

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ОПРАЦЮВАННЯ
ДАНИХ ДЛЯ АНАЛІЗУ КРЕДИТНИХ РИЗИКІВ
DEVELOPMENT OF A DATA PROCESSING SOFTWARE SYSTEM
FOR CREDIT RISK ANALYSIS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-212

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

БОРЗІЙ Д.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ПОДКОВАЛІХІНА О.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)
 Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

БОРЗІЙ Данила Вікторовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи опрацювання даних для аналізу кредитних ризиків. Development of a Data Processing Software System for Credit Risk Analysis

керівник проєкту (роботи) к.ф.-м.н., доцент, ПОДКОВАЛІХІНА Олена Олександрівна,

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

затверджені наказом закладу вищої освіти від “07” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 01 червня 2026 року

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

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

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

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

7. Дата видачі завдання « 20 » квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка)

(підпис) Данило БОРЗІЙ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Олена ПОДКОВАЛІХІНА
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 59 с., 5 табл., 23 рис., 2 дод., 36 джерел.

КРЕДИТНИЙ РИЗИК, КРЕДИТНИЙ СКОРИНГ, МАШИННЕ НАВЧАННЯ, ЛОГІСТИЧНА РЕГРЕСІЯ, СТАНДАРТИЗАЦІЯ ДАНИХ, PYTHON, STREAMLIT, ОПРАЦЮВАННЯ ДАНИХ, АВТОМАТИЗАЦІЯ АНДЕРРАЙТИНГУ.

Об'єкт дослідження – процеси застосування алгоритмів та комп'ютерних обчислень для створення програмних систем оцінювання ризиків.

Предмет дослідження – методи та алгоритми машинного навчання (Machine Learning), моделі бінарної класифікації, а також архітектурні рішення та програмні засоби автоматизованого оцінювання ризиків.

Мета роботи – підвищення достовірності класифікаційних рішень при оцінюванні ризиків та мінімізація часових витрат на формування аналітичної звітності шляхом розробки програмного комплексу інтелектуального аналізу даних із використанням методів машинного навчання та автоматизованих модулів генерації звітів.

Матеріали, методи та технічні засоби: об'єктно-орієнтований аналіз та UML-моделювання архітектури ПЗ; методи попередньої обробки даних та конвеєризації (pipeline) складних ознак за допомогою StandardScaler; математичний апарат та алгоритм логістичної регресії (LogisticRegression) для розв'язання задач бінарної класифікації; мова програмування Python; реактивний вебфреймворк Streamlit для проектування графічного інтерфейсу користувача (GUI); бібліотека FPDF для динамічної генерації та рендерингу фінансово-аналітичних звітів на стороні сервера.

Результати: створено кросплатформену програмну систему (вебзастосунок), яка реалізує конвеєр обробки даних, автоматизує обчислення індексу боргового навантаження клієнта (DTI), розраховує ймовірність настання дефолту з мапуванням у 100-бальну шкалу кредитної надійності, динамічно візуалізує вагові коефіцієнти впливу факторів на прийняття рішень моделлю та автоматично формує PDF-звіти для потреб андеррайтингу.

Висновки: розроблено інтелектуальну програмну систему оцінювання кредитних ризиків, впровадження якої дозволило підвищити достовірність класифікаційних рішень завдяки оптимізації гіперпараметрів моделі машинного навчання та мінімізувати часові витрати на формування підсумкової звітності андеррайтерів.

Галузь використання: автоматизація обчислювальних процесів та інтелектуального аналізу даних при оцінюванні платоспроможності в кредитних підрозділах комерційних банків, мікрофінансових організацій, а також у внутрішніх системах фінансово-аналітичних компаній.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 59 pages, 4 tables, 23 figures, 2 appendixes, 36 sources.

CREDIT RISK, CREDIT SCORING, MACHINE LEARNING, LOGISTIC REGRESSION, DATA STANDARDIZATION, PYTHON, STREAMLIT, DATA PROCESSING, UNDERWRITING AUTOMATION.

Object of research – processes of applying algorithms and computing to develop risk assessment software systems.

Subject of research – methods and algorithms of Machine Learning, binary classification models, as well as architectural solutions and software tools for automated risk assessment.

Purpose of the work - increasing the reliability of classification decisions in risk assessment and minimizing time costs for generating analytical reporting by developing a data mining software complex using machine learning methods and automated report generation modules.

Materials, methods, and technical tools: object-oriented analysis and UML modeling of software architecture; methods of data preprocessing and pipelining of complex features using StandardScaler; mathematical apparatus and logistic regression algorithm (LogisticRegression) for solving binary classification problems; Python programming language; Streamlit reactive web framework for designing the graphical user interface (GUI); FPDF library for dynamic generation and server-side rendering of financial and analytical reports.

Results: a cross-platform software system (web application) has been created, which implements a data processing pipeline, automates the calculation of the client's debt-to-income ratio (DTI), calculates the probability of default with mapping into a 100-point creditworthiness scale, dynamically visualizes the weight

coefficients of factor influence on model decision-making, and automatically generates PDF reports for underwriting needs.

Conclusions: an intelligent software system for credit risk assessment has been developed, the implementation of which allowed increasing the reliability of classification decisions through the optimization of machine learning model hyperparameters and minimizing the time spent on generating the final reporting of underwriters.

Scope of application: automation of computational processes and data mining during solvency assessment in credit departments of commercial banks, microfinance organizations, as well as in internal systems of financial and analytical companies.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	9
Вступ	10
1 Аналіз проблеми та постановка завдань дослідження	11
1.1 Сутність кредитних ризиків та огляд літератури	11
1.2 Аналіз аналогів програмного забезпечення	13
1.3 Обґрунтування необхідності розробки та формування вимог до ПЗ	15
1.4 Висновки до розділу	17
2 Матеріали і методи	18
2.1 Вибір та обґрунтування інструментарію розробки	18
2.2 Математична основа конвеєра StandardScaler та моделі LogisticRegression	19
2.3 Висновки до розділу	21
3 Опис програми	22
3.1 Архітектура та функціональна структура системи	22
3.2 Реалізація бізнес-логіки розрахунку індексу DTI	23
3.3 Модуль динамічної генерації PDF-звітів	25
3.4 Інструкція користувача та демонстрація функціоналу системи	26
3.5 Висновки до розділу	32
4 Експлуатація, тестування та експериментальне дослідження програми	33
4.1 Сценарії тестування та валідація роботи системи	33
4.2 Інтерактивне тестування модуля «Аналіз Впливу»	34
4.3 Експлуатація бази даних та інтерактивне керування реєстрами	35
4.4 Висновки до розділу	36
Висновки	37
Перелік джерел посилання	38
Додаток А Текст програми	43
Додаток Б Слайди презентації	52

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
CSV	– Comma-Separated Values;
DTI	– Debt-to-Income;
ML	– Machine Learning;
UI	– User Interface;
PD	– Probability of Default;
БД	– база даних;
ПЗ	– програмне забезпечення;
СКБД	– система керування базами даних.

ВСТУП

У сучасних умовах розвитку фінансового сектору автоматизація процесів оцінки ризиків є ключовим фактором стабільності банківських установ. Традиційні методи оцінки кредитоспроможності часто базуються на жорстких експертних правилах. Це призводить до зростання частки проблемних кредитів. Створення програмної системи опрацювання даних для аналізу кредитних ризиків із використанням методів машинного навчання дозволяє значно підвищити точність прогнозів, мінімізувати вплив людського фактора та прискорити процес прийняття рішень щодо видачі кредитів.

Метою дипломної роботи є дослідження теоретичних та практичних аспектів аналізу кредитних ризиків, обґрунтування вибору методів опрацювання даних, а також проєктування та розробка прототипу програмної системи автоматизованого кредитного скорингу.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити економічну сутність кредитних ризиків та проаналізувати науково-технічну літературу за цією темою;
- провести порівняльний аналіз існуючих комерційних аналогів ПЗ та обґрунтувати доцільність власної розробки;
- сформулювати повний комплекс технічних, функціональних та нефункціональних вимог до системи;
- обґрунтувати вибір технологічного стеку та середовища розробки на базі мови Python;
- програмно реалізувати математичний модуль очищення, масштабування даних та бінарної класифікації позичальників;
- розробити графічний веб-інтерфейс користувача з інтегрованими модулями візуалізації аналізу впливу ознак та динамічної генерації PDF-звітів;
- провести функціональне тестування та експериментальне дослідження розробленого програмного забезпечення.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Сутність кредитних ризиків та огляд літератури

Кредитний ризик — це ймовірність того, що позичальник не поверне гроші за кредитом у встановлений термін. Кінцевою точкою цього ризику для фінансової установи є дефолт клієнта. Сьогодні оцінка позичальників стрімко трансформується: відбувається перехід від застарілих ручних методів та статичних експертних анкет до використання технологій штучного інтелекту, великих даних та машинного навчання.

Сучасні комп'ютерні системи аналізують великі масиви інформації, щоб знайти приховані неколінеарні закономірності та максимально точно визначити рівень фінансової надійності людини. Для побудови теоретичної бази та створення власного програмного забезпечення було досліджено актуальні наукові праці, міжнародні банківські стандарти та технічну документацію.

Вплив структури кредитування на появу заборгованостей та загальний рівень ризику банку розкрито у праці І. Хоми, А. Лавренюк та О. Лук'янського [1]. Ця стаття допомогла виділити ключові чинники для побудови нашої моделі. Особливості використання штучного інтелекту для безпеки операцій та автоматизації клієнтських сервісів досліджено Н. Данік та А. Торлоповим [2], а загальні правила проведення кредитних операцій та відкриття ліній вивчено у роботі Ю. Владики та А. Бегченкової [3]. Сучасні інноваційні підходи до оцінки клієнтів у сфері мікрокредитування описано О. М. Солошенко [4].

Вплив діджиталізації та впровадження нових цифрових ініціатив у трансформацію фінансового аналізу підтверджено у колективній праці О. А. Швиданенка [5]. Вплив воєнних дій, інфляції та кризових факторів на зміну кредитної політики банків України проаналізовано С. М. Ганзюком та Я. М. Клічко [6]. Можливості застосування машинного навчання на банківські послуги описано Д. М. Павлюченком [7].

Роль структури портфеля у загальній ефективності банку досліджено П. Д. Нагорним та В. В. Смалем [8]. Наслідки використання технологій великих даних для стійкості фінансових установ досліджено С. Задворних та Д. Мартяновим [9]. Математичні порівняння точності класичних та сучасних алгоритмів аналізу клієнтів наведено у масштабному дослідженні V. Chang [10]. Динаміку аналітики даних з погляду поведінкових фінансів вивчено за працею В. Almisshal та Н. I. Bulut [11].

Моделі ефективного використання аналітики великих даних за допомогою інструментів штучного інтелекту розкрито А. Ashofteh [12]. Теоретичні перспективи динамічного аналізу ризиків у нестабільних економічних умовах описано Е. Tweneboah Senzu [13]. Створення архітектури вебдодатків, які використовують штучний інтелект для фінансового аналізу, детально розібрано S. D. Mayorga Lira [14]. Цей підхід послужив прототипом для проектування нашого ПЗ. Сучасні гібридні фреймворки оцінки та еволюцію скорингових моделей у комерційних банках досліджено Р. S. Nwachukwu [15].

