ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система підтримки рішень для аналізу фінансових ризиків із використанням машинного навчання»

на здобуття освітнього ступеня бакалавра
зі спеціальності 124 Системний аналіз
(код, найменування спеціальності)
освітньо-професійної програми Системний аналіз
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис) *Нікіта КАЧАН*
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. САД-41
Нікіта КАЧАН
Ім'я, ПРІЗВИЩЕ
Керівник: Михайло КУЗЬМІЧ
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент:
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем на технологій

Ступінь вищої освіти Бакалавр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Системний аналіз

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ
_____Каміла СТОРЧАК
«____» _____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Качан Нікіта Андрійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Система підтримки рішень для аналізу фінансових ризиків із використанням машинного навчання»

керівник кваліфікаційної роботи Михайло КУЗЬМІЧ, доцент кафедри інформаційних систем та технологій, доктор філософії

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «___» _____2025 р. № ____

2. Строк подання кваліфікаційної роботи «___» _____2025 р.

3. Вихідні дані до кваліфікаційної роботи:
   1. Статистичні дані щодо кредитоспроможності клієнтів і рівня дефолтів.
   2. Відкриті датасети з платформ Open Banking, Kaggle, Credit Bureaus.
   3. Методи машинного навчання, зокрема Logistic Regression, Random Forest, XGBoost.
   4. Технології інженерії ознак, кластеризації (KMeans) та їх вплив на точність моделей.
   5. Наукові публікації з питань автоматизації скорингу, боротьби з дисбалансом класів, інтерпретованості моделей.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Визначення основних проблем кредитного скорингу та суб'єктивності рішень.

2. Аналіз наявних підходів до оцінки кредитного ризику в банківській сфері.

3. Формалізація задачі скорингу як задачі бінарної класифікації.\

4. Розробка концептуальної моделі скорингової системи з використанням ML.

5. Проведення кластеризації клієнтів для покращення сегментації.

6. Тестування моделей та порівняльний аналіз за метриками Recall, Precision, F1-score, AUC.

7. Пропозиції щодо впровадження моделей у бізнес-процеси банку.

8. Оцінка ефективності системи та впливу на процес прийняття кредитних рішень.

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «___» _____2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір необхідної літератури	16.03.2025-17.03.2025	Виконано
2	Написання першого розділу	18.03.2025-24.03.2025	Виконано
3	Написання практичної частини (коду)	25.03.2025-30.03.2025	Виконано
4	Написання другого розділу	2.03.2025-14.04.2025	Виконано
5	Написання третього розділу	15.04.2025-22.04.2025	Виконано
6	Написання висновків	23.04.2025-24.04.2025	Виконано
7	Підготовка доповіді	20.05.2025-21.05.2025	Виконано
8			

Здобувач(ка) вищої освіти _____ Нікіта КАЧАН

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Михайло КУЗЬМІЧ

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 61 стор., 12 рис., 1 табл., 15 джерел.

Мета роботи – розробка системи підтримки рішень для оцінки кредитних ризиків з використанням методів машинного навчання, що дозволяє покращити точність прогнозування ймовірності неповернення кредиту та забезпечити ефективну класифікацію клієнтів для потреб фінансових установ.

Об’єкт дослідження – процес оцінювання кредитоспроможності позичальників на основі історичних даних про фінансову поведінку клієнтів.

Предмет дослідження – математичні моделі машинного навчання для передбачення кредитного ризику, особливості їх використання в задачах класифікації, вплив інженерії ознак, а також побудова прикладної системи підтримки рішень.

Короткий зміст роботи:
У сучасних умовах цифровізації та зростаючих обсягів даних фінансові установи потребують ефективних рішень для управління кредитними ризиками. Традиційні скорингові системи часто не здатні врахувати приховані закономірності в поведінці клієнтів, що обмежує їхню точність. У цьому контексті актуальним стає впровадження систем на основі машинного навчання, здатних до самоадаптації та виявлення складних зв’язків у великих наборах даних.

У роботі здійснено повний цикл розробки системи підтримки рішень: від аналізу предметної області та підготовки даних до порівняння ефективності різних моделей машинного навчання, зокрема Logistic Regression, Random Forest та XGBoost. Особливу увагу приділено метриці Recall як ключовій у контексті ризик-менеджменту. Застосовано також метод кластеризації KMeans для виявлення прихованих груп клієнтів на етапі інженерії ознак.

Результати дослідження свідчать, що модель XGBoost демонструє найкращий баланс між точністю та стабільністю прогнозування ризиків. Запропонована система може бути інтегрована в процес прийняття рішень кредитного менеджера та використовуватись для автоматизованої попередньої оцінки заявок. Практична цінність полягає в підвищенні ефективності кредитного скорингу, зниженні рівня дефолтів і зростанні довіри до автоматизованих систем аналізу.

КЛЮЧОВІ СЛОВА:
кредитний ризик, машинне навчання, класифікація, XGBoost, Random Forest, Logistic Regression, Recall, скоринг, підтримка рішень, інженерія ознак, кластеризація, KMeans, фінанси, кредитоспроможність.

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня бакалавра (магістра)

Направляється здобувач Качан Н.А. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 124 Системний аналіз
(код, найменування спеціальності)
освітньо-професійної програми «Інтелектуальні системи управління»
(назва)
на тему: «Система підтримки рішень для аналізу фінансових ризиків із використанням машинного навчання».

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____ Катерина НЕСТЕРЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Качан Нікіта у процесі виконання кваліфікаційної роботи продемонстрував глибоке розуміння теми, високий рівень самостійності та вміння працювати з сучасними технологіями машинного навчання. Робота присвячена актуальній проблемі – підвищенню ефективності оцінки кредитних ризиків, що має важливе значення для фінансових установ в умовах швидких змін ринку. У роботі вдало поєднано теоретичний аналіз підходів до кредитного скорингу із практичною реалізацією моделей класифікації. Окрему увагуприділено проблемі дисбалансу класів і використанню кластеризації (KMeans) як інструмент покращення ознак. Результати моделювання та оцінки ефективності моделей підтверджують доцільність використання запропонованого підходу.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Качан Нікіта Андрійович на оцінку «відмінно» та присвоїти йому кваліфікацію «бакалавр із Системного аналізу».

Керівник кваліфікаційної роботи _____ Михайло КУЗЬМІЧ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

«____» _____20___року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач(ка) *Качан Н.А* допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою ІСТ _____ Каміла СТОРЧАК
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня бакалавра (магістра)

здобувача(ки) вищої освіти Качан Нікіта Андрійович
(прізвище, ім'я, по батькові)
на тему «Система підтримки рішень для аналізу фінансових ризиків із використанням машинного навчання»

Актуальність.

