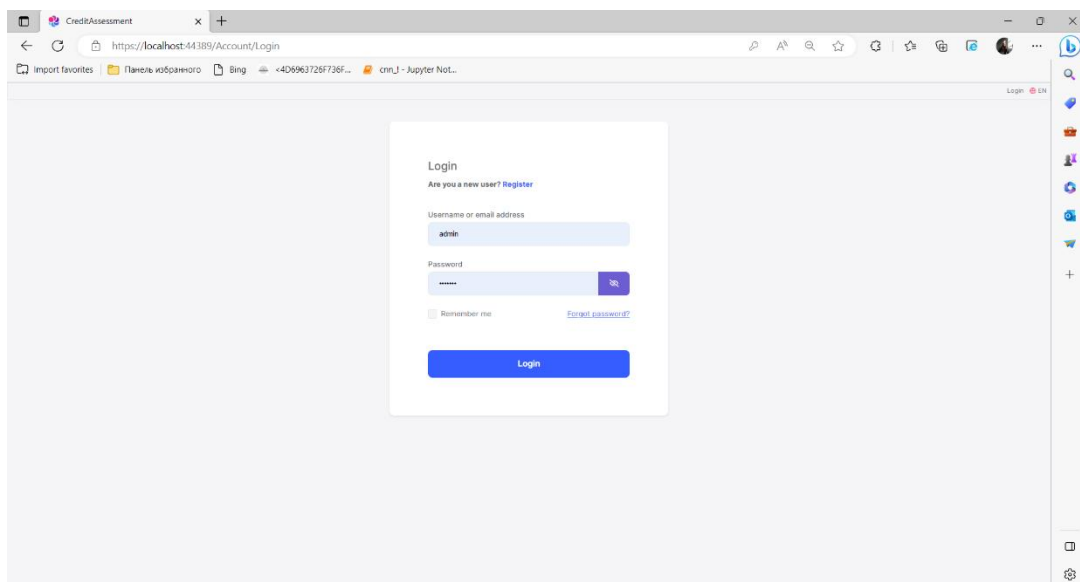


Рисунок 4.3 Форма входу

В меню адміністрації, що зображено на рисунку 4.4 можна переглянути користувачів, відредагувати їх інформацію або дозволи та за необхідністю створити **НОВИХ**.

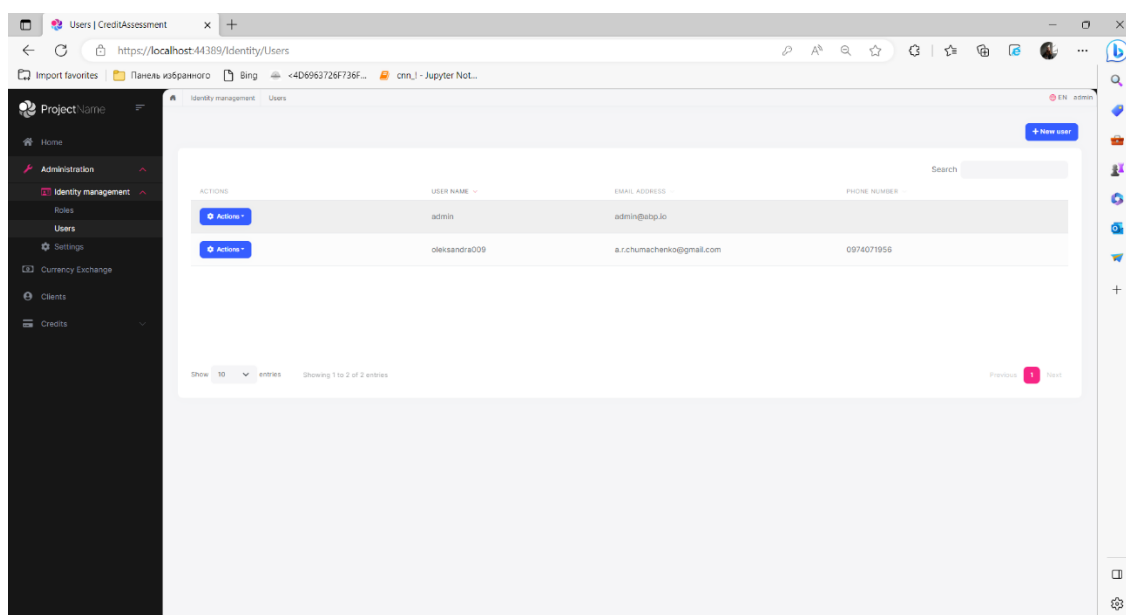


Рисунок 4.4 Сторінка адміністрації. Користувачі

На рисунку 4.5 зображений процес редагування дозволів користувача.