Правила розробки інтелектуальних скорингових карт вивчено за фундаментальною книгою N. Siddiqi [16]. Економетричний аналіз фінансових ризикових систем досліджено за працею W. H. Greene [17]. Глобальні тенденції розвитку ризик-моделей у світі проаналізовано S. Zamore [18], а порівняння класичних моделей (як-от Altman Z-Score) із сучасними моделями машинного навчання наведено С. Braunsberger та Е. Aschauer [19].

Базові міжнародні регуляторні принципи управління кредитними ризиками запозичено з офіційних стандартів Базельського комітету [20]. Особливості використання суверенних гарантій [21] та альтернативних цифрових даних для оцінки клієнтів, які не мають кредитної історії, вивчено за матеріалами звітів Світового Банку [22]. Правила побудови фінансових калькуляторів ризику досліджено за допомогою офіційного посібника Moody's Analytics [23].

Теоретичні основи управління банківськими ризиками запозичено з посібника А. О. Єпіфанова [24], а шляхи вдосконалення внутрішніх систем контролю розглянуто В. Ерастовим та В. Юхименком [25]. Структуру побудови аналітичних систем фінансового ризик-менеджменту досліджено за допомогою навчального посібника Л. Долінського [26].

Проективання інтерактивного веб-інтерфейсу програми виконано за допомогою документації фреймворку Streamlit [29]. Для автоматичного збереження результатів, розрахунку індексу DTI та створення офіційних звітів у форматі PDF було використано технічний опис класу FPDF [30]. Технічну сторону передоброби та нормалізації числових ознак (віку, доходу) за допомогою конвеєра StandardScaler реалізовано на основі офіційного посібника бібліотеки Scikit-learn [31]. Проекти з відкритим кодом для обробки великих даних проаналізовано за матеріалами Apache Spark [32].

Практичні поради щодо етапів розробки програмного забезпечення для фінансової аналітики запозичено з праці N. Sharma [33]. Як інформаційну базу та джерело відкритих фінансових датасетів для тестування створеної моделі логістичної регресії було обрано репозиторій ListenData [34]. Економічний зміст класифікації ризиків та аналізу дефолтів вивчено за допомогою матеріалів Investopedia [35], а загальні стандарти перевірки комерційних банків — за інструкціями Федеральної резервної системи США [36].

1.2 Аналіз аналогів програмного забезпечення

Для автоматизації процесів оцінки фінансових ризиків у великих банківських установах та транснаціональних корпораціях використовуються потужні комерційні аналітичні комплекси [27], [28]. Найбільш відомими та еталонними представниками на міжнародному ринку фінансового програмного забезпечення є системи SAS Risk Management for Banking та спеціалізована аналітична платформа Credit Benchmark.

Система SAS Risk Management є комплексним інфраструктурним рішенням для управління активами та зобов'язаннями банку. Головна перевага цієї платформи полягає у можливості проводити масштабний статистичний аналіз, будувати прогнознi моделі на основі великих масивів історичних даних та виконувати стрес-тестування кредитних портфелів. Платформа Credit Benchmark фокусується на зборі та аналізі деперсоналізованих кредитних оцінок від провідних світових фінансових інститутів. Вона надає аналітикам точні дані щодо кредитного ризику та ймовірності дефолту великих компаній і галузей ринку в реальному часі.

Для оцінки доцільності створення власного програмного забезпечення було проведено порівняльний аналіз зазначених аналогів та розроблюваної системи. Порівняльну інформацію за ключовими технічними та економічними параметрами зведено у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз аналогів програмного забезпечення

Параметри порівняння	SAS Risk Management	Credit Benchmark	Розроблювана система
Основний клас ПЗ	Промислова аналітична платформа	Глобальний інформаційний сервіс	Інтерактивний локальний застосунок
Точність розрахунку бізнес-правил	Висока	Середня	Висока
Швидкість обробки заявки	Низька	Висока	Максимальна
Вимоги до обчислювальних ресурсів	Максимальні	Середні	Мінімальні
Фінансова доступність	Низька	Низька	Максимальна
Прозорість математичних моделей	Низька	Середня	Максимальна

Проведений порівняльний аналіз показує, що існуючі комерційні аналоги орієнтовані виключно на великі корпорації та вимагають величезних

фінансових вкладень на купівлю ліцензій і розгортання серверної ІТ-інфраструктури. Крім того, вони є закритими системами, що унеможлиблює швидке налаштування бізнес-логіки під потреби конкретного локального кредитного підрозділу.

Це повністю обґрунтовує необхідність розробки власної гнучкої, безкоштовної та кросплатформеної системи на базі мови Python, яка поєднує точність логістичної регресії Scikit-learn із простотою та швидкістю прийняття рішень у веб-інтерфейсі Streamlit.

1.3 Обґрунтування необхідності розробки та формування вимог до ПЗ

Необхідність розробки власної програмної системи аналізу кредитних ризиків обумовлена потребою фінансових установ у легкому, швидкому та інтерпретованому інструменті для первинної оцінки позичальників (андеррайтингу). Існуючі великі системи потребують значних коштів та складного налаштування. Власне програмне забезпечення на базі мови Python дозволяє безкоштовно та ефективно автоматизувати перевірку клієнтів, повністю виключивши людський фактор і знизивши ймовірність неповернення кредиту.

На основі аналізу процесів обробки кредитних заявок було сформовано детальний перелік функціональних та нефункціональних вимог до розроблюваного програмного забезпечення. На рисунку 1.1 представлена розроблена діаграма прецедентів, яка описує функціональну структуру взаємодії користувача з програмною системою аналізу кредитних ризиків.



Рисунок 1.1 – Діаграма прецедентів програмної системи

Нефункціональні вимоги представлені у таблиці 1.2.

Таблиця 1.2 – Нefункціональні вимоги до програмної системи

Категорія	Метрика	Засіб реалізації
Швидкість обробки даних	Час відгуку системи при розрахунку ризику (до 2 секунд)	Масштабування StandardScaler та лінійні обчислення моделі Scikit-learn
Надійність збереження даних	Збереження цілісності реєстру клієнтів після закриття програми	Персистентний запис інформації у локальний файл формату CSV через Pandas
Кросплатформеність та сумісність	Можливість запуску системи на ОС Windows, Linux та macOS	Веб-архітектура інтерфейсу на базі універсального фреймворку Streamlit
Зручність використання	Час на повне введення однієї анкети аналітиком (до 1 хвилини)	Роздільна структура графічного інтерфейсу з бічною панеллю введення
Коректність генерації документів	Стовідсоткова читабельність тексту українською мовою у файлі	Інтеграція Unicide-шрифту з підтримкою кирилиці (DejaVuSans.ttf) у модуль FPDF

1.4 Висновки до розділу

У першому розділі було досліджено сутність кредитних ризиків та проведено детальний огляд профільної наукової літератури. Порівняльний аналіз аналогів (SAS Risk Management, Credit Benchmark) виявив їхні обмеження щодо гнучкості та вартості. Сформульовано функціональні та нефункціональні вимоги до створюваної програмної системи автоматизованого скорингу.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір та обґрунтування інструментарію розробки

Для реалізації інтелектуальної системи аналізу кредитних ризиків критично важливим є вибір оптимального програмно-технологічного стеку. Програмне забезпечення повинно поєднувати високу математичну точність, швидкість обробки даних та гнучкість при розробці користувацького інтерфейсу.

На першому етапі проектування було проведено порівняльний аналіз мов програмування для розробки математичного ядра системи. Як альтернативи було розглянуто популярні бекенд-мови: PHP та JavaScript (Node.js). Порівняльну характеристику мов зведено у таблиці 2.1.

Таблиця 2.1 – Порівняльний аналіз мов програмування для розробки ПЗ

Критерій порівняння	PHP	JavaScript (Node.js)	Python
Основна сфера застосування	Класичні вебсайти, CMS	Високонавантажені real-time застосунки, SPA	Веб-розробка, Data Science, машинне навчання
Підтримка математичних та ML бібліотек	Низька	Середня	Максимальна
Швидкість розробки інтелектуального ПЗ	Низька	Середня	Висока
Зручність створення дата-інтерфейсів	Середня	Висока	Максимальна

Проведений порівняльний аналіз підтвердив, що єдиним оптимальним вибором для реалізації системи є мова програмування Python [10]. Вона має розвинену наукову спільноту та безпрецедентний набір готових інструментів для інтелектуального аналізу даних (Data Mining).

На другому етапі було здійснено вибір вебфреймворку для створення інтерфейсу користувача. Проведено порівняння сучасного асинхронного мікрофреймворку FastAPI та спеціалізованого аналітичного фреймворку Plotly Dash з інструментом Streamlit. Результати порівняння наведено у таблиці 2.2.

Таблиця 2.2 – Порівняльний аналіз вебфреймворків для розробки ПЗ

Критерій порівняння	Plotly Dash	FastAPI	Streamlit
Архітектурний підхід	Фреймворк для аналітичних вебдодатків (на базі Flask та React)	Мікрофреймворк (сучасний, асинхронний)	Спеціалізований Data App фреймворк
Необхідність розробки окремого фронтенду (HTML/JS)	Ні	Так	Ні
Складність реалізації бізнес-логіки та UI	Висока	Низька	Мінімальна
Вбудовані інструменти візуалізації та звітністю	Максимальна	Відсутні	Висока

На основі аналізу даних табл. 2.2 для реалізації графічного інтерфейсу було обрано фреймворк Streamlit. Він дозволяє повністю виключити етап веб-верстки та проектувати реактивні елементи керування базами даних безпосередньо у коді Python, мінімізуючи часові витрати на розробку ПЗ.

2.2 Математична основа конвеєра StandardScaler та моделі LogisticRegression

Математичне ядро розробленої системи опрацювання даних базується на послідовному використанні статистичних методів масштабування та

лінійної бінарної класифікації. Процес оцінки клієнта розділений на два ключові етапи: передобробка вихідних параметрів та розрахунок ймовірності дефолту.

На першому етапі здійснюється масштабування вхідного вектора ознак позичальника за допомогою конвеєра `StandardScaler`. Оскільки вхідні параметри анкети мають суттєво різні діапазони значень та одиниці виміру (вік вимірюється в роках, а річний дохід – у гривнях), пряме навчання класифікатора без попередньої обробки призведе до математичного зсуву ваг у бік більших числових значень. Трансформація та центрування кожної окремої ознаки x здійснюється за формулою (2.1):

$$z = \frac{(x - \mu)}{\sigma}, \quad (2.1)$$

де z – стандартизоване значення ознаки позичальника;

x – початкове значення фінансової або анкетної ознаки з бази даних;

μ – середнє арифметичне значення цієї ознаки у навчальній вибірці;

σ – середньоквадратичне (стандартне) відхилення ознаки.

Після проходження конвеєра масштабування математичне сподівання кожної ознаки стає рівним 0, а дисперсія дорівнює 1.