У сучасних умовах цифровізації банківської діяльності проблема ефективної оцінки кредитних ризиків набуває особливого значення. Застосування методів машинного навчання у фінансовій аналітиці дозволяє підвищити точність прогнозування й автоматизувати прийняття рішень. Тема обраної роботи є надзвичайно актуальною, оскільки вона відповідає сучасним викликам банківської системи, а також інтегрує новітні технології у процес кредитного скорингу.

Позитивні сторони

1.У роботі проведено глибокий аналіз теоретичних засад кредитного скорингу

2.Автором реалізовано практичну систему підтримки прийняття рішень із використанням алгоритмів машинного навчання

3.Робота має прикладне значення, демонструє вміння працювати з великими обсягами даних.

Недоліки.

1.Деякі розділи роботи містять надмірну кількість загальної інформації.

2.Відсутній блок критичної оцінки ризиків впровадження моделі в реальному банківському середовищі

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи бакалаврської (магістерської).

Висновок: *кваліфікаційна робота на здобуття ступеня бакалавра (магістра) заслуговує оцінку "*відмінно*", а здобувач* Качан Нікіта Андрійович *заслуговує присвоєння кваліфікації:* бакалавр із Системного аналізу

Рецензент:

науковий ступінь, вчене звання	*підпис*	*Ім'я, ПРІЗВИЩЕ*

ЗМІСТ

ВСТУП

У сучасних умовах майже всі в усіх сферах діяльності застосовуються технології штучного інтелекту та машинного навчання, вони допомагають у таких галузях як: фінанси(кредитний скоринг, виявлення шахрайства), маркетинг(рекомендаційні системи, аналіз відгуків клієнтів), медицина(діагностика хвороб, прогнозування перебігу захворювання) тощо.

Завдяки цим новим технологіям великі та малі компанії мають змогу обробляти велику кількість даних, проаналізувати та спланувати подальші дії в майбутньому, або уникнути та запобігти негативним наслідкам.

Мета дипломної роботи полягає в тому, щоб:

- Проведено аналіз сучасних методів оцінки кредитоспроможності клієнтів з використанням алгоритмів машинного навчання
- описати та реалізувати відповідні моделі класифікації
- зробити кластеризацію (KMeans), а також провести додаткові етапи обробки даних для покращення моделі та результатів
- провести порівняння всіх обраних моделей за ключовими метриками
- обгрунтувати які метрики підходять найкраще саме в цієї оцінки кредитного скорингу
- здійснити візуалізації результатів для кращої інтерпретації моделі
- сформулювати рекомендації щодо покращення моделі

Об’єкт дослідження дипломної роботи є — оцінка кредитоспроможності клієнтів та прийняття рішень у кредитному аналізі. Предмет дослідження являє собою застосування моделей машинного навчання, кластеризації та додаткових методів обробки даних, що підвищують ефективність моделі.

Щоб досягти поставлених цілей в дослідженні були використані методи машинного навчання такі як: Logistic Regression, Random Forest та XGBoost

які найкраще підходять для бінарної класифікації. Та декілька етапів інженерії ознак що включають в себе Feature Engineering & KMeans.

Перед тренуванням моделі та перевірки результатів була проведена попередня обробка даних завдяки масштабуванню ознак, заповнення пропущених значень та кодування категоріальних змінних. Оцінка ефективності моделей здійснюється за допомогою кросс-валідації та аналізу метрик якості класифікації — Accuracy, Precision, Recall, F1-score.

Практична частина була реалізована завдяки сервісу Google Colab, що використовує мову програмування Python з використанням бібліотек scikit-learn, xgboost, pandas, matplotlib та інших інструментів обробки та візуалізації даних.

Вперше досліджено вплив кластеризації як етапу інженерії ознак на ефективність моделей кредитного скорингу з акцентом на метрику Recall. Запропоновано підхід, що поєднує KMeans та XGBoost для підвищення точності виявлення ризикових клієнтів.

Результати дослідження можуть бути використані для вдосконалення систем підтримки прийняття рішень у банках і фінансових організаціях. Це зменшить кількість необґрунтованих кредитних рішень та фінансових втрат.

Основні результати дослідження було представлено у вигляді тез на науковій студентській конференції в рамках секції №5 Державного університету телекомунікацій.

Дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У роботі було представлено аналіз проблеми, опис методів машинного навчання, практичну реалізацію моделі та оцінку їх ефективності

Застосування методів машинного навчання у кредитному скорингу дозволяє не лише підвищити об’єктивність рішень, а й значно прискорити процес оцінки клієнтів у великих масивах даних. Автоматизовані алгоритми здатні виявляти приховані закономірності, які важко ідентифікувати за допомогою традиційного андеррайтингу. Завдяки цьому зменшується суб’єктивний вплив людського фактора та ймовірність помилкових рішень. Застосування кластеризації в поєднанні з класифікацією відкриває нові можливості в сегментації клієнтів, що є особливо актуальним для побудови адаптивної кредитної політики.

Особливу увагу в роботі приділено проблемі дисбалансу класів, що є типовою для задач прогнозування дефолтів. Було протестовано кілька підходів до вирішення цієї проблеми, зокрема використання параметра class_weight та методу SMOTE. Додатково оцінювався вплив дискримінаційного порогу класифікації на фінальні результати моделей. Це дозволило сформулювати практичні рекомендації для адаптації моделей у змінних умовах ринку.

Також у роботі враховано сучасні вимоги до прозорості та інтерпретованості моделей, що є важливими з точки зору нормативного регулювання у фінансовій сфері. Під час розробки акцент було зроблено не лише на точності, а й на гнучкості та стабільності роботи системи. Результати дослідження мають прикладне значення та можуть бути впроваджені в інформаційні системи банківських установ для покращення процесу ухвалення рішень на основі даних.

Під час роботи над дослідженням також було виявлено, що застосування моделей із імовірнісним виходом дозволяє формувати рейтингову шкалу ризику для кожного клієнта. Це відкриває можливість не тільки ухвалювати рішення про видачу кредиту, а й визначати оптимальні умови обслуговування. У процесі дослідження було дотримано принципів науковості, об’єктивності та відтворюваності експериментів. Усі етапи моделювання були задокументовані та реалізовані з використанням сучасного програмного інструментарію. Практичні результати роботи можуть стати основою для подальших досліджень у сфері фінансової аналітики та інтелектуального аналізу даних.