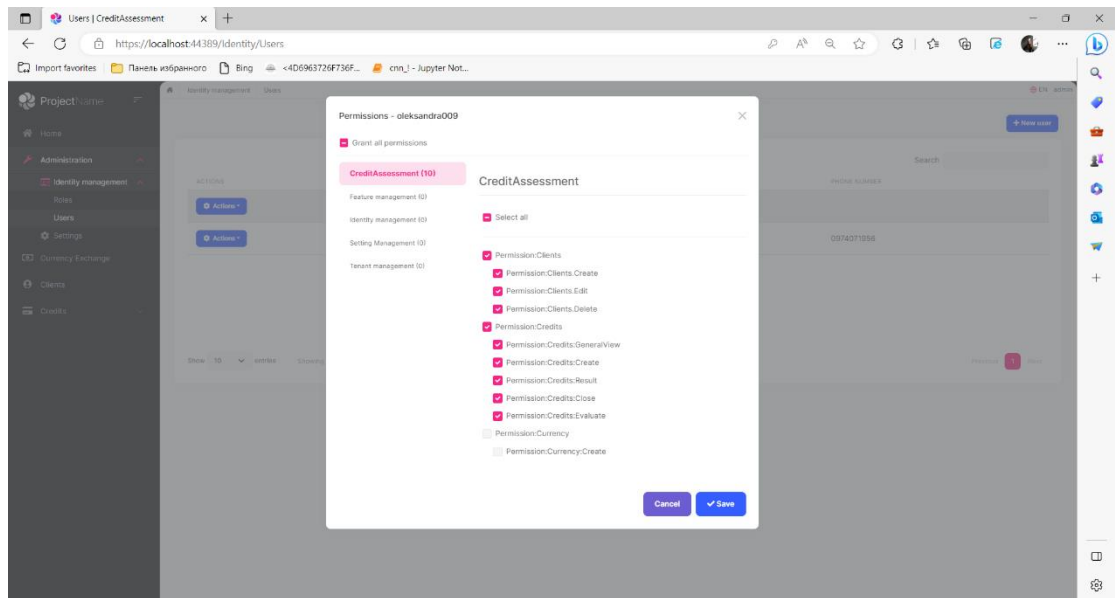


Рисунок 4.5 Редагування дозволів

На рисунку 4.6 зображено сторінку перегляду клієнтів. На цій сторінці можна створити нового клієнта, відредагувати загальну інформацію існуючого, або перейти до детального відображення інформації по клієнту.