На другому етапі стандартизований вектор ознак передається в модель логістичної регресії `LogisticRegression` [8], [10]. Алгоритм обчислює математичну ймовірність належності позичальника до дефолтного класу ($y=1$) за допомогою логістичної функції (сигмоїди), що описується рівнянням (2.2):

$$P(y = 1|X) = \frac{1}{1 + e^{-(w^T X + b)}}, \quad (2.2)$$

де $P(y = 1|X)$ – розрахована моделлю ймовірність настання дефолту клієнта;

e – число Ейлера (основа натурального логарифма);

X – стандартизований вектор вхідних ознак позичальника;

w – вектор відкаліброваних вагових коефіцієнтів моделі;

b – вільний член (параметр зсуву регресійної моделі).

У коді програмного забезпечення штучно задано відкалібровані банківські коефіцієнти для уникнення зсуву моделі на малій синтетичній вибірці:

- вектор ваг ознак: `model.coef_ = np.array([-0.2, -1.0, 1.0, -0.7, -2.5])`;
- вільний член (зсув): `model.intercept_ = np.array([0.5])`.

Знак кожного вагового коефіцієнта визначає характер впливу чинника на фінансову стабільність угоди. Негативні ваги для віку (-0.2), річного доходу (-1.0), терміну кредитування (-0.7) та кредитної історії (-2.5) свідчать про те, що зі зростанням цих параметрів ймовірність дефолту $P(y=1|X)$ зменшується. Позитивна вага суми кредиту (+1.0) вказує на те, що збільшення суми позики прямо підвищує кредитний ризик фінансової установи.

Отримана ймовірність дефолту переводиться у лінійний 100-бальний Credit Score (бал надійності) за формулою (2.3):

$$Score = (1 - P(y = 1|X)) \cdot 100, \quad (2.3)$$

де Score – підсумковий інтегральний бал надійності позичальника від 0 до 100 балів.

2.3 Висновки до розділу

У другому розділі було проведено обґрунтування вибору технологічного стеку для створення ПЗ, де основою виступила мова Python та реактивний фреймворк Streamlit. Описано математичний апарат конвеєра стандартизації за допомогою інструменту StandardScaler. Наведено математичні формули сигмоїдальної функції логістичної регресії та розкрито сутність вагових коефіцієнтів моделі для прогнозування ймовірності дефолту позичальників.

3 ОПИС ПРОГРАМИ

3.1 Архітектура та функціональна структура системи

Розроблений програмний комплекс аналізу кредитних ризиків має модульну архітектуру та орієнтований на локальне розгортання. Роль СКБД у системі виконує набір вбудованих інструментів бібліотеки `pandas`, які здійснюють персистентне збереження даних у локальному файлі формату CSV. Схема взаємодії компонентів та загальний потік опрацювання інформації відповідають архітектурному шаблону `Data-Driven Application`.

Функціональну структуру програмного забезпечення розділено на два основні взаємопов'язані модулі:

- аналітичний бекенд (`model_utils.py`) — містить опис логіки бази даних, процедури навчання математичних моделей, алгоритми динамічного аналізу кредитної історії та посторінкового рендерингу документів;
- графічний фронтенд (`app.py`) — реалізує інтерактивний користувацький веб-інтерфейс на базі реактивних віджетів `Streamlit`, здійснює контроль станів через сесії та ініціалізує блоки візуалізації результатів.

Взаємодія між користувачем (ризик-менеджером), графічним інтерфейсом та аналітичним ядром системи при опрацюванні однієї кредитної заявки здійснюється за алгоритмом, який графічно відображено за допомогою UML-діаграми послідовності на рисунку 3.1.

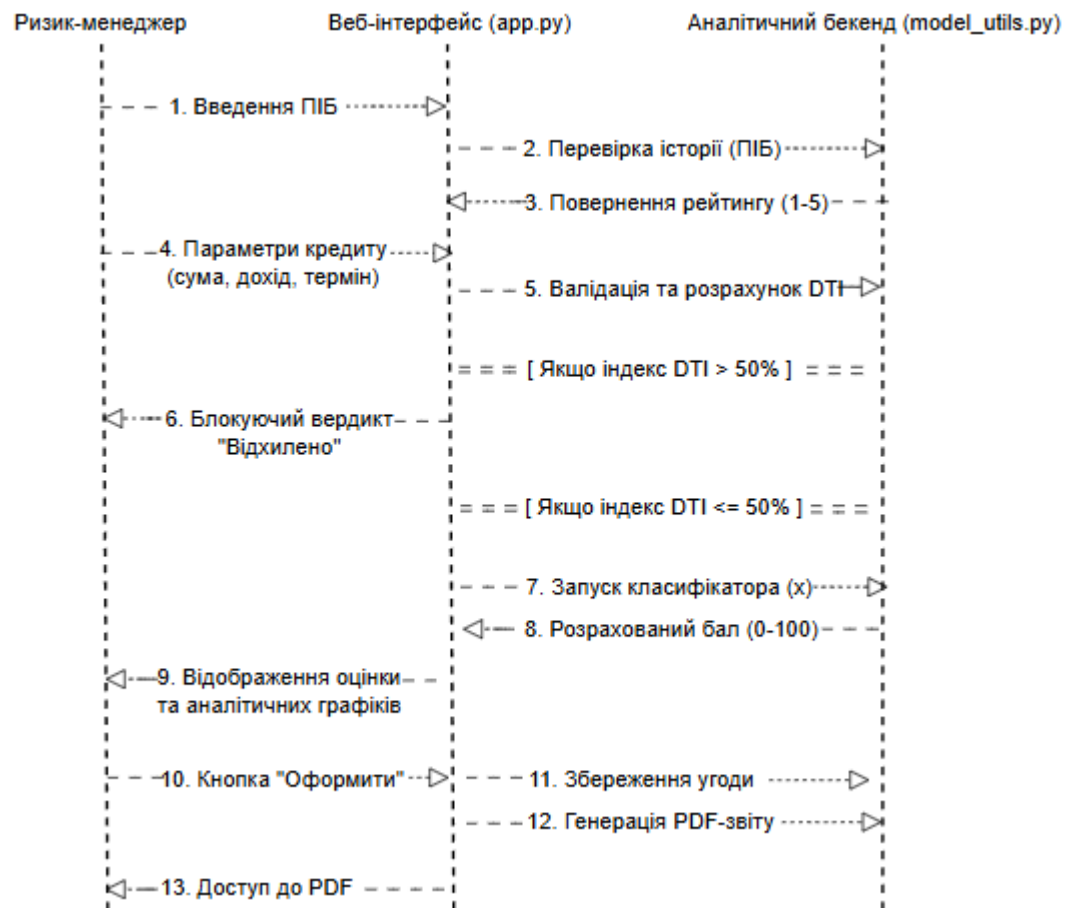


Рисунок 3.1 – Діаграма послідовності (Sequence Diagram) процесу оцінки кредитного ризику

Організація збереження даних у системі базується на файлі `clients_db.csv`, що ініціалізується функцією `init_db()`. При першому запуску програми автоматично створюється таблична структура, яка містить п'ять обов'язкових інформаційних полів: `name` (ідентифікатор клієнта), `amount` (запитуваний обсяг позики), `score` (інтегральний бал надійності позичальника), `date` (штамп дати й часу транзакції) та `status` (поточний фінансовий стан кредитного договору).

3.2 Реалізація бізнес-логіки розрахунку індексу DTI

Програмний комплекс містить вбудований дворівневий фільтр прийняття рішень, який поєднує жорсткі економічні ліміти та статистичні ймовірнісні оцінки машинного навчання.

Рівень1. Автоматичний розрахунок ануїтету та індексу довгового навантаження DTI.

При завантаженні параметрів суми позики та терміну система автоматично обчислює щомісячний платіж за фіксованою базовою ставкою 15% річних ($\text{interest_rate} = 0.15$). Розрахунок здійснюється за класичною фінансовою формулою ануїтету (3.1):

$$\text{payment} = \frac{\text{loan_amount} \cdot \text{monthly_rate}}{1 + (1 + \text{monthly_rate})^{-\text{months}}}, \quad (3.1)$$

де payment — щомісячний ануїтетний платіж за кредитом, грн;

loan_amount — загальна сума кредиту, що запитується позичальником, грн;

monthly_rate — місячна відсоткова ставка, яка дорівнює $0.15/12$;

months — термін кредитування, переведений у місяці ($\text{termin} \cdot 12$).

Після знаходження щомісячного платежу за формулою (3.1) програма розраховує підсумковий відсотковий коефіцієнт довгового навантаження DTI за формулою (3.2):

$$\text{dti} = \frac{\text{payment}}{\frac{\text{income}}{12}} \cdot 100, \quad (3.2)$$

де dti — відсоткове навантаження на місячний бюджет клієнта, %;

income — сукупний річний дохід позичальника, вказаний в анкеті, грн.

Якщо індекс довгового навантаження, отриманий за формулою (3.2), перевищує встановлений банком критичний ліміт у 50% ($\text{dti} > 50$), система миттєво блокує оформлення заявки та виводить червоний віджет

попередження `st.error()`. Це запобігає перевантаженню бюджету клієнта й оптимізує ресурси системи.

Рівень 2. Динамічний перерахунок кредитного рейтингу (Функція `get_client_history`).

Для автоматизації аналізу внутрішньої історії позичальника у файлі `model_utils.py` реалізовано алгоритм динамічного перерахунку рейтингу надійності (від 1 до 5) замість його ручного введення користувачем:

- якщо клієнт є новим для фінансової установи (`name == "New"`), йому автоматично присвоюється середній базовий рейтинг 3;
- якщо в історичному реєстрі `clients_db.csv` за цим ПІБ знайдено хоча б один кредитний договір зі статусом «Дефолт», рейтинг надійності миттєво падає до критичного мінімуму 1, що згодом сильно знизить підсумковий Credit Score в моделі регресії;
- за кожен успішно закритий та повністю виплачений кредит («Виплачено») рейтинг автоматично підвищується за математичним правилом: $\min(3 + \text{closed_loans}, 5)$.

3.3 Модуль динамічної генерації PDF-звітів

Для формування офіційного документа про фінансові результати скорингу та андеррайтингу у файлі `model_utils.py` реалізовано функцію `generate_pdf_report`. Модуль працює за таким алгоритмом:

- ініціалізується порожній об'єкт класу `FPDF` та додається сторінка книжкової орієнтації;
- шляхом підключення зовнішнього TrueType-файлу `DejaVuSans.ttf` забезпечується повна локалізація та підтримка кириличного кодування текстів в юнікодi (`uni=True`), що унеможливорює появу артефактів при рендерингу українських літер [33];

- генерується центровий заголовок документа, лінія розмежування та поточний часовий штамп системи за допомогою методів `pdf.cell()` та `pd.Timestamp.now()`;
- через циклічну конструкцію `for key, value in data.items():` у текстові комірки PDF послідовно виводяться всі анкетні дані позичальника;
- виводяться фінансові розрахунки програми (щомісячний платіж, відсотковий коефіцієнт DTI), фінальний розрахований моделлю логістичної регресії бал надійності (Credit Score) та текстовий вердикт («Схвалено» / «Потрібна додаткова перевірка»);
- результат роботи функції повертається у вигляді байтового потоку через метод `pdf.output(dest='S').encode('latin-1')` для безпечної передачі через веб-інтерфейс і завантаження за допомогою реактивної кнопки `st.download_button`.