1 ТЕОРЕТИЧНІ ЗАСАДИ ОЦІНКИ КРЕДИТНИХ РИЗИКІВ

1.1 Історія виникнення кредитного скорингу

У часи, коли ще не було потужних комп'ютерів та провідних технологій усі провідні банки миру використовували послуги андеррайтерів. Андеррайтер може бути як окремі спеціалісти або співробітники які працюють на аутсорсі, так і штатний співробітник великої фінансової компанії. Робота таких спеціалістів полягала в тому, щоб оцінити платіжно спроможність клієнта, але такий метод оцінки є суб’єктивний. На рішення андеррайтера могли вплинути такі упередження як: раса, стать, власні судження, поганий настрій та ще великий перелік факторів. Крім точності висновків такий спосіб оцінки є дуже повільним.

Через це в 1956-му році була розроблена перша модель кредитного скорингу компанією FICO (Fair Isaac Corporation), вона була заснована Білом Фєром та Ерлом Айзеком. Хоча модель була інноваційна для свого часу, вона була призначення лише для певного переліку бізнесів та не могла бути універсальної для всіх галузей.

В 1980-х роках банківська сфера діяльності почала все більше застосовувати нові технології та оцифровувати дані своїх клієнтів, через це використання алгоритмів значно підвищилася та стала більше широко використовуватися.

В 1989 році FICO створила нову та універсальну модель яка підходить для всіх кредиторів. На той час фіксоване ціноутворення було нормою, незважаючи на суттєві відмінності у витратах клієнтів (зокрема, у вартості ризику), а річна відсоткова ставка за кредитом змінювалася лише за рішенням експертів.

В 1990-х роках ринок кредитних моделей став насичуватися більш складними системами, які в першу чергу були розраховані на оцінку клієнтів кредитних карток. Уряд США став спонсором розробки моделей, які були використані в таких компаніях як Fannie Mae и Freddie Mac.

Нові алгоритми оцінки були націленні на оптимизацію процессів та автоматизації оцінки позичальника. Завдяки цьому час на аналіз та рішення значно скоротився, а прийняття рішень стало більш послідовним.

На початку 2000-х років прогрес у обчислювальній потужності та аналізі даних призвів до розробки більш складних статистичних моделей, таких як логістична регресія, щоб точніше прогнозувати ризик дефолту. Банки почали наймати спеціалістів із обробки даних і статистиків для розробки цих моделей, які навчалися на великих наборах даних із сотнями змінних поведінки позичальників. Механізми прийняття кредитних рішень поступово стали невід'ємною частиною технологічних пакетів фінансових установ, дозволяючи приймати рішення в реальному часі на основі широкого діапазону даних про позичальників.

Наступний стрибок у кредитному рейтингу стався з інтеграцією технологій штучного інтелекту (AI) і машинного навчання (ML). Ці інструменти дозволяли кредиторам аналізувати великі обсяги традиційних і нетрадиційних даних (наприклад, активність у соціальних мережах або комунальні платежі), щоб приймати детальніші рішення щодо кредитування.

Системи, керовані штучним інтелектом, можуть оцінювати надійність позичальників поза традиційними критеріями, такими як кредитні рейтинги, розширюючи доступ до кредиту для осіб з обмеженою кредитною історією. Штучний інтелект також покращує оцінку ризиків, визначаючи моделі поведінки позичальників, які традиційні моделі можуть не помітити. Проте з новим поколінням моделей виникло кілька проблем, зокрема відсутність прозорості та потенційний ризик упередженості моделі, що може призвести до дискримінації.

Дедалі більше використання альтернативних даних спонукало до розробки моделей ШІ. Краще використовуючи великі набори даних із нелінійно корельованими та різноманітними даними, моделі ШІ довели свою перевагу над традиційними статистичними моделями. Альтернативні дані для кредитного скорингу зазвичай включають:

1. відкриті банківські дані
2. історію платежів за комунальні послуги та орендаря

3. відкриті адміністративні дані
4. цифровий слід

Такий підхід виявився особливо корисним для груп населення, які недостатньо обслуговуються, наприклад, нещодавніх іммігрантів або працівників великої економіки, які інакше були б виключені з традиційних систем кредитування.

Сьогодні багато кредиторів використовують аналітику даних у реальному часі, щоб миттєво приймати рішення щодо кредитних заявок. Автоматизовані системи тепер можуть інтегрувати альтернативні джерела даних, такі як орендна плата чи рахунки за комунальні послуги, щоб оцінити позичальників, які можуть не мати надійної кредитної історії. Ця зміна зробила кредитування більш інклюзивним, зберігаючи сувору практику управління ризиками.

Крім того, сучасні системи прийняття рішень тепер забезпечують більшу гнучкість у прийнятті рішень за допомогою конфігурованих правил, які можна налаштувати без значного перепрограмування.

Хоча моделі штучного інтелекту існують десятиліттями — починаючи з 1980-х років із логістичною регресією — вони були обмежені меншими наборами даних і меншою обчислювальною потужністю. Навпаки, основи моделей GenAI є відносно новими, починаючи з розробки моделей Transformer компанією Google у 2017 році та випуску BERT у 2018 році. BERT ознаменував прорив у обробці мови та зробив революцію в здатності розуміти контекст за допомогою своєї архітектури на основі Transformer.

У листопаді 2022 року OpenAI випустив ChatGPT, який швидко став противним продуктом завдяки здатності генерувати текст, схожий на людину. Протягом двох місяців воно набрало понад 100 мільйонів користувачів, що зробило його найшвидше зростаючим споживчим додатком в історії. Через два роки моделі GenAI продовжували вдосконалюватись, і кілька нових великих гравців конкурували з ChatGPT: Gemini, Android і Mistral.

1.2 Сутність і структура сучасного кредитного скорингу

Кредитний скоринг – це математична або статистична модель, створена на основі кредитної історії попередніх клієнтів, яку банк використовує для визначення ймовірності того, що окремий потенційний позичальник поверне кредит у узгоджений термін; систематизована шкала якісних та кількісних показників і балів, що використовується для оцінки ймовірності повернення кредиту.