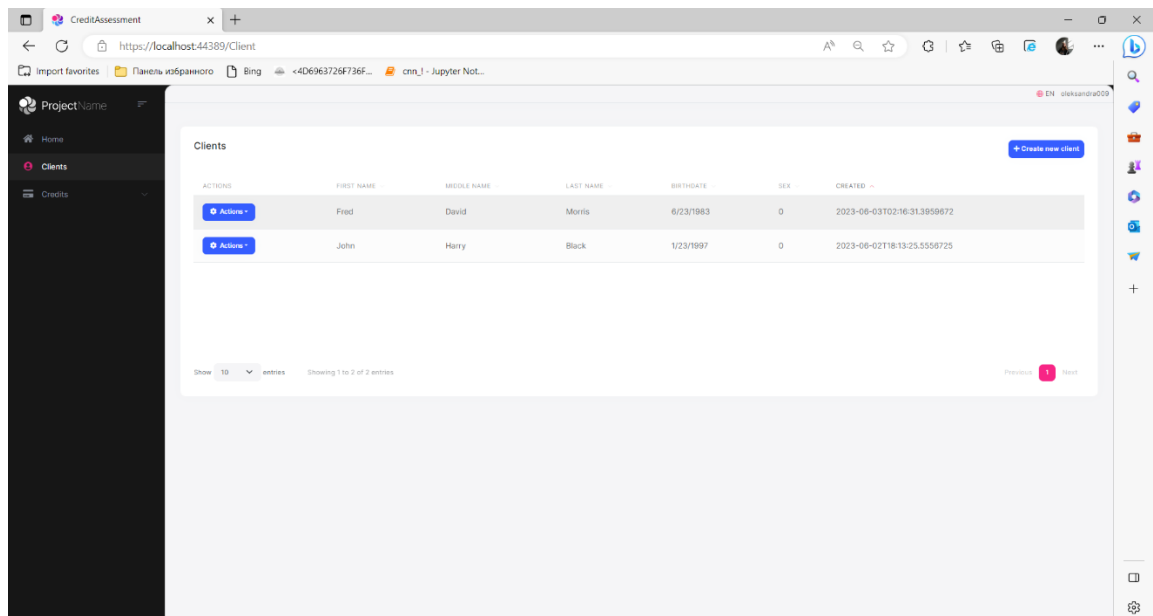


Рисунок 4.6 Список всіх клієнтів

На рисунку 4.7 зображено сторінку детальної інформації про клієнта. На цій сторінці можна відредагувати інформацію про клієнта, додати нову інформацію, або перейти до відображення кредитів та кредитної історії клієнта.

The screenshot shows a web browser window with the URL `https://localhost:44389/Client/View?id=b28e6ca4-a864-2b95-d511-3a0b8ed88425`. The page is titled 'Client account' and features a sidebar with 'Home', 'Clients', and 'Credits'. The main content area is titled 'Client' and includes a button 'Open credit information'. Below this, there are two sections: 'General information' and 'Address information'. The 'General information' section contains fields for 'FirstName' (John), 'MiddleName' (Harry), 'LastName' (Black), 'BirthDate' (1/23/1997), 'Okpo' (11111111), and 'Email' (j.black@gmail.com). The 'Address information' section contains fields for 'Street' (Windsor), 'HouseNumber' (722), 'FlatNumber' (1B), 'Additional information' (2 floor), 'City' (Woonsocket), 'Region' (RI), 'PostalCode' (02895), and 'Country' (United State of America). There are also buttons '+ Edit' and '+ Edit address'.

Рисунок 4.7 Сторінка детального перегляду клієнта

На рисунку 4.8 зображено форму додавання нової інформації щодо майна клієнта.

The screenshot shows a form titled 'Add new property' with a close button. The form includes the following fields: 'Type' (dropdown menu with 'Estate' selected), 'PossessionEstate' (dropdown menu with 'InCapitalOwn' selected), 'Price' (text input with '56000'), 'OwnedFrom' (calendar icon), 'SharedPro' (checkbox), 'Currency' (dropdown menu with 'USD' selected), and a 'Save' button. A calendar widget is open, showing the month of January 2016. The calendar has a grid with days of the week (Su, Mo, Tu, We, Th, Fr, Sa) and dates (1-31). The 'Save' button is highlighted in blue.

Рисунок 4.8 Додавання нової інформації про майно

На рисунку 4.9 зображена сторінка перегляду кредитної історії та кредитних запитів клієнта. Тут можна створити новий кредит, переглянути кожний кредитний запит асоційований з клієнтом та запросити кредитну історію.