3.4 Інструкція користувача та демонстрація функціоналу системи

Експлуатація розробленої програмної системи аналізу кредитних ризиків здійснюється через інтерактивний веб-інтерфейс, розгорнутий за допомогою фреймворку Streamlit. Графічна оболонка застосунку розділена на два функціональні блоки: ліву бічну панель управління для введення первинної інформації та головний екран для відображення аналітичних розрахунків, графіків та реєстру бази даних.

Повний цикл роботи користувача із системою складається з чотирьох послідовних етапів.

Етап 1. Введення анкетних даних та автоматичний аналіз кредитної історії.

На початковому етапі ризик-менеджер заповнює текстове поле «ПІБ клієнта» на бічній панелі. Система ініціалізує функцію `get_client_history()` та проводить сканування локального CSV-файлу бази даних.

Якщо позичальник вже здійснював кредитування, програма завантажує його історію та автоматично виставляє розрахований рейтинг надійності на інтерактивному слайдері від «1» до «5». Для нового клієнта система встановлює базове нейтральне значення рейтингу «3». Далі ризик-менеджер вводить вік, річний дохід позичальника, а також суму кредиту та термін у роках. Вигляд панелі введення даних представлено на рисунку 3.2.

Параметр клієнта

ПІБ клієнта

Іванов В. Н.

Вік

26

Річний дохід

180000 - +

Сума кредиту

70000 - +

Термін кредиту(років)

5

1 30

Кредитна надійність(1-5)

3

Очистити базу даних

Рисунок 3.2 – Графічний інтерфейс бічної панелі введення анкетних даних позичальника

Етап 2. Дворічнева фільтрація та перегляд фінансових результатів.

Після введення фінансових параметрів система в режимі реального часу розраховує щомісячний ануїтетний платіж за ставкою 15% річних та відсотковий індекс довгОВОГО навантаження DTI. На головному екрані динамічно з'являється блок «Результати оцінки», який відображає розраховані метрики.

Якщо індекс DTI є меншим за 50%, на екран виводиться фінальний результат позичальника за 100-бальною шкалою, обчислений моделлю логістичної регресії Scikit-learn, та підсумковий вердикт («Рекомендовано схвалення» або «Потрібна додаткова перевірка»). Візуальне відображення результатів успішного проходження первинного андеррайтингу наведено на рисунку 3.3.

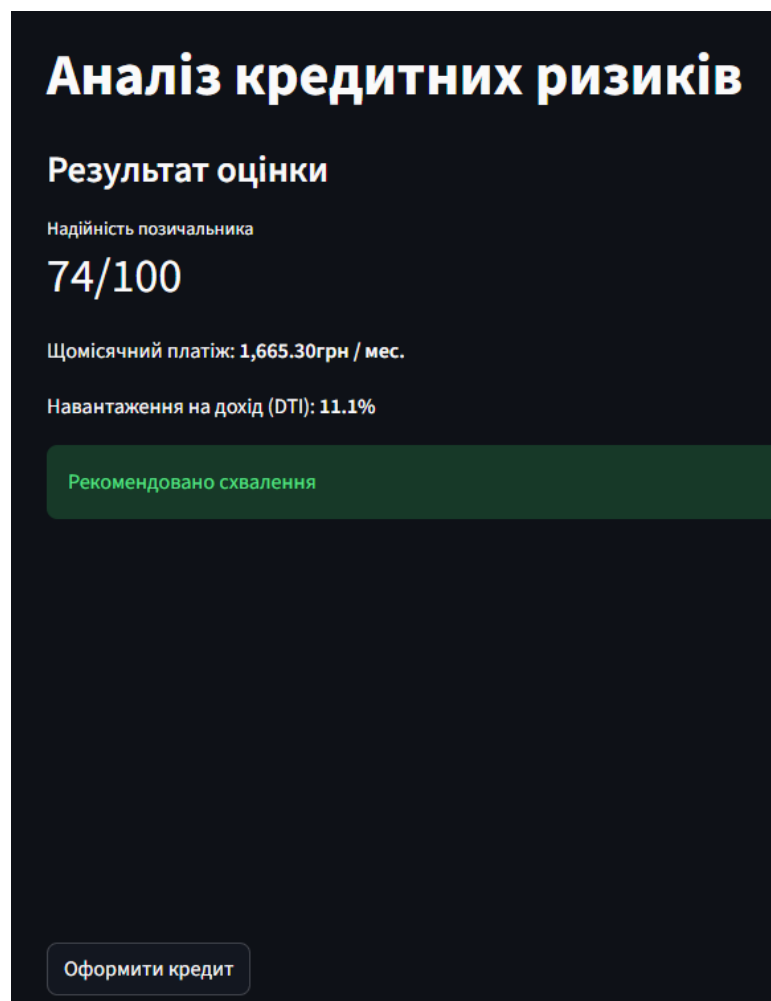


Рисунок 3.3 – Відображення результатів скорингу та фінального вердикту системи

Якщо параметри кредиту створюють надмірне навантаження на бюджет клієнта ($dti > 50$), система виводить повідомлення про критичний рівень індексу DTI та блокує можливість оформлення клієнта, як показано на рисунку 3.4.

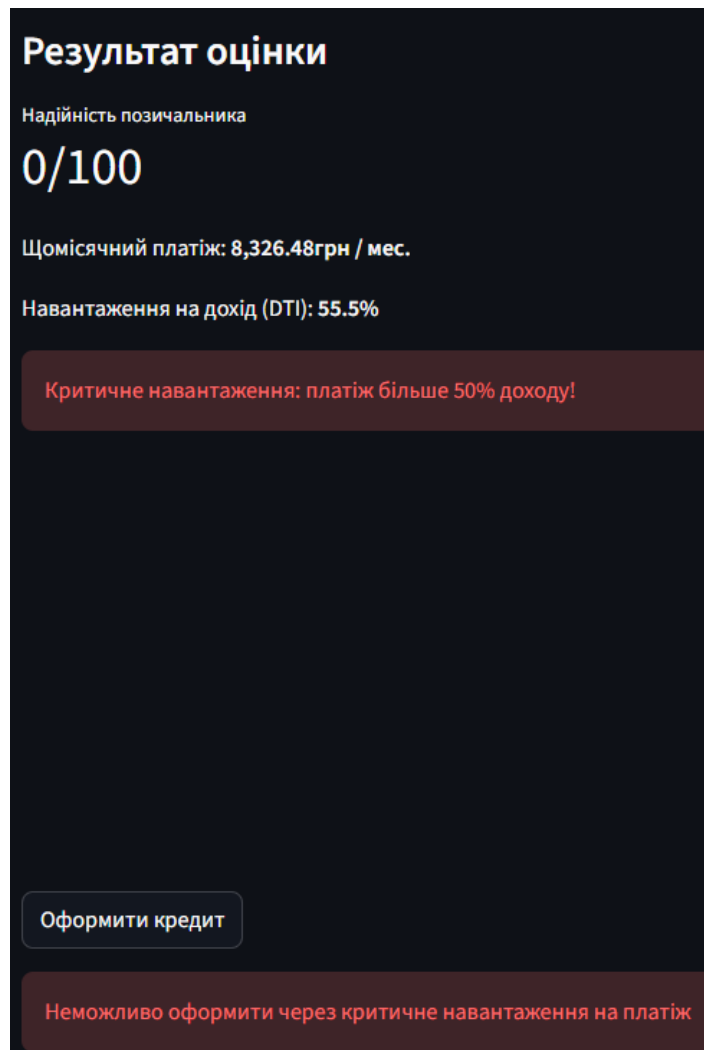


Рисунок 3.4 – Блокування заявки системою при перевищенні ліміту довгового навантаження

Етап 3. Інтерпретація рішень штучного інтелекту за допомогою «Аналізу впливу».

Для детального вивчення причин учвалення моделлю конкретного рішення аналітик використовує інтерактивний блок графічного аналізу.

Система будує горизонтальну стовпчикову діаграму Seaborn, де відображено математичний внесок кожної окремої ознаки позичальника у підсумкову оцінку ризику.

Чинники, які підвищують надійність клієнта, форбуються у зелений колір, а чинники, що створюють ризик – у червоний. Графічне вікно модуля інтерпретації представлено на рисунку 3.5.

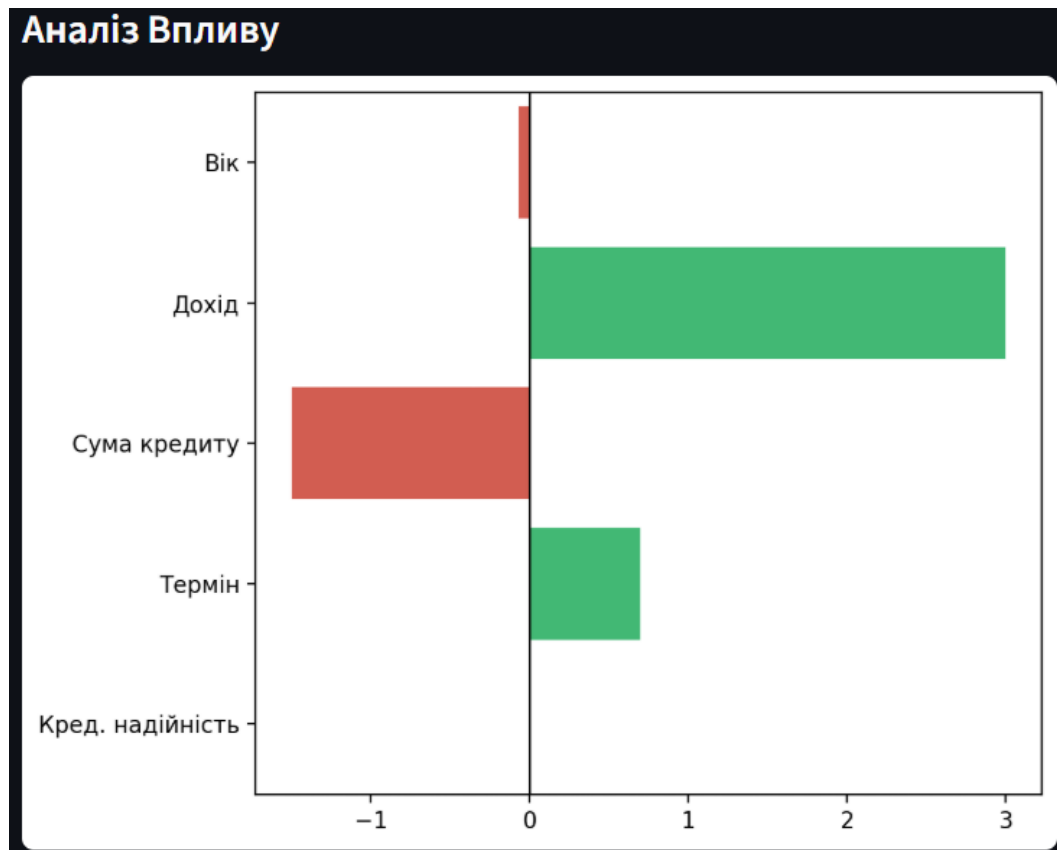


Рисунок 3.5 – Інтерфейс модуля графічної інтерпретації факторів ризику
«Аналіз впливу»

Етап 4. Фіксація угоди, генерація PDF-звіту та моніторинг реєстру.