Сучасний кредит скоринг поділяється на 4 типи:

1. аплікаційний скоринг (application scoring) — кредитори оцінюють вашу кредитну заявку за набором критеріїв, які вони мають для свого ідеального клієнта. Рейтинг заявки є результатом цього процесу та допомагає кредитору прийняти рішення.
2. поведінковий скоринг (behavioral scoring) — стосується моделей кредитного скорингу для існуючих клієнтів компанії (на відміну від нових клієнтів), для яких скоринговий агент зібрав історію ефективності. Рахунки можуть оцінюватися щомісяця або щокварталу. Ця додаткова інформація зазвичай дозволяє точніше передбачити ймовірність виникнення простроченої заборгованості за існуючим рахунком. Точніший профіль ризику, у свою чергу, може дозволити компанії підвищити свою ефективність.
3. колекторський скоринг (collection scoring)-модель оцінювання стягнення боргів призначена для оцінки ймовірності платежу клієнтами, які вже перебувають у стані дефолту. Це означає, що цільова аудиторія моделі стягнення складається з клієнтів, які не впоралися зі своїми зобов'язаннями у терміни, узгоджені з кредиторами. Цей тип моделі є інструментом, який допомагає оцінити збитки на основі ймовірності платежу клієнтами, які вже перебувають у стані дефолту.
4. шахрайський скоринг (fraud scoring)-Так званий бал шахрайства визначається на основі правил, які ви склали у своєму наборі правил. Щоб визначити бал шахрайства для пристрою або транзакції, ми додаємо

значення всіх правил, що застосовуються до відповідної транзакції. Чим вищий бал шахрайства, тим більша ймовірність того, що пристрій або транзакція є шахрайськими.

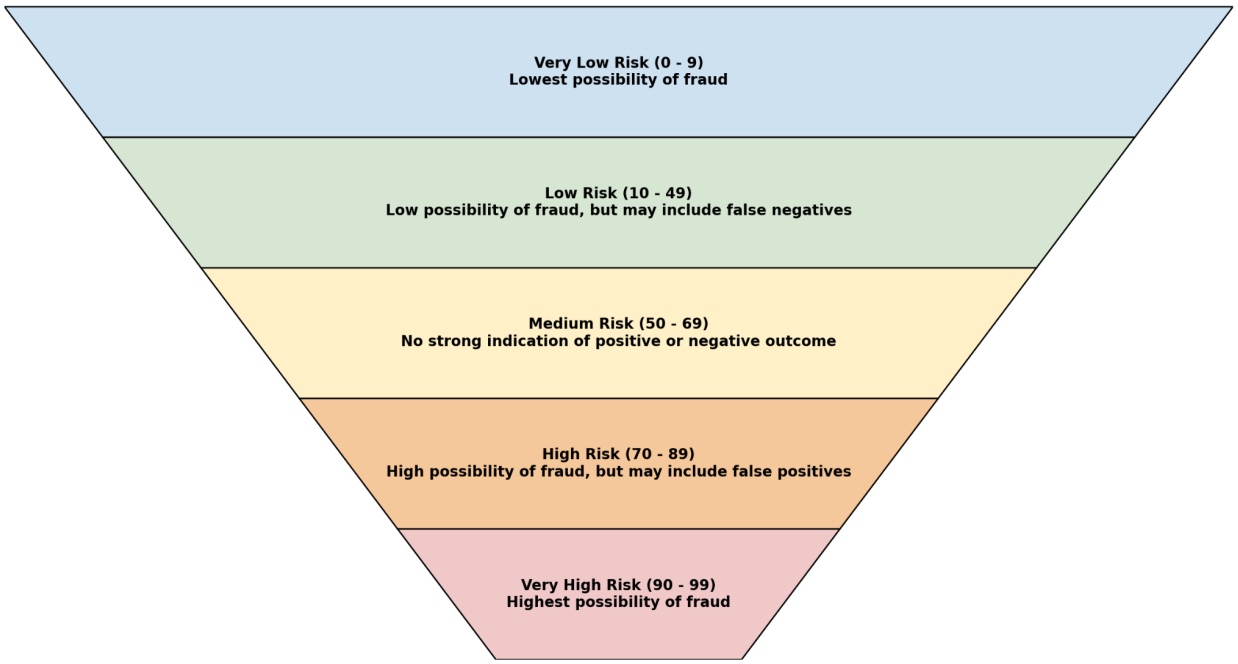


Рис. 1.1 Градація рівнів ризику на основі скорингової оцінки клієнтів

Модель оцінювання включає кілька змінних, які разом можуть оцінити ризик онлайн-транзакції. Багато з цих факторів базуються на персональній ідентифікації, загальних характеристиках шахрайства та тенденціях, що розвиваються. Нижче наведено список найпоширеніших факторів, які впливають на визначення балу шахрайства: збіг платіжної адреси та адреси доставки, збіг BIN, IP-адреса та відстань виставлення рахунків, виявлення проксі, вік електронної пошти.

Ключові етапи формування скоринг моделі полягають в тому щоб зібрати дані, обробити їх, на основі нової обробленої бази даних навчити модель різними видами навчання(Logistic regression, XGBoost, Random Forest), протестувати отриману модель різними метриками та інтегрувати модель з найкращими показниками в систему фінансової компанії. При інтегруванні моделі треба

спиратися на стандарти ISO/IEC, це стандарти якості які застосовуються для управління інформаційною системою, це дуже важливий аспект адже банки обробляють чутливі персональні дані.

Найбільш застосовувані стандарти ISO/IEC є:

1. вимоги до систем управління інформацією(27001)
2. стандарти оцінки якості ПО, що включають в себе зручність, надійність, безпеку тощо(25010)
3. Стандарт корпоративного управління ИТ, яких визначає принципи за якими у великих організаціях управляють інформаційними технологіями (38500)

Складова моделі включає в себе інформаційні зміни(features) такі як: вік, дохід, стаж, кількість кредитів, кредитне навантаження та ще велику кількість мешн впливових ознак які в комбінації з іншими можуть дати більш точну оцінку клієнта. [GJETA, 2024, p. 4].

Крім стандартних ознак, моделі великих міжнародних банків повинні відповідати вимогам Basel III, які були согласовані через кризис 2007-09 років. Стандарти Basel є мінімальними критеріями, які застосовуються к банкам. Ці міри направлені на посилення регулювання та управління ризиками.

Результати моделі інтерпретується як ймовірність дефолту клієнта. В більш великих та складних моделях позичальники класифікуються по рівнях ризику від низького до високого. Завдяки цьому є можливість створити багаторівневу стратегію кредитування. [JFIF, 2015, p. 7].

1.3 Методи оцінки ризику кредитування

Оцінка кредитного ризику набагато складніший та більш суб’єктивний в порівнянні з іншими фінансовими ризиками. Такі ризики як зміна валютного курсу або відсоткова ставка оцінюється всіма учасниками ринку в той час як оцінкою кредитного ризику займається один банк. Через це оцінка позичальника може відрізнятися від банку до банку навіть з застосуванням машинного навчання через те, що в кожного банку своя модель, яка була натренерована на різних даних

з застосуванням різних методів. Проблема суб’єктивності також не дозволяє використовувати такі методи, як статистика та теорія ймовірності. Використання цих методів спрямовано на виявлення закономірностей, які характеризуються повторюваністю, послідуваністю та порядком у процессах.