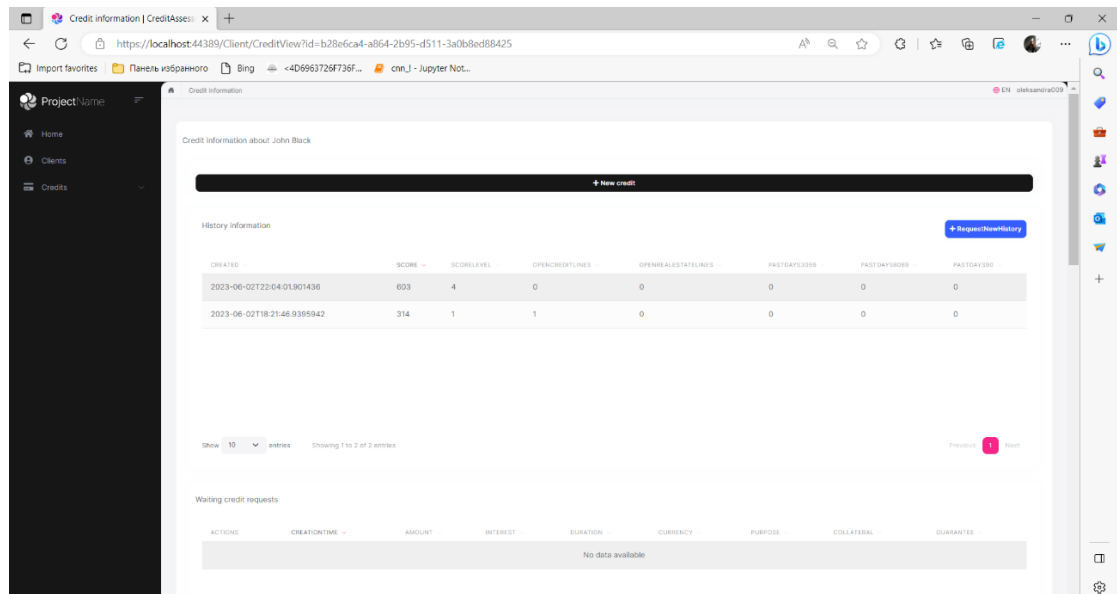


Рисунок 4.9 Сторінка перегляду інформації про кредити клієнта

На рисунку 4.10 зображено процес створення нового кредитного запиту.

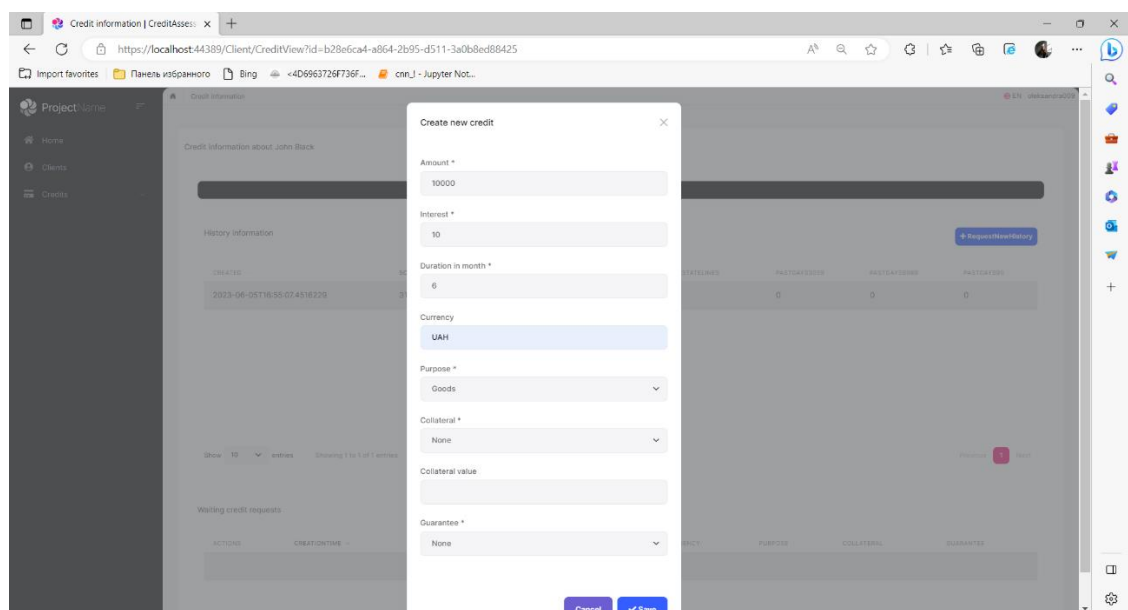


Рисунок 4.10 Створення нового кредитного запиту

На рисунку 4.11 зображена сторінка перегляду списку кредитів, що очікують на підтвердження. Сторінки всіх кредитних запитів, відкритих та закритих кредитів мають аналогічний вигляд. З цієї сторінки є можливість відкрити детальний перегляд кредитів.