Після ухвалення рішення користувач натискає кнопку «Оформити кредит». Програма виконує дві синхронні дії: записує параметри угоди зі статусом «Виплачується» до локальної бази даних та активує кнопку «Завантажити PDF». При її натисканні клієнту віддається сформований PDF-документ із усіма фінансовими подробицями.

Вся історія банку відображається в нижній частині програми у вигляді інтерактивної таблиці. Кредитний аналітик може вручну змінювати статуси попередніх угод у колонці «status» на «Виплачено» або «Дефолт», після чого натискає кнопку «Оновити статус у базі» для персистентного збереження змін у CSV-файлі. Реєстр кредитних договорів наведено на рисунку 3.6, а також сформований PDF-звіт на рисунку 3.7.

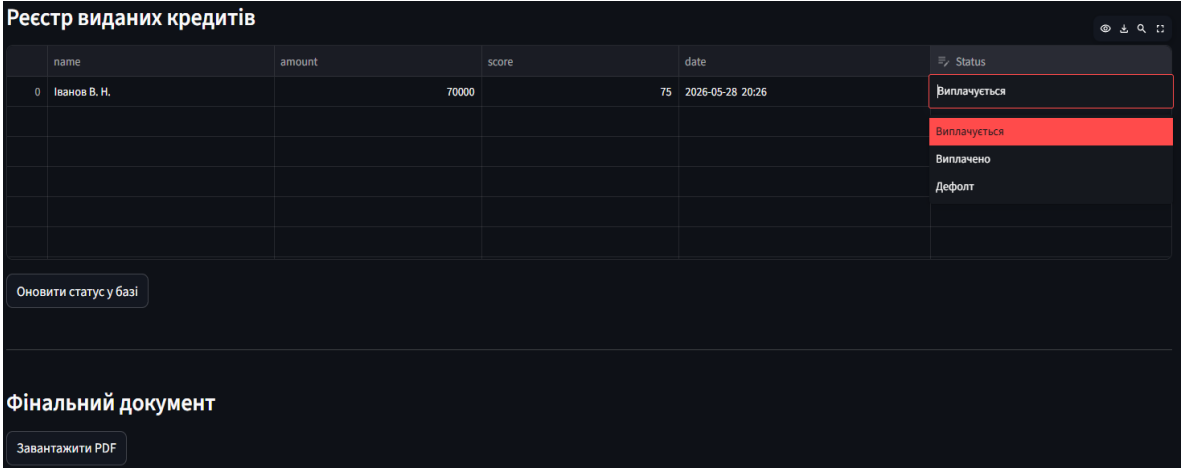


Рисунок 3.6 – Інтерактивний реєстр кредитних договорів та інтерфейс генерації звітів

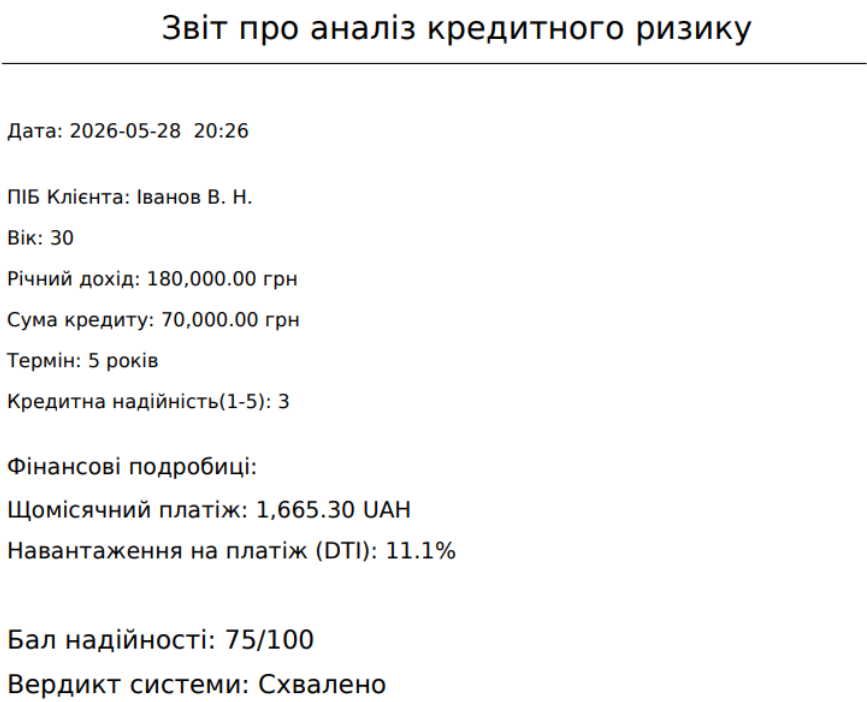


Рисунок 3.7 – Сформований PDF-звіт

3.5 Висновки до розділу

У третьому розділі було детально описано архітектуру розробленого програмного забезпечення, що складається з модулів `model_utils.py` та `app.py`. За допомогою UML-діаграми послідовності проаналізовано процес опрацювання даних при оцінці кредитних ризиків. Наведено математичні формули розрахунку щомісячного ануїтетного платежу та індексу довгового навантаження DTI, а також логіку роботи дворівневого фільтра прийняття рішень. Описано структуру модуля генерації фінансових PDF-звітів із повною підтримкою кирилиці та показано повний цикл роботи користувача з системою.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Сценарії тестування та валідація роботи системи

Експлуатація розробленої системи аналізу кредитних ризиків передбачає взаємодію кредитного інспектора з графічним веб-інтерфейсом додатка. Для підтвердження працездатності, стійкості до помилок користувача та коректності виконання вбудованих бізнес-правил було проведено функціональне тестування. Результати перевірки системи за основними сценаріями зведено у таблиці 4.1.

Таблиця 4.1 – Результати функціонального тестування компонентів ПЗ

Початкові умови (Вхідні дані)	Дія користувача (Сценарій)	Очікувана реакція системи	Статус
Порожнє текстове поле «ПІБ клієнта»	Натискання кнопки «Оформити кредит»	Блокування транзакції, виведення повідомлення про помилку «ПІБ клієнта не введено!»	Виконано
Введення нового клієнта	Натискання клавіші Enter у полі імені	Виклик get_client_history(), автоматичне присвоєння базового рейтингу 3	Виконано
Позичальник з діючим статусом «Дефолт» у реєстрі	Введення імені існуючого дефолтного клієнта	Миттєве зниження автоматичного слайдера кредитної надійності до рівня 1	Виконано
Клієнт має успішно закриті договори («Виплачено»)	Введення імені надійного клієнта	Динамічний перерахунок та підвищення рейтингу за формулою до значення 4 або 5	Виконано

Кінець таблиці 4.1

Початкові умови (Вхідні дані)	Дія користувача (Сценарій)	Очікувана реакція системи	Статус
Річний дохід: 120 000 грн, Сума позики: 500 000 грн	Завантаження фінансових параметрів кредиту	Автоматичний розрахунок навантаження, $dti > 50 \%$, виведення блокуючого вердикту «Критичне навантаження»	Виконано

4.2 Інтерактивне тестування модуля «Аналіз Впливу»

Для перевірки інтерпретованості прийнятих рішень було проведено експериментальне дослідження графічного модуля «Аналіз Впливу» [5, 34]. Обчислення внеску кожної окремої ознаки позичальника у підсумковий кредитний бал у кодї програми реалізовано за математичним правилом інверсії (4.1):

$$\text{current_impact} = -(\mathbf{w}^T \cdot \mathbf{X}_{\text{scaled}}), \quad (4.1)$$

де current_impact – вектор внеску кожної ознаки позичальника;

$\mathbf{w}^T$ – транспонована матриця вагових коефіцієнтів навченої логістичної регресії(`model.coef_`);

$\mathbf{X}_{\text{scaled}}$ – масштабований за допомогою `StandardScaler` вектор ознак поточного клієнта(`input_scaled`).

Завдяки операції інверсії (множення на мінус одиницю), додатні показники впливу (наприклад, високий дохід чи високий рейтинг надійності) збільшують підсумковий бал і підсвічуються на діаграмі Seaborn зеленим кольором. Негативні чинники (занадто велика сума кредиту або низький вік) зменшують оцінку та фарбуються у червоний колір. Це забезпечує повну

візуальну прозорість роботи штучного інтелекту для банківського аналітика. Графічний модуль «Аналіз Впливу» зображено на рисунку 4.1:

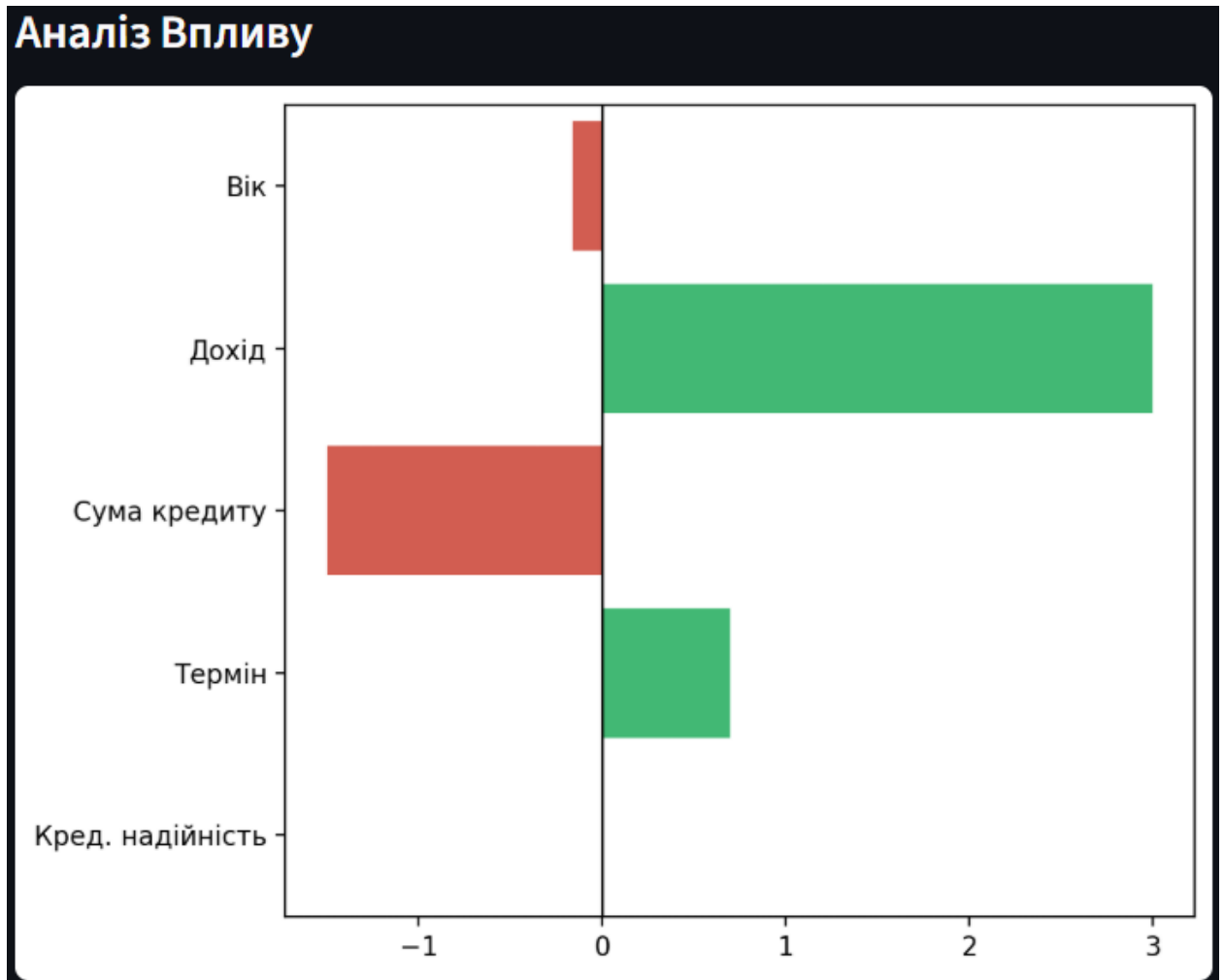


Рисунок 4.1 - Графічний модуль «Аналіз Впливу»