Через те що кредитний ризик має масовий характер супроводжуючи всі активні операції банку, виникає постійна потреба в оцінці та в удосконаленні методів класифікації. Адже треба чітко відрізняти ризик не повернення боргу, з іншими видами ризику, через різність застосованих методів управління.

З огляду на те, що практична частина данного дослідження зосереджена на оцінці дефолту позичальника, доцільним є розглянути методи управління ризиками на рівні кредитного портфелю.

Кредитний портфель — це сукупність кредитів, які були надані банком на певну дату, але через різні ознаки ще не були погашені.

Формування кредитного портфелю це основний етап для реалізації кредитної політики банку. До нього приступають після того, як визначено мету і стратегію кредитної діяльності установи.

Класифікація методів управління кредитними ризиками на рівня кредитного портфелю поділяється на:

1. диверсифікацію кредитного портфеля — розподіл кредитного портфеля серед великого пола позичальників, які відрізняються за ознаками та умовами діяльності.
2. концентрацію кредитного портфеля — зосередження кредитних операцій в певній галузі.
3. установлення лімітів кредитування — встановлення максимально допустимих розмірів наданих позиків.
4. формування резервів - створення резервів для відшкодування втрат на випадок неповернення кредитів позичальниками
5. сек'юритизація кредитів — фінансування або рефінансування активів шляхом перетворення в торгівельно ліквідну форму через випуск облігацій чи інших цінних паперів.

6. страхування кредитних ризиків — комплекс страхових послуг, які забезпечують захист інтересів кредиторів через ризик неповернення кредитів внаслідок неплатоспроможності клієнта.

Управління ризиками на рівні кредитного портфелю є критично важливим для будь якої фінансової компанії що надає кредитування, що дозволяє планувати та оцінювати стійкість установи до дефолту.

Проте забезпечення ефективного моніторингу ризиків на всіх етапах має необхідність застосовувати локальні методи оцінки на рівні окремого позичальника або транзакції. Локальні методи дають змогу ідентифікувати проблемні заявки на ранній стадії та запобігти прийняттю неправильного рішення. Для цього все більш популярними стали сучасні алгоритми машинного навчання. Завдяки ним у банків є можливість гнучко моделювати залежність між ознаками та змінними та адаптуватися до змін.

В загальну класифікації методів оцінки кредитного ризику з допомогою машинного навчання є метод логістичної регресії, яка історично відноситься до статистичних методів, але в сучасному контексті більше згадується як базова модель машинного навчання. На зміну такому методу прийшли більш гнучкі та точні алгоритми. До списку нових методів входять: XGBoost, Random Forest. Вони показують виші результати по багатьом важливими метриками завдяки новим підходам щодо класифікації та оцінці ознак, які використовуються для машинного навчання.

Такі моделі найбільш широко використовуються в таких задачах як: рівень позичальника, рівень транзакції(fraud scoring), рівень сегментації для покращення ознак.

Однак, використання нових методів навчанне це лише один з ключових факторів точної та гнучкої оцінки. До складових які впливають на рівень моделі входять такі пункти як: ризик перенавчання, регулярна валідація даних, складність інтерпретації, відповідність регулятивним вимогам згідно з Basel та найважливіший аспект це якість даних для навчання. Якщо даних буде мало, або їх якість не буде на високому рівні є ризик того, що при наявності усіх інших

пунктів, модель не зможе якісно виконувати поставленні задачі, або взагалі не навчиться.

Дані, які використовуються для моделей поділяються на кількісні та якісні.

Кількісні характеризуються доходами, віком, заборгованістю.

Якісні ж складаються з репутації, стосунками з банком, намірами тощо.

Хоча більшість моделей базуються на кількісних ознаках, є винятки що інтегрують якісні ознаки через спеціальні ознаки та ваги.

Про створенні моделей для оцінки багато уваги приділяється балансу класів, адже при наявності дисбалансу моделі перенавчаються та не можуть робити точну оцінку. У кредитному скорингу клас дефолту займає(10-20%) це ускладнює процес навчання.

Також для багатьох банків, особливо які планують опанувати новий ринок, наприклад відкрити свої філіали в новій країні або збільшити клієнтську базу завдяки залученню новою вікової категорії. Велику увагу приділяють проблемі холодного старту. Холодний старт це явище, яке виникає, коли у фінансової установи немає історичних даних про нового клієнта. Така явище ускладнює оцінку його кредитоспроможності. Воно характерно для осіб без кредитної історії, нових підприємств, або для клієнтів які змінили свої географію перебування.

Щоб уникнути хибних рішень через недостаток інформації треба використовувати додаткові методи. Один з таких методів є transfer learning. Цей метод використовує запозичені дані, які були використані для виконання інших задач. Найпоширеніший приклад використання transfer learning — використання даних з регіону чий демографічний та економічний показник еквівалентний або схожий на показники нового регіону для якого даних не достатню з метою навчання моделі в майбутньому з використанням даних з потрібного регіону

Для практичного ознайомлення з практичною частиною варто використовувати відкриті бази даних або дата сети. Найчастіше якісні дані можна знайти на Open Banking, Credit Bureaus, Google API. Тоді можна мінімізувати ризик неякісних даних, помилок в заповненні дата сету та валідності.

1.4 Основні метрики класифікації(Recall, Precision, F1-score, Accuracy)

Кожна модель яка була розроблена має свою задачу та результат виконаної роботи(в контексті дослідження це класифікація позичальника). Для різних моделей використовуються різні метрики, які демонструють точність.

Всі базові метрики основані на Confusion Matrix що складається з True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

Наприклад для моделей що ідентифікують ракові пухлини слід використовувати Precision, адже для цієї задачі важливо, щоб False Positive зустрічався частіше, а не навпаки. Адже в такому випадку навіть якщо модель зробить хибний висновок, летальних наслідків буде менше. В сфері кредитних ризиків метрику обирають виходячи з цілей, які були поставленні фінансовими установами.

- Recall — якщо ціль банку це зменшення частки неплатоспроможних клієнтів
- Precision — якщо ціль банку це збільшення клієнтської бази або боїться відмовити надійному клієнту.
- F1-score — золота середина, збалансований варіант між Recall та Precision. Використовується коли потрібна як точність, так і повнота.

Крім базових метрик при оцінці моделі використовують ROC-криву.

ROC-крива це візуальне представлення ефективності моделі за всіма пороговими значеннями. Довга версія назви – «робоча характеристика приймача» – є пережитком радіолокаційного виявлення під час Другої світової війни.

Площа під кривою ROC (AUC) представляє ймовірність того, що модель, якщо їй надати випадково вибраний позитивний і негативний приклади, оцінить позитивний вище, ніж негативний.