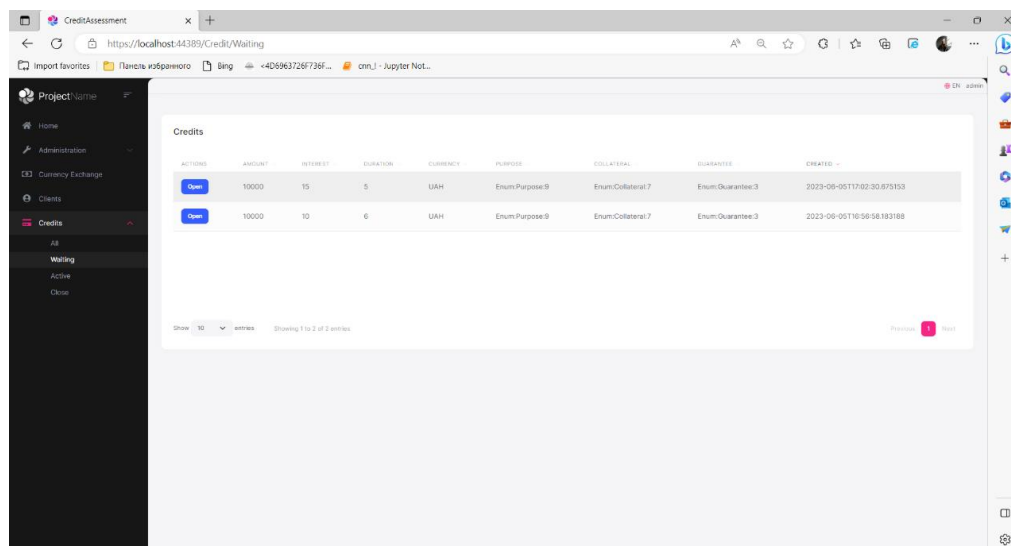


Рисунок 4.11 Перегляд списку кредитів, що очікують на підтвердження

На рисунку 4.12 зображена сторінка перегляду інформації про кредитний запит. Тут можна схвалити кредитний запит, закрити відкритий кредит, обрахувати кредитоспроможність, переглянути та додати коментарі.

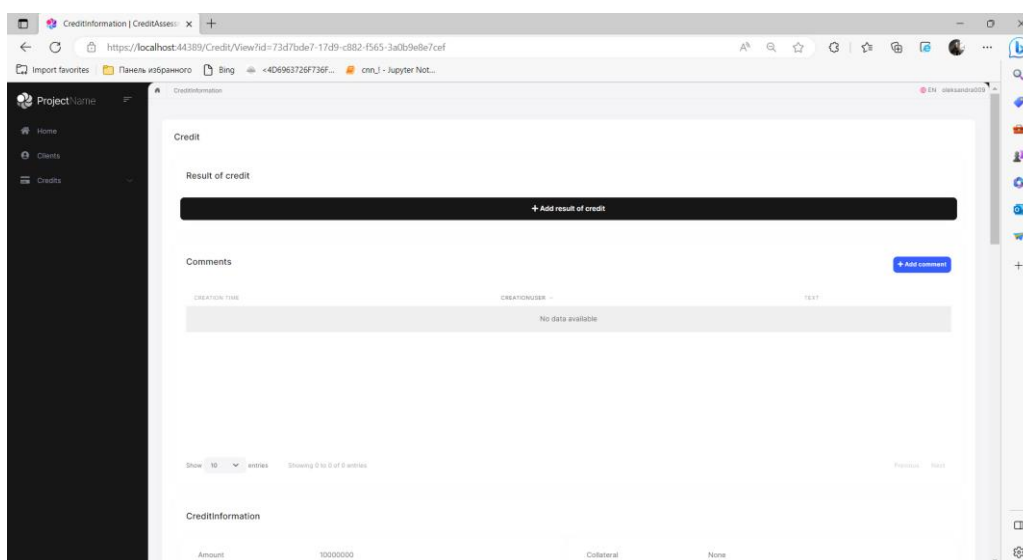


Рисунок 4.12 Перегляд кредитного запиту

На рисунку 4.13 зображена сторінка перегляду додаткової інформації про кредитний запит. Тут можна побачити результат кредитного скорингу та результати моделей з ймовірностями.