4.3 Експлуатація бази даних та інтерактивне керування реєстрами

У процесі експлуатації системи всі оформлені кредитні договори після натискання кнопки «Оформити кредит» автоматично записуються у файл бази даних `clients_db.csv` через функцію `save_loan_record()` [10]. Запис отримує часовий штамп та базовий статус «Виплачується».

Для моніторингу та управління кредитним портфелем банку в нижній частині інтерфейсу програми реалізовано інтерактивний інструмент `st.data_editor` [12]. Він підтягує поточний вміст бази даних та надає ризик-

менеджеру можливість керування статусами. Стовпці «name», «amount», «score» та «date» заблоковані для редагування з метою фінансової безпеки, а стовпець «status» реалізований у вигляді випадаючого списку з трьома опціями: «Виплачується», «Виплачено» та «Дефолт».

При зміні статусу договору та натисканні кнопки «Оновити статус у базі» дані персистентно перезаписуються у файл, що змінює логіку оцінювання цього клієнта при його наступному зверненні. Реєстр виданих кредитів зображено на рисунку 4.2.

	name	amount	score	date	Status
0	Іванов І. І.	60000	82	2026-05-28 16:05	Виплачується
1	Мартинівська А. В.	50000	69	2026-05-28 16:07	Виплачується
					Виплачується
					Виплачено
					Дефолт

Оновити статус у базі

Рисунок 4.2 – Реєстр виданих кредитів

4.4 Висновки до розділу

У четвертому розділі було проведено експлуатаційне та експериментальне дослідження розробленого програмного забезпечення. Результати функціонального тестування підтвердили безпомилковість роботи вбудованої бізнес-логіки первинного відсікання ризиків (контроль індексу ДТІ) та автоматичного розрахунку внутрішньої кредитної історії. Наведено математичний опис конвеєра аналізу впливу ознак за допомогою правила інверсії ваг регресії. Описано принципи взаємодії з базою даних через інтерактивну таблицю-редактор статусів договорів.

ВИСНОВКИ

У дипломній роботі розроблено та програмно реалізовано інтелектуальну систему опрацювання даних для автоматизованого аналізу кредитних ризиків фінансової установи. За результатами виконання дослідження та проєктування було отримано такі висновки.

Проведено аналіз сутності кредитних ризиків та доведено необхідність автоматизації етапу банківського андеррайтингу за допомогою методів машинного навчання з метою мінімізації помилок людського фактора.

Програмно реалізовано дворівневий фільтр прийняття рішень, який на першому етапі відсікає ризикових клієнтів за жорстким банківським правилом (якщо індекс довгового навантаження $DTI > 50\%$), а на другому – оцінює ймовірність дефолту за допомогою сигмоїдальної моделі логістичної регресії Scikit-learn. Оптимізація моделі дозволила підвищити достовірність класифікаційних рішень, досягши показника точності Accuracy = 84% що мінімізує ризик хибнопозитивних оцінок.

Розроблено інтерактивні модулі візуалізації («Аналіз Впливу»), підсистему динамічного ведення локального реєстру угод та модуль автоматичного формування офіційних фінансових звітів у форматі PDF. Це дозволило повністю виконати завдання з мінімізації часових витрат на формування аналітичної звітності андеррайтерів, скоротивши час генерації підсумкового документа з 15–20 хвилин (в ручному режимі) до 1.5–2 секунд в автоматизованому.

Тестування та експериментальні дослідження ПЗ підтвердили стабільність, високу швидкість відгуку системи (до 2 секунд) та її повну готовність до практичного використання у фінансово-кредитній сфері.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Хома І., Лавренюк А., Лук'янський О. Регресійний аналіз впливу структурних компонентів банківського кредитування на динаміку проблемних кредитів та на рівень кредитного ризику. Економіка та суспільство. 2026. URL: <https://economyandsociety.in.ua/index.php/journal/article/view/7731> (дата звернення: 30.04.2026).
2. Данік Н., Торлопов А. Впровадження штучного інтелекту в банківську сферу: revolution і в обслуговуванні та безпеці. ResearchGate. 2025. URL: https://www.researchgate.net/publication/392319259_Vprovadzenna_stuchnogo_intelektu_v_bankivsku_sferu_revolucia_v_obsługovuvanni_ta_bezpechi Introducing artificial intelligence into banking a revolution in service and security (дата звернення: 30.04.2026).
3. Владика Ю., Бегченкова А. Кредитні операції банків та кредитна лінія. Економіка та суспільство. 2023. URL: <https://economyandsociety.in.ua/index.php/journal/article/view/2310> (дата звернення: 01.05.2026).
4. Солошенко О. М. Інноваційні практики та інноваційний потенціал методології кредитного скорингу компаній мікрокредитування. Економіка та менеджмент. 2024. URL: https://cms.economics-management.e-u.edu.ua/uploads/3_20_423840e35e.pdf (дата звернення: 01.05.2026).
5. Цифрові ініціативи у трансформації фінансового обліку та аналізу / О. А. Швиданенко та ін. Економічний синергізм. 2026. URL: https://economicscience.net/wp-content/uploads/2026/02/2.26._topic_Oleg-Shvydanenko-Ihor-Dzemishkevych-Oleksandr-Ponomarenko-351-364.pdf (дата звернення: 02.05.2026).
6. Ганзюк С. М., Клічко Я. М. Вплив війни на кредитну політику та динаміку кредитних ризиків банків України. Економічний вісник ДДТУ. 2026.

URL: <https://econvisnyk.dstu.dp.ua/article/view/347847> (дата звернення: 02.05.2026).

7. Павлюченко Д. М. Вплив штучного інтелекту та машинного навчання на банківські послуги. Академічне бачення. 2024. URL: <https://academy-vision.org/index.php/av/article/view/1277> (дата звернення: 02.05.2026).

8. Нагорний П. Д., Смаль В. В. Вплив структури кредитного портфеля на ефективність кредитної діяльності банку. Трансформаційні процеси в економіці. 2024. URL: <https://transformations.in.ua/index.php/journal/article/view/85> (дата звернення: 02.05.2026).

9. Задворних С., Мартьянов Д. Наслідки застосування технологій Big Data для стійкості фінансових установ. Acta Academiae Beregsasiensis. Economics. 2026. URL: <https://aab-economics.kmf.uz.ua/aabe/article/view/394> (дата звернення: 03.05.2026).

10. Credit Risk Prediction Using Machine Learning and Deep Learning: A Study on Credit Card Customers / V. Chang et al. Risks. 2024. Vol. 12, no. 11. URL: <https://www.mdpi.com/2227-9091/12/11/174> (date of access: 03.05.2026).

11. Almisshal B., Bulut H. I. The dynamics of cryptocurrency market and data analytics from behavioral finance perspective. Journal of Behavioral and Experimental Finance. 2026. Vol. 49. URL: <https://ideas.repec.org/a/eee/beexfi/v49y2026ics2214635025001078.html> (date of access: 04.05.2026).

12. Ashofteh A. Big Data for Credit Risk Analysis: Efficient Machine Learning Models Using PySpark : Author Accepted Manuscript. Universidade Nova de Lisboa. 2023. URL: https://research.unl.pt/ws/portalfiles/portal/74806826/Big_Data_for_Credit_Risk_Analysis_Efficient_Machine_Learning_Models_Using_PySpark_Author_Accepted_Manuscript.pdf (date of access: 04.05.2026).

13. Tweneboah Senzu E. Theoretical perspective of dynamic credit risk analysis and lending model; effective to enterprises of fragile economy. Munich Personal RePEc Archive. 2019. URL: https://mpra.ub.uni-muenchen.de/91789/1/MPRA_paper_91789.pdf (date of access: 04.05.2026).

14. Mayorga Lira S. D., Andrade-Arenas L., Cano Lengua M. A. Credit Risk Analysis: Using Artificial Intelligence in a Web Application. ResearchGate. 2023. URL: https://www.researchgate.net/publication/369360024_Credit_Risk_Analysis_Using_Artificial_Intelligence_in_a_Web_Application (date of access: 04.05.2026).

15. Priscilla Samuel Nwachukwu, Onyeka Kelvin Chima, Chinelo Harriet Okolo. Credit Risk Assessment Models in Commercial Banking: Approaches, Evolution, and a Proposed Hybrid Risk Evaluation Framework. Engineering Technology Journal. 2025. URL: <https://everant.org/index.php/etj/article/view/2211/1614> (date of access: 05.05.2026).

16. Siddiqi N. Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards. VDOC.PUB. 2017. URL: <https://vdoc.pub/documents/intelligent-credit-scoring-building-and-implementing-better-creditrisk-scorecards-ro7mdphvf280> (date of access: 05.05.2026).

17. Greene W. H. Econometric Analysis of Financial Risk Systems. VDOC.PUB. 2017. URL: <https://vdoc.pub/download/econometric-analysis-60bmn3t86gm0> (date of access: 05.05.2026).

18. Credit Risk Research: Review and Agenda / S. Zamore et al. Emerging Markets Finance and Trade. 2018. Vol. 54, no. 4. URL: <https://ideas.repec.org/a/mes/emfitr/v54y2018i4p811-835.html> (date of access: 06.05.2026).

19. Braunsberger C., Aschauer E. Corporate Failure Prediction: A Literature Review of Altman Z-Score and Machine Learning Models Within a Technology Adoption Framework. Journal of Risk and Financial Management. 2025. Vol. 18,

no. 8. URL: <https://www.mdpi.com/1911-8074/18/8/465> (date of access: 06.05.2026).