Для прикладу візьмемо графік ROC-кривої побудований на основі моделі яку буде використано в практичній частині роботи.1.

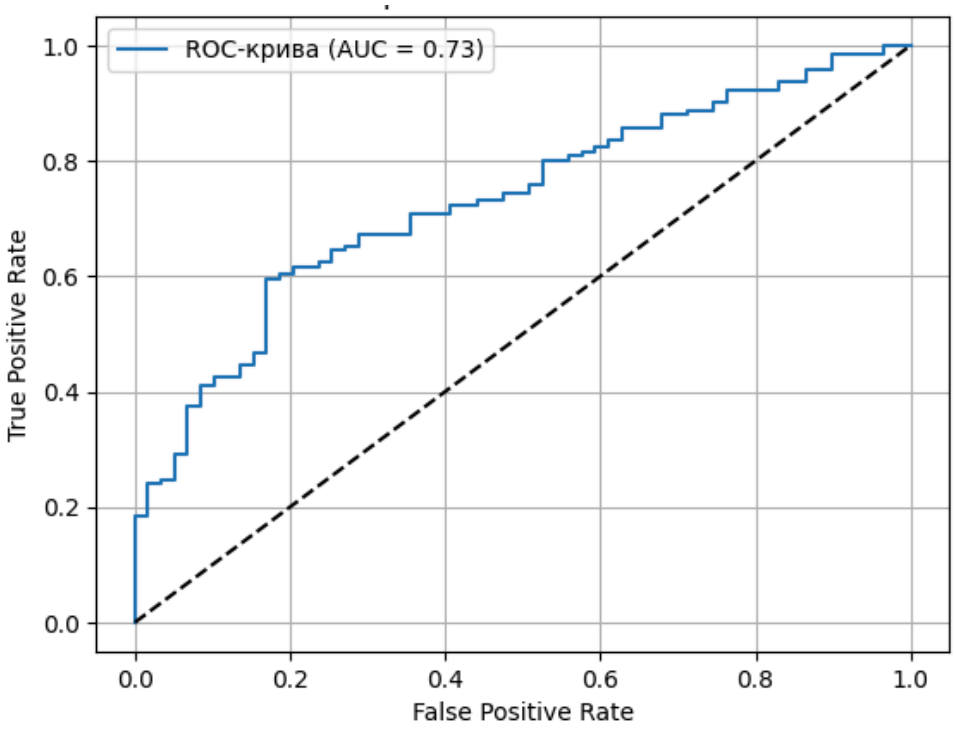


Рис 1.2 ROC-крива для моделі(AUC = 0.73)

На цьому графіку можна побачити коефіцієнт AUC що дорівнює 0. 73, що свідчить про непоганий рівень. Існує 73% ймовірність, що модель поставить вищу ймовірність позитивного класу для випадкового позитивного прикладу, ніж для випадкового негативного.

Gini Coefficient (коефіцієнт Джині) – індикатор нерівності розподілу обсягів банків, що виводиться з кривої Лоренца.

Крива Лоренца – графік, в якому по горизонтальній осі відкладені процент населення від найбідніших верств до найбагатших, а по вертикальній – процент одержуваного ними доходу

Формула якою розраховується коефіцієнт Джині:

$$Gini = 2*AUC-1 \quad (1.1)$$

З цього випливає його основна перевага — простота інтерпретації та висока кореляція з AUC-ROC, з якої він безпосередньо виводиться.

Отриманий результат находиться в межах від -1 до 1, де 1 це точне розділення позитивного та негативного класів, 0 вказує на повну випадковість моделі, а негативні значення на протилежну класифікацію.

Коефіцієнт Джині також можна визначити як площу між кривою ROC і діагоналлю випадкової класифікації. Таким чином, це вважається дуже важливим показником у моделі оцінки, оскільки він дає сукупну оцінку якості класифікації незалежно від вибору порогового значення. Цей показник став модним відповідно до нормативних вимог до моделей ризику в банківській галузі.

У багатьох моделях машинного навчання, таких як логістична регресія, результатом є не прогнозована мітка класу, а скоріше ймовірність належності об'єкта до певного класу. Прийняття остаточного рішення щодо класифікації вимагає встановлення порогу розрізнення, який визначатиме, з яким класом буде пов'язано спостереження.

Це найбільше важливо в питаннях оцінки кредитоспроможності, коли зміна порогу дозволяє впоратися з балансом між точністю і чутливістю моделі, що впливає на співвідношення хибно позитивних або негативних виборів. Краще пояснення цієї ідеї наведено нижче.

У статистиці та машинному навчанні дискримінаційним порогом називають значення, що приймається функцією, що дискримінує, в задачах бінарної класифікації, яке дозволяє розділяти класи. Значення дискримінаційного порога налаштовується таким чином, щоб зменшити кількість помилок класифікації.

Типовим прикладом використання порогу дискримінації є логістична регресія. Виходом моделі логістичної регресії є значення, яке називається рейтингом, яке змінюється в інтервалі 0.. 1 і може бути інтерпретовано як ймовірність даного результату класифікації. Потім поріг дискримінації задає розділ класів за рейтингом.

Існує кілька способів вибору порога:

Задається користувачем.

Максимальна чутливість та специфічність моделі.

Точка рівноваги — чутливість дорівнює специфічності. Мінімальна кількість помилок класифікації типу I або II; Мінімальна кількість помилок класифікації. Поріг дискримінації застосовується також в інших моделях класифікації, наприклад, у деревах рішень, де він використовується для поділу

підмножин у вузлах за заданим атрибутом, і який автоматично обчислюється на основі того, щоб зробити підмножини отримані максимально однорідними за складом класів.

У банківській практиці поріг класифікації стає стратегічним, оскільки різні типи помилок спричиняють неоднакові фінансові витрати. Відповідно до дослідження [Finance Research Letters, 2022] було виявлено, що з урахуванням вартості розгляду пропущеного дефолту або втраченого прибутку оптимальний поріг дискримінації може значно відхилятися від стандартного значення 0, 5.

Таким чином, вплив порогового значення класу виходить за рамки просто статистичної перевірки та переходить у область бізнес-вибору та уваги до ризиків.