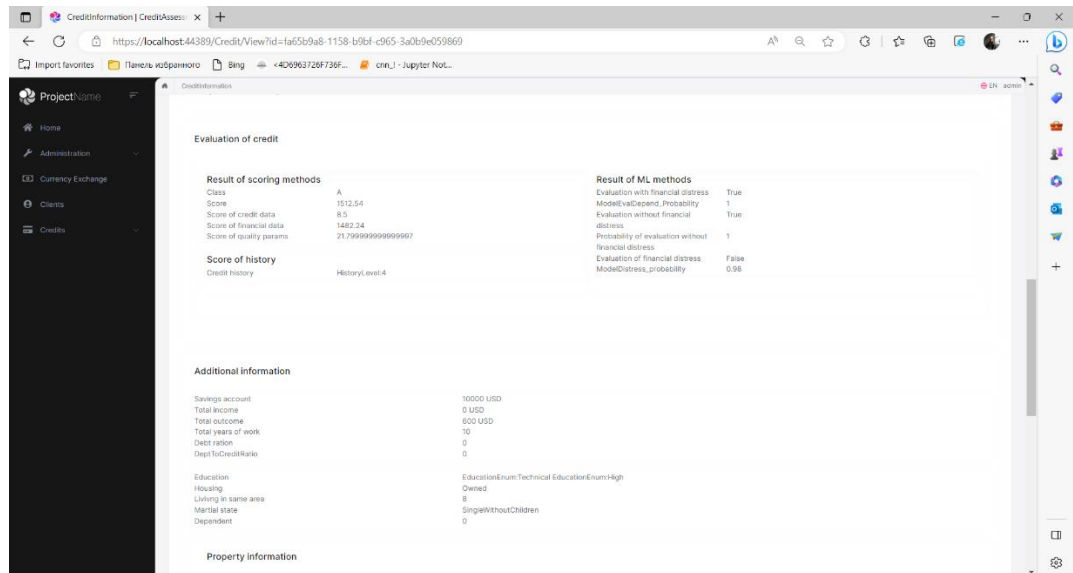


Рисунок 4.13 Інформація про кредитний запит

На рисунку 4.14 можна побачити форму надання схвалення кредиту. Зазначення коментаря при додаванні кредиту є обов'язковим.

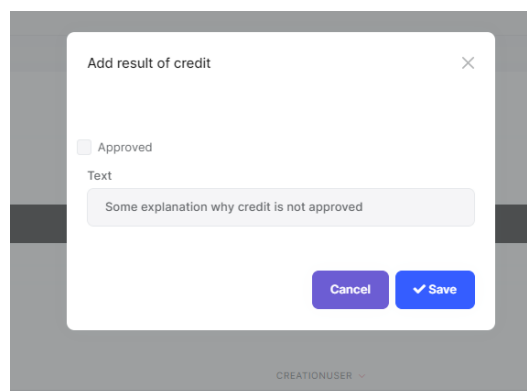


Рисунок 4.14 Додання результату до кредитного запиту

На рисунку 4.15 можна побачити як виглядає сторінка перегляду кредиту коли кредит був підтверджено та успішно закрито.