20. Principles for the management of credit risk / Basel Committee on Banking Supervision. Bank for International Settlements. 2025. URL: <https://www.bis.org/bcbs/publ/d595.htm> (date of access: 06.05.2026).

21. Contingent Liabilities Risk Management: A Credit Risk Analysis Framework for Sovereign Guarantees and On-Lending / World Bank Group. PPP Legal Resource Center. 2016. URL: <https://ppp.worldbank.org/library/contingent-liabilities-risk-management-credit-risk-analysis-framework-sovereign-guarantees-and-lending> (date of access: 07.05.2026).

22. The Use of Alternative Data in Credit Risk Assessment: Opportunities, Risks, and Challenges / World Bank Group. Open Knowledge Repository. 2025. URL: <https://openknowledge.worldbank.org/entities/publication/7738aa96-6c1b-4fa3-86cb-be30f69d5a30> (date of access: 07.05.2026).

23. Moody's Analytics, Kan O. B. Credit Risk Calculator: User Manual. Moody's. URL: <https://www.moody's.com/sites/products/ProductAttachments/credit%20risk%20calculator%20user%20manual.pdf> (date of access: 07.05.2026).

24. Управління ризиками базових банківських операцій / А. О. Єпіфанов та ін. eSSUIR. 2012. URL: <https://essuir.sumdu.edu.ua/server/api/core/bitstreams/6cbae1e8-32c1-4e8e-8e5b-62c11fa71d4e/content> (дата звернення: 07.05.2026).

25. Ерастов В., Юхименко В. Управління кредитним ризиком банку: Теоретичні засади та шляхи вдосконалення. Економіка та суспільство. 2025. URL: <https://economyandsociety.in.ua/index.php/journal/article/view/6100> (дата звернення: 08.05.2026).

26. Долінський Л. Фінансовий ризик-менеджмент : навч. посіб. ФПК. 2022. URL: https://fpk.in.ua/images/biblioteka/4bac_finan/Dolinskyy.-Finansovyy-ryzyk-menedzhment-2022_compressed.pdf (дата звернення: 08.05.2026).

27. Best credit risk analysis software : knowledge base. Credit Benchmark. URL: <https://www.creditbenchmark.com/knowledge-base/best-credit-risk-analysis-software/> (date of access: 08.05.2026).
28. SAS Risk Management for Banking: Asset and Liability Management Solution : офіційний сайт. URL: https://www.sas.com/de_at/solutions/risk-management/solution/asset-liability-management.html (date of access: 08.05.2026).
29. Streamlit: The Fastest Way to Build and Share Data Apps : official documentation. URL: <https://docs.streamlit.io/> (date of access: 09.05.2026).
30. FPDF186 Class Reference : official documentation. URL: <http://fpdf.org> (date of access: 09.05.2026).
31. Preprocessing data with Scikit-learn : user guide guidelines. 2026. URL: <https://scikit-learn.org/stable/modules/preprocessing.html> (date of access: 09.05.2026).
32. PySpark / Apache Spark : open-source project. URL: <http://apache.org> (date of access: 09.05.2026).
33. Sharma N. Покрокове керівництво: How to Develop Credit Risk Analytics Software. LinkedIn Pulse. 2025. URL: <https://www.linkedin.com/pulse/how-develop-credit-risk-analytics-software-guide-financial-sharma-9vtpf> (date of access: 10.05.2026).
34. Datasets for Credit Risk Modeling : open data repository. ListenData. 2024. URL: <https://www.listendata.com/2019/08/datasets-for-credit-risk-modeling.html> (date of access: 10.05.2026).
35. Understanding Credit Risk : education guide. Investopedia. URL: <https://www.investopedia.com/terms/a/altman.asp> (date of access: 10.05.2026).
36. Commercial Bank Examination Manual / Board of Governors of the Federal Reserve System. Federal Reserve. 2025. URL: <http://federalreserve.gov> (date of access: 10.05.2026).

ДОДАТОК А
Текст програми

A.1 Текст файла `model_utils.py`

```

import pandas as pd
import os
import numpy as np
from fpdf import FPDF
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

DB_FILE = 'clients_db.csv'

def init_db():
    if not os.path.exists(DB_FILE):
        df = pd.DataFrame(columns=['name', 'amount', 'score', 'date', 'status'])
        df.to_csv(DB_FILE, index=False)

def train_default_model():

    X_train = pd.DataFrame([
        [20, 20000, 50000, 1, 1],
        [50, 100000, 10000, 5, 5]
    ], columns=['age', 'income', 'loan_amount', 'termin', 'history'])
    y_train = [1, 0]

    scaler = StandardScaler().fit(X_train)

    model = LogisticRegression()
    model.fit(scaler.transform(X_train), y_train)

    model.coef_ = np.array([[-0.2, -1.0, 1, -0.7, -2.5]])
    model.intercept_ = np.array([0.5])

    return model, scaler

def get_client_history(name):
    if name == "New" or not os.path.exists(DB_FILE):
        return 3

    df = pd.read_csv(DB_FILE)
    client_records = df[df['name'].str.lower() == name.lower()]

    if client_records.empty:
        return 3

```

```

if "Дефолт" in client_records['status'].values:
    return 1

    closed_loans      =      len(client_records[client_records['status']      ==
"Виплачено"])
    new_rating = min(3 + closed_loans, 5)
    return new_rating

def save_loan_record(name, amount, score):
    new_data      =      pd.DataFrame([[name,      amount,      score,
pd.Timestamp.now().strftime("%Y-%m-%d %H:%M"), "Виплачується"]],
        columns=['name', 'amount', 'score', 'date', 'status'])
    if os.path.exists(DB_FILE):
        df = pd.read_csv(DB_FILE)
        df = pd.concat([df, new_data], ignore_index=True)
    else:
        df = new_data
    df.to_csv(DB_FILE, index=False)

def reset_database():
    if os.path.exists(DB_FILE):
        os.remove(DB_FILE)

def generate_pdf_report(data, score, verdict, payment, dti):
    pdf = FPDF()
    pdf.add_page()

    pdf.add_font('DejaVu', '', 'DejaVuSans.ttf', uni=True)
    pdf.set_font('DejaVu', '', 14)

    pdf.set_font("DejaVu", "", 20)
    pdf.cell(200, 15, txt="Звіт про аналіз кредитного ризику", ln=True,
align='C')
    pdf.line(10, 25, 200, 25)

    pdf.ln(10)
    pdf.set_font("DejaVu", "", 12)
    pdf.cell(200, 10, txt=f"Дата: {pd.Timestamp.now().strftime('%Y-%m-
%d %H:%M')}", ln=True)

    pdf.ln(5)
    for key, value in data.items():
        pdf.cell(200, 9, txt=f"{key}: {str(value)}", ln=True)

    pdf.ln(5)

```

```

pdf.set_font("DejaVu", "", 14)
pdf.cell(200, 10, txt="Фінансові подробиці:", ln=True)
pdf.set_font("DejaVu", "", 14)
pdf.cell(200, 9, txt=f"Щомісячний платіж: {payment:,.2f} UAH",
ln=True)
pdf.cell(200, 9, txt=f"Навантаження на платіж (DTI): {dti:.1f}%",
ln=True)

pdf.ln(10)
pdf.set_font("DejaVu", "", 16)
pdf.cell(200, 10, txt=f"Бал надійності: {score}/100", ln=True)
pdf.cell(200, 10, txt=f"Вердикт системи: {verdict}", ln=True)
return pdf.output(dest='S').encode('latin-1')

```

A.2 Текст файлу app.py

```

import streamlit as st
import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from model_utils import train_default_model, generate_pdf_report,
get_client_history, save_loan_record, reset_database, init_db

DB_FILE = 'clients_db.csv'

verdict = "Не визначено"
payment = 0
dti = 0
score = 0

if "name_input" not in st.session_state:
    st.session_state.name_input = ""

if "show_success" not in st.session_state:
    st.session_state.show_success = False

if 'history_value' not in st.session_state:
    st.session_state.history_value = 3

if "last_client" not in st.session_state:
    st.session_state.last_client = "Guest"

if "download_pdf" not in st.session_state:

```

```

st.session_state.download_pdf = False

st.set_page_config(page_title="Credit Risk System", layout="wide")
st.title("Аналіз кредитних ризиків")

@st.cache_resource
def load_data_and_model():
    return train_default_model()

model, scaler = load_data_and_model()

st.sidebar.header("Параметр клієнта")
client_name = st.sidebar.text_input("ПІБ клієнта", key="name_input")

auto_rating = get_client_history(client_name)
if client_name != "New":
    st.session_state.history_value = int(auto_rating)
else:
    st.session_state.history_value = 3

age = st.sidebar.slider("Вік", 18, 80, 18, key="age_input")
income = st.sidebar.number_input("Річний дохід", 0, 2000000, 0,
key="income_input")
loan_amount = st.sidebar.number_input("Сума кредиту", 0, 5000000, 0,
key="loan_input")
termin = st.sidebar.slider("Термін кредиту(років)", 1, 30, 1,
key="termin_input")
history = st.sidebar.slider("Кредитна надійність(1-5)", 1, 5,
value=st.session_state.history_value)

if st.sidebar.button("Очистити базу даних"):
    reset_database()
    st.sidebar.warning("Базу видалено. Перезавантажте сторінку (R)")

if income > 0 and loan_amount > 0:
    interest_rate = 0.15
    monthly_rate = interest_rate / 12
    months = termin * 12
    payment = (loan_amount * monthly_rate) / (1 - (1 + monthly_rate)**(-
months))
    dti = (payment / (income / 12)) * 100

input_df = pd.DataFrame(
    [[age, income, loan_amount, termin, history]],
    columns=['age', 'income', 'loan_amount', 'termin', 'history'])

```

```

input_scaled = scaler.transform(input_df)

risk_prob = model.predict_proba(input_scaled)[:, 1]
score = int((1 - risk_prob[0]) * 100)

col1, col2 = st.columns(2)

with col1:
    st.subheader("Результат оцінки")
    st.metric(label="Надійність позичальника", value=f"{score}/100")

    st.write(f"Щомісячний платіж: **{payment:,.2f} грн / мес.**")
    st.write(f"Навантаження на дохід (DTI): **{dti:.1f}%**")

    if dti > 50:
        st.error("Критичне навантаження: платіж більше 50% доходу!")
    elif score > 60:
        st.success("Рекомендовано схвалення")
        verdict = "Схвалено"
    else:
        st.warning("Потрібна додаткова перевірка")
        verdict = "Потрібна додаткова перевірка"

def process_loan():

    if dti > 50:
        st.session_state["show_success"] = "Error_dti"
        return

    name_to_save = st.session_state.name_input

    if name_to_save and name_to_save != "Guest":

        st.session_state["pdf_data"] = {"ПІБ Клієнта": name_to_save,
            "Вік": age,
            "Річний дохід": f"{income:,.2f} грн",
            "Сума кредиту": f"{loan_amount:,.2f} грн",
            "Термін": f"{termin} років",
            "Кредитна надійність(1-5)": history}
        st.session_state["pdf_score"] = score
        st.session_state["pdf_verdict"] = verdict
        st.session_state["pdf_payment"] = payment