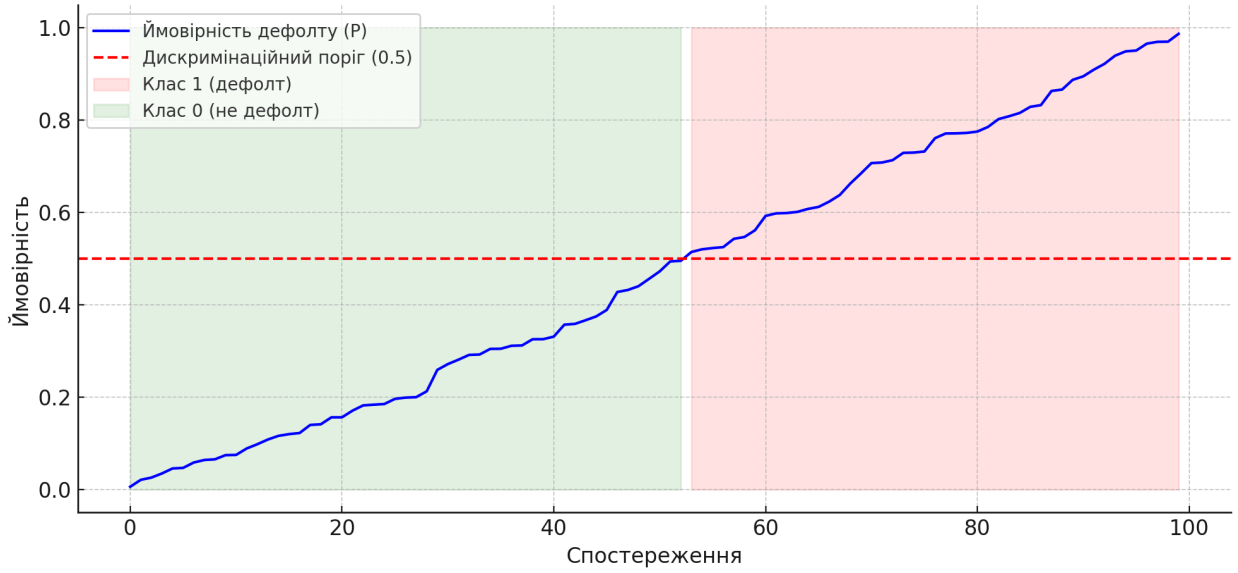


Рис. 1.3 Вплив дискримінаційного порогу на класифікацію

1.5 Роль машинного навчання у фінансовій аналітиці

Оцінка ймовірності дефолту позичальника є однією з найважливіших задач у банківському секторі, від якої безпосередньо залежить ефективність управління кредитним ризиком. У сучасних умовах ця задача дедалі частіше розв’язується за допомогою методів машинного навчання, які демонструють високу точність та здатність адаптуватися до змін у поведінці клієнтів.

Разом із цим, машинне навчання відіграє важливу роль і в інших аспектах фінансової аналітики, де класичні методи вже не забезпечують необхідної гнучкості та швидкості обробки даних. Одним з найбільш практичних напрямів застосування машинного навчання у фінансовій аналітиці є оцінка ймовірності дефолту клієнта, а саме на цьому зосереджено практичну частину даної роботи.

Проте значення цієї оцінки не обмежується лише рішенням про видачу кредиту: ймовірність дефолту (PD) є складовою більш складної системи управління ризиками, зокрема у формулі очікуваних збитків (Expected Loss), яка дозволяє банку аналізувати вразливість портфеля, прогнозувати резерви та дотримуватись регуляторних вимог.

Нижче розглянуто роль машинного навчання у таких задачах на рівні всього кредитного портфеля.

Перше що треба розглянути це очікуваний збиток (Expected Loss) — це середній розмір втрат, яких може зазнати кредитор у разі дефолту позичальників. Це ключова фінансова метрика, яка використовується у портфельному аналізі ризиків.

Очікуваний збиток обчислюється за даною формулою:

$$EL=PD*LGD*EAD \tag{1.2}$$

де:

PD (Probability of Default) — ймовірність дефолту,

LGD (Loss Given Default) — частка збитків у разі дефолту,

EAD (Exposure at Default) — сумма заборгованості на момент дефолту.

Автор статті порівнює обчислення EL із базовими правилами середнього значення в ймовірностях, знайомими зі школи. Ідея проста: якщо у вас є ймовірність події (наприклад, 5% дефолтів), і ви знаєте, скільки витрачаєте кожного разу (наприклад, 80% від $10, 000), тоді очікуваний збиток — це середнє значення цих втрат на великій вибірці клієнтів [Plain English AI, 2021].

Однак розрахунок очікуваних збитків (Expected Loss) відображає лише середньостатистичний рівень ризику. У реальних умовах фінансові системи піддаються впливу непередбачуваних та стресових факторів, таких як економічна рецесія, девальвація валюти або різке зростання ставок.

У цьому контексті важливу роль відіграє стрес-тестування портфеля — метод, що моделює поведінку ризиків під впливом негативних сценаріїв розвитку економіки. Як зазначено у роботі Медведєвої І. Б., процедура стрес-тестування передбачає визначення критичних макропараметрів (ВВП, рівень інфляції, ставка НБУ тощо), побудову регресійних залежностей і перевірку сценаріїв (рецесійного, кризового) на основі змін у цих показниках. Такий підхід дозволяє оцінити, наскільки змінюється ризик дефолту та потреба у формуванні резервів за умов різких коливань економіки [Медведєва, 2017].

Якщо стрес-тестування дозволяє оцінити вразливість конкретного банку, то аналіз системного ризику охоплює всю банківську систему країни. У роботі Покатаєвої О. В. та Славкіної М. А. запропоновано побудову інтегрального показника системного ризику, що базується на таких чинниках, як частка проблемних кредитів, співвідношення депозитів і кредитів, курс національної валюти, ВВП на душу населення та ін. Динаміка цього показника моделюється поліноміальними рівняннями, що дозволяє передбачити зростання загроз для фінансової стабільності на ранніх етапах [Покатаєва, Славкіна, 2019].

Таким чином, застосування машинного навчання в аналізі ризиків дозволяє не лише оцінювати ймовірність дефолту окремого позичальника, але й побудувати прогнозні моделі для всього портфеля та системи в цілому.

Окрім оцінки дефолтів та управління ризиками портфеля, машинне навчання активно використовується у фінансах у ширшому контексті. Воно

забезпечує автоматизацію процесів, підвищує точність прогнозів і дозволяє банкам приймати рішення в реальному часі. До ключових напрямів належить виявлення шахрайських транзакцій, сегментація клієнтів і рекомендаційні системи.

У сфері виявлення фінансового шахрайства (fraud detection) ML-моделі аналізують транзакції на наявність аномалій та нетипової поведінки. Як зазначає Безруков С. В. [Безруков, 2023], ефективні результати демонструють моделі виявлення відхилень, здатні самостійно адаптуватися до змін без участі оператора, блокуючи потенційно небезпечні дії.

Сегментація клієнтів здійснюється за допомогою методів кластеризації (наприклад, K-means), що дозволяє формувати цільові пропозиції для кожної групи користувачів. Як описано в дослідженні Ткачик О. О. [Ткачик, с. 13–15], кластеризація за поведінковими та фінансовими ознаками дає змогу підвищити ефективність маркетингу та рівень утримання клієнтів.

Також застосовуються рекомендаційні системи, які прогнозують інтерес клієнтів до певних послуг банку. У роботі Кравченко О. В. [Kravchenko, 2022] продемонстровано використання нейромережевої моделі, що формує індивідуальні пропозиції на основі історичних та поведінкових даних.

Таким чином, машинне навчання у фінансовому секторі виходить за межі скорингу, відіграючи значну роль у виявленні ризиків, персоналізації послуг та покращенні клієнтського досвіду

2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Опис предметної області та специфіки кредитного скорингу

Кредитування є одним із ключових механізмів, що забезпечує розвиток економіки, підтримку підприємництва, стабільність банківської системи та інвестиційну привабливість держави. За даними Стратегії розвитку кредитування, затвердженої Радою з фінансової стабільності, саме кредитна підтримка є основою відновлення після кризової економіки, зокрема в умовах воєнного стану [Lending Development Strategy 2024, с. 2–5] Банківське кредитування охоплює як сектор бізнесу, так і населення, і має стратегічне значення для забезпечення платоспроможності, інноваційного розвитку та зайнятості.

У структурі банківських активів кредити є основним джерелом прибутку: вони формують до 70% активів і до 2/3 доходів банків. Проте водночас вони є джерелом найвищого фінансового ризику, а саме — кредитного ризику, що виникає у випадку невиконання позичальником своїх зобов’язань. Кредитний ризик посідає центральне місце серед банківських ризиків, становлячи до 50% потенційних втрат [Боронос і Влізько, 2010, с. 183]. Його зростання прямо загрожує стабільності банківської системи: збільшується частка проблемних кредитів, необхідність формування резервів, а отже, знижується прибутковість.

Особливістю сучасної банківської сфери є її масовість та потреба в автоматизації. Через велику кількість кредитних заявок та потребу в об'єктивному ухваленні рішень банки дедалі більше застосовують скорингові системи — автоматизовані моделі оцінки ймовірності дефолту клієнта. Скоринг дозволяє уніфікувати процес оцінки, зменшити суб’єктивний вплив людського фактора, підвищити швидкість обробки заявок та зменшити частку безпідставних відмов.

Разом з тим впровадження скорингових систем має свої специфічні особливості, пов’язані з високою ціною помилки та потребою в точних і репрезентативних даних. Як зазначають Брітченко І. Г. та Момот О. М., серед ключових проблем застосування скорингу є:

- складність налаштування моделей;
- нестача якісних даних;
- потреба в інтеграції з іншими банківськими системами;
- забезпечення прозорості та інтерпретованості моделей у відповідності до вимог НБУ і європейських стандартів [Брітченко, Момот, 2011, с. 32–36]

Крім того, в українських банках найчастіше використовуються моделі application-скорингу (оцінка при подачі заявки) та collection-скорингу (виявлення ризиків у роботі з простроченою заборгованістю), тоді як поведінкові (behavioral) або шахрайські (fraud) скорингові моделі застосовуються рідше, попри їхню ефективність у розвинутих країнах.

У контексті кредитного скорингу також важливо враховувати різноманітність клієнтів, для яких розробляються кредитні продукти, а також особливості самих фінансових інструментів, що впливають на структуру моделі оцінки ризику. На практиці клієнти банків класифікуються за кількома критеріями: за правовим статусом (фізичні особи, фізичні особи-підприємці, юридичні особи), за рівнем доходу, типом зайнятості, а також за призначенням кредиту (споживчі, іпотечні, бізнес-кредити тощо).

Згідно з класифікацією, наведеною у праці Синициної Г. А. та Рачкован О. Д. [Синицина, Рачкован, 2020, с. 49–52], клієнтами фінансових установ можуть виступати не лише приватні особи, а й суб'єкти малого та середнього бізнесу, великі корпоративні клієнти, а також державні установи у рамках державно-приватного партнерства. Для кожного з цих сегментів характерні різні підходи до оцінювання кредитоспроможності, що зумовлено обсягом кредиту, строками повернення, типом застави та ступенем регуляторної відповідальності.

Щодо самих кредитних продуктів, банки пропонують широкий спектр рішень: від стандартних споживчих кредитів на побутові потреби до іпотечних кредитів, автокредитів, овердрафтів, кредитних ліній для бізнесу, а також фінансування проектів і кредитування інвестиційної діяльності. Кожен із цих продуктів вимагає від скорингової системи адаптації — наприклад, моделі для

короткострокових кредитів можуть використовувати інші змінні, ніж моделі для довгострокових інвестицій.

Таким чином, для ефективного застосування скорингових технологій у банківській практиці важливо враховувати не лише характеристики клієнта, але й тип кредитного продукту, адже різні продукти мають різну структуру ризику, частоту дефолтів і вимоги до перевірки.

Окрім врахування різних типів клієнтів і продуктів, побудова ефективної скорингової моделі передбачає глибоке розуміння обмежень та складностей самої предметної області. Оцінка кредитного ризику, незважаючи на активний розвиток методик, залишається однією з найскладніших задач у фінансовій сфері. Як зазначає Андрушків Т. (2008), проблематика оцінки кредитоспроможності позичальників є ключовою у процесі управління ризиками в банківській діяльності. В умовах ринку, що характеризується нестабільністю, політичною невизначеністю та економічною кризовістю, кредитний ризик має не лише фінансовий, а й стратегічний характер.

Однією з головних складностей є те, що поведінка позичальників має вірогіднісний характер і не піддається точному прогнозуванню. Це означає, що навіть за наявності якісної інформації про клієнта, гарантувати відсутність дефолту практично неможливо. Банки змушені обирати компроміс між прибутковістю та ризиком, прагнучи водночас мінімізувати можливі втрати і підтримувати активну кредитну політику.

Серед основних системних проблем у предметній області кредитного скорингу можна виділити: суб’єктивність прийняття рішень у випадку відсутності автоматизованих моделей; відсутність єдиних стандартів і нормативів для оцінки кредитоспроможності; недосконалість існуючих методик — надмірна емпіричність, низький рівень математичного обґрунтування, значна залежність від людського фактору; брак якісної статистичної бази, відсутність централізованих джерел даних про історію позичальників (наприклад, дієві кредитні бюро); відсутність спадкоємності експертних рішень, оскільки знання неможливо формалізувати та передати без втрат; наявність неформалізованих факторів, таких

як репутація, моральна надійність або корпоративна культура клієнта, які важко оцифрувати.

Також дослідник звертає увагу