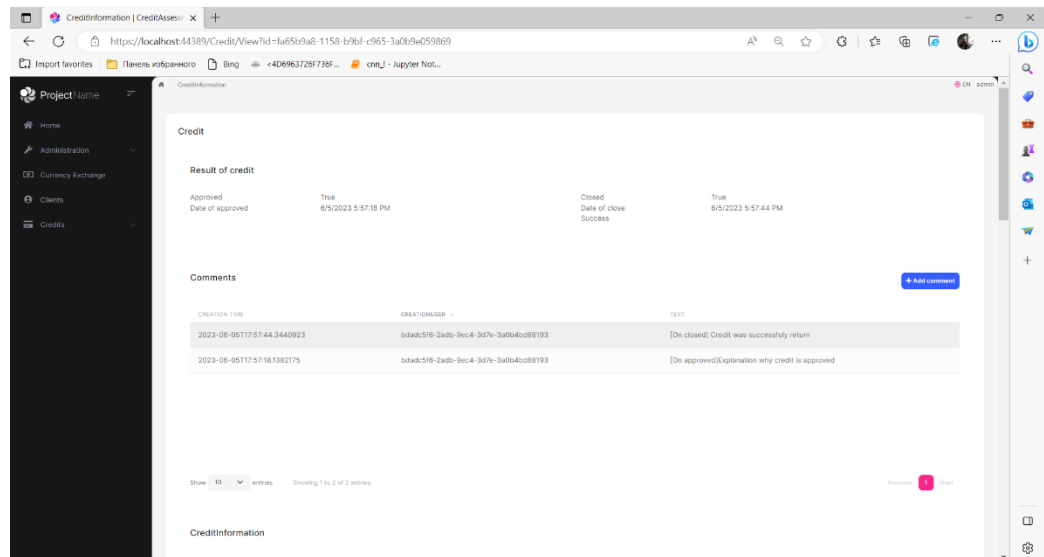


Рисунок 4.15 Сторінка детальної інформації закритого кредиту

На рисунку 4.16 зображено сторінку перегляду курсів обміну валют. Тут можна створити нову пару для конвертації валют

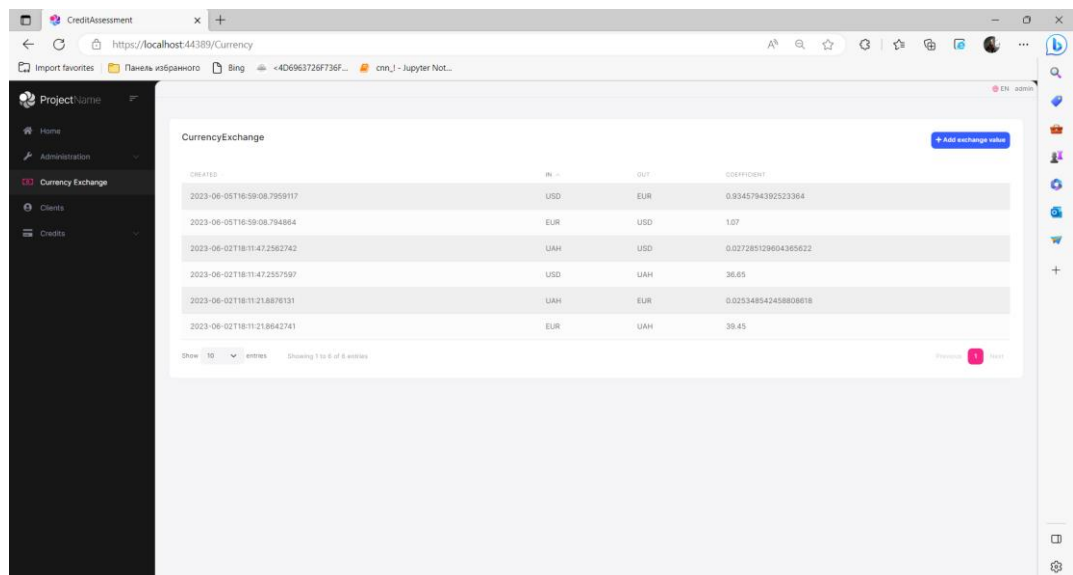


Рисунок 4.16 Сторінка інформації обміну валют

4.4 Висновок за главою 4

В даному розділі приведено результати дослідження чотирьох алгоритмів машинного навчання на двох наборах даних. Для інформаційної системи підтримки прийняття рішень було обрано моделі з найкращими метриками. Було описано програмний та користувацький інтерфейс застосунку.

Висновки

У ході роботи було проаналізовано чотири моделі машинного навчання, створені за допомогою технології ML.NET, для оцінки кредитоспроможності фізичної особи на основі даних з датасету для визначення можливих фінансових труднощів особи в найближчий час та набору даних для прогнозування чи поверне особа взятий кредит. Було підібрано моделі та гіперпараметри, метрики яких були найкращими серед досліджуваних.

Було створено інформаційну систему підтримки прийняття рішень, що надає рекомендації по оцінці кредитоспроможності індивіда. Рекомендація системи складається з результату кредитного скорингу та передбачень трьох моделей: моделі, що прогнозує фінансові труднощі, моделі, що прогнозує чи поверне особа кредит та моделі що послідовно поєднує перші дві. Таким чином, система надає користувачеві більше інформації вибору для вдалого рішення. Інформаційна система побудована враховуючи підходи принципу DDD.

Унікальність отриманого прототипу системи полягає в тому, що в майбутньому моделі, що використовуються для прогнозування можна покращити та перетренувати на реальних, збережених системою даних. Окрім цього архітектура системи дозволить легко розширитись та збільшити функціонал за необхідності, або додати модулі для підключення до інших вже існуючих корпоративних систем. Ще однією перевагою даної системи є реалізований метод кредитного скорингу, що відповідає вимогам НБУ та є в певній мірі універсальним.

Подальше покращення системи можливе в використанні реальних даних для навчання моделей, створення більш складних моделей машинного навчання, наприклад, з використанням попередньо натренованих моделей, додання системи автоматичної обробки документів, що надає клієнт.

Список використаної літератури

1. Карпов, Ігор, and Євген Буров. "ОСОБЛИВОСТІ ВИКОРИСТАННЯ ОНТОЛОГІЙ У СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ: ЗАКОНОМІРНОСТІ ТА ОСОБЛИВОСТІ." Collection of scientific papers «SCIENTIA» May 27, 2022; Stockholm, Sweden (2022): 76-77.
2. Дутченко, Олена Олегівна. Проблеми використання методик оцінки кредитоспроможності позичальника з урахуванням галузевих особливостей. Diss. Київський національний торговельно-економічний університет, 2007.
3. Бюро кредитних історій. Забезпечення конституційних прав і свобод фізичних осіб при організації формування та обігу кредитних історій. Електронний ресурс. Режим доступу: https://minjust.gov.ua/m/str_15099
4. Владичин, У. В. "Банківське кредитування: навч. посіб." К.: Атака (2008)
5. Закон України «Про затвердження Положення про порядок формування та використання банками України резервів для відшкодування можливих втрат за активними банківськими операціями». Електронний ресурс. Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0231-12#n46>
6. Охрименко, Ірина Борисівна, and Ирина Борисовна Охрименко. "Вдосконалення оцінки кредитоспроможності фізичних осіб в банках шляхом застосування диференційованого підходу до позичальників." (2016). Режим доступу: <https://chmnu.edu.ua/wp-content/uploads/2019/06/Ekonomika-i-suspilstvo-5-2016.pdf>
7. Терентьєв, Олександр Миколайович, et al. "Оцінювання кредитоспроможності фізичних осіб за допомогою дерев рішень." (2008).
8. Deep learning for credit scoring: Do or don't? / B. R. Gunnarsson та ін. European Journal of Operational Research. 2021. Т. 295, № 1. С. 292–305. URL: <https://doi.org/10.1016/j.ejor.2021.03.006>
9. West D. Neural network credit scoring models. Computers & Operations Research. 2000. Т. 27, № 11-12. С. 1131–1152. URL: [https://doi.org/10.1016/s0305-0548\(99\)00149-5](https://doi.org/10.1016/s0305-0548(99)00149-5)

10. Give Me Some Credit | Kaggle. Kaggle: Your Machine Learning and Data Science Community. URL: <https://www.kaggle.com/c/GiveMeSomeCredit>
11. Nahirna, Alla, Oleksandra Chumachenko, and Olena Pyechkurova. "Information System Assessment of the Creditworthiness of an Individual." *Наукові записки НаУКМА. Комп'ютерні науки* 5 (2022): 45-48. URL: <https://doi.org/10.18523/2617-3808.2022.5.45-48>
12. Documentation Center | ABP.IO. Documentation Center | ABP.IO. URL: <https://docs.abp.io/en/abp>
13. Evans, Eric. *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional, 2004.
14. ML.NET Documentation - Tutorials, API Reference. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/machine-learning/>
15. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/machine-learning/media/ml-dotnet-annotated-workflow.png>
16. FastTreeBinaryTrainer Class (Microsoft.ML.Trainers.FastTree). Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree.fasttreebinarytrainer?view=ml-dotnet>
17. FastForestBinaryTrainer Class (Microsoft.ML.Trainers.FastTree). Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree.fastforestbinarytrainer?view=ml-dotnet>
18. LbfgsLogisticRegressionBinaryTrainer Class (Microsoft.ML.Trainers). Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.lbfgslogisticregressionbinarytrainer?view=ml-dotnet>

19. `SdcaLogisticRegressionBinaryTrainer` Class (Microsoft.ML.Trainers). Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.sdcalogisticregressionbinarytrainer?view=ml-dotnet>