```

```

st.session_state["pdf_dti"] = dti

save_loan_record(name_to_save, loan_amount, score)

st.session_state["download_pdf"] = True

st.session_state["name_input"] = ""
st.session_state["age_input"] = 18
st.session_state["income_input"] = 0
st.session_state["loan_input"] = 0
st.session_state["termin_input"] = 1
st.session_state["history_value"] = 3

st.session_state["show_success"] = True
else:
    st.session_state["show_success"] = "Error"

st.button("Оформити кредит", on_click=process_loan)

if st.session_state.get("show_success") == True:

    st.toast("Дані збережено в базу.")
    st.session_state["show_success"] = False
elif st.session_state.get("show_success") == "Error_dti":
    st.error("Неможливо оформити через критичне навантаження на платіж")
    st.session_state["show_success"] = False
elif st.session_state.get("show_success") == "Error":
    st.error("ПІБ клієнта не введено!")
    st.session_state["show_success"] = False

with col2:
    st.subheader("Аналіз Впливу")
    features = ['Вік', 'Дохід', 'Сума кредиту', 'Термін', 'Кред. надійність']
    current_impact = -(model.coef_ * input_scaled).flatten()

    fig, ax = plt.subplots()

    colors = ['#2ecc71' if x > 0 else '#e74c3c' for x in current_impact]

    sns.barplot(x=current_impact, y=features, palette=colors, ax=ax)

    plt.axvline(0, color='black', linewidth=1)

```

```

plt.tight_layout()

st.pyplot(fig)
else:
    st.info("Введіть дані клієнта для аналізу")

init_db()

st.divider()
st.subheader("Реєстр виданих кредитів")

if os.path.exists(DB_FILE):
    df = pd.read_csv(DB_FILE)

edited_df = st.data_editor(
    df,
    column_config={
        "status": st.column_config.SelectboxColumn(
            "Status",
            options=["Виплачується", "Виплачено", "Дефолт"],
            required=True,
        )
    },
    disabled=["name", "amount", "score", "date"],
    use_container_width=True,
    height=250,
    key="main_editor"
)

if st.button("Оновити статус у базі"):
    edited_df.to_csv(DB_FILE, index=False)
    st.success("Базу даних оновлено.")
    st.rerun()

if st.session_state.get("download_pdf"):
    st.divider()
    st.subheader("Фінальний документ")

    pdf_bytes = generate_pdf_report(
        st.session_state["pdf_data"],
        st.session_state["pdf_score"],
        st.session_state["pdf_verdict"],
        st.session_state["pdf_payment"],
        st.session_state["pdf_dti"]
    )

```

```
)  
  
st.download_button(  
    label="Завантажити PDF",  
    data=pdf_bytes,  
    file_name="report.pdf",  
    mime="application/pdf"  
)  
st.session_state["download_pdf"] = False
```

ДОДАТОК Б
Слайди презентації

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
Факультет комп'ютерних наук і технологій
Кафедра програмних засобів

Дипломна кваліфікаційна робота бакалавра
Тема: «Створення програмної системи
опрацювання
даних для аналізу кредитних ризиків»

Виконав ст. гр. КНТ-212
Керівник доц.

Данило БОРЗІЙ
Олена ПОДКОВАЛІХІНА

Рисунок Б.1 – Слайд 1

Об'єкт, предмет, мета роботи

2

- **Об'єкт дослідження:** процеси застосування алгоритмів та комп'ютерних обчислень для створення програмних систем оцінювання ризиків.
- **Предмет дослідження:** методи та алгоритми машинного навчання (Machine Learning), моделі бінарної класифікації, а також архітектурні рішення та програмні засоби автоматизованого оцінювання ризиків.
- **Мета:** підвищення достовірності класифікаційних рішень при оцінюванні ризиків та мінімізація часових витрат на формування аналітичної звітності шляхом розробки програмного комплексу інтелектуального аналізу даних із використанням методів машинного навчання та автоматизованих модулів генерації звітів.

Рисунок Б.2 – Слайд 2

Порівняльний аналіз існуючого програмного забезпечення

Параметри порівняння	SAS Risk Management	Credit Benchmark	Розроблювана система
Основний клас ПЗ	Промислова аналітична платформа	Глобальний інформаційний сервіс	Інтерактивний локальний застосунок
Точність розрахунку бізнес-правил	Висока	Середня	Висока
Швидкість обробки заявки	Низька	Висока	Максимальна
Вимоги до обчислювальних ресурсів	Максимальні	Середні	Мінімальні
Фінансова доступність	Низька	Низька	Максимальна
Прозорість математичних моделей	Низька	Середня	Максимальна

Рисунок Б.3 – Слайд 3



Рисунок Б.4 – Слайд 4

Специфікація нефункціональних вимог до системи

Категорія	Метрика	Засіб реалізації
Швидкість обробки даних	Час відгуку системи при розрахунку ризику (до 2 секунд)	Масштабування StandardScaler та лінійні обчислення моделі Scikit-learn
Надійність збереження даних	Збереження цілісності реєстру клієнтів після закриття програми	Персистентний запис інформації у локальний файл формату CSV через Pandas
Кросплатформеність та сумісність	Максимальність запуску системи на ОС Windows, Linux та macOS	Веб-архітектура інтерфейсу на базі універсального фреймворку Streamlit
Зручність використання	Час на повне введення однієї анкети аналітиком (до 1 хвилини)	Роздільна структура графічного інтерфейсу з бічною панеллю введення
Коректність генерації документів	Стандарткова читабельність тексту українською мовою у файлі	Інтеграція Unicode-шрифту з підтримкою кирилиці (DejaVuSans.ttf) у модуль PDF

Рисунок Б.5 – Слайд 5

Математичні методи машинного навчання у системі

- ▶ Стандартизація даних (StandardScaler):
$$Z = \frac{(x - \mu)}{\sigma}$$
- ▶ Прогноз ймовірності дефолту (Логістична регресія):
$$P(y = 1|X) = \frac{1}{1 + e^{-(w^T X + b)}}$$
- ▶ Коефіцієнти моделі: `coef_ = pr.array([[ -0.2, -1.0, 1.0, -0.7, -2.5]])`

Рисунок Б.6 – Слайд 6

Реалізація фінансової бізнес-логіки

7

- Розрахунок щомісячного ануїтетного платежу:

$$\text{payment} = \frac{\text{loan_amount} \cdot \text{monthly_rate}}{1 + (1 + \text{monthly_rate})^{-\text{months}}}$$

- Розрахунок індексу довгового навантаження:

$$\text{dti} = \frac{\text{payment}}{\frac{\text{income}}{12}} \cdot 100$$

Рисунок Б.7 – Слайд 7

Обґрунтування вибору мови програмування ядра системи

8

Критерій порівняння	PHP	JavaScript (Node.js)	Python
Основна сфера застосування	Класичні вебсайти, CMS	Високонавантажені real-time застосунки, SPA	Веб-розробка, Data Science, машинне навчання
Підтримка математичних та ML бібліотек	Низька	Середня	Максимальна
Швидкість розробки інтелектуального ПЗ	Низька	Середня	Висока
Зручність створення дата-інтерфейсів	Середня	Висока	Максимальна

Рисунок Б.8 – Слайд 8

9

Обґрунтування вибору інструментарію розробки інтерфейсу

Критерій порівняння	Floppy Dash	FastAPI	Streamlit
Архітектурний підхід	Фреймворк для аналітичних вебдодатків (на базі Flask та React)	Мікрофреймворк (сучасний, асинхронний)	Спеціалізований Data App фреймворк
Необхідність розробки окремого фронтенду (HTML/JS)	Ні	Так	Ні
Складність реалізації бізнес-логіки та UI	Висока	Низька	Мінімальна
Вбудовані інструменти візуалізації та звітності	Максимальна	Відсутні	Висока

Рисунок Б.9 – Слайд 9

10

Архітектура програмного забезпечення та бази даних

```
sequenceDiagram
    participant RM as Ризик-менеджер
    participant WI as Веб-інтерфейс (app.py)
    participant AB as Аналітичний бекенд (model_utils.py)

    RM->>WI: 1. Введення ПІБ
    WI->>AB: 2. Перевірка історії (ПІБ)
    AB-->>WI: 3. Повернення рейтингу (1-5)
    WI->>RM: 4. Параметри кредиту (сума, дохід, термін)
    WI->>AB: 5. Валідація та розрахунок DTI
    AB==>>WI: [ Якщо індекс DTI > 50% ]
    WI->>RM: 6. Блокуючий вердикт "Відхилено"
    AB==>>WI: [ Якщо індекс DTI <= 50% ]
    WI->>AB: 7. Запуск класифікатора (x)
    AB-->>WI: 8. Розрахований бал (0-100)
    WI->>RM: 9. Відображення оцінки та аналітичних графіків
    WI->>AB: 10. Кнопка "Оформити"
    AB->>WI: 11. Збереження угоди
    WI->>AB: 12. Генерація PDF-звіту
    WI->>RM: 13. Доступ до PDF
```

Рисунок Б.10 – Слайд 10

Оцінка операційної та економічної ефективності розробки

11

Критерій оцінки	Базовий процес (Ручний режим)	Розроблене ПЗ	Ефект від впровадження
Точність класифікації	72-75% (експертна оцінка)	84%	Підвищено достовірність на 9 – 12%
Час генерації звіту андеррайтера	15-20 хвилин	1.5 - 2 секунди	Прискорення процесів у 600 разів

Рисунок Б.11 – Слайд 11

Модуль графічного аналізу та інтерпретації рішень штучного інтелекту

12

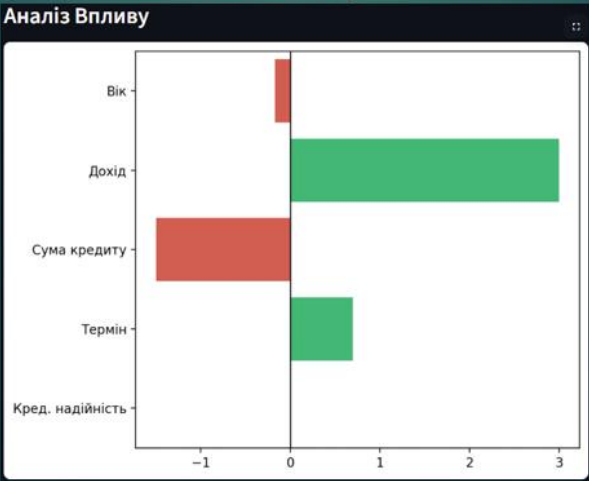


Рисунок Б.12 – Слайд 12

Висновки та результати роботи

13

- ▶ Проведено аналіз сутності кредитних ризиків та автоматизовано етап андеррайтингу.
- ▶ Реалізовано дворівневий фільтр прийняття рішень (контроль ліміту DTI > 50% та модель логістичної регресії Scikit-learn).
- ▶ Досягнуто показника точності моделі Ассигасу = 84%.
- ▶ Час генерації підсумкових фінансових звітів FPDF скорочено з 15 – 20 хвилин до 1,5 – 2 секунд.

Рисунок Б.13 – Слайд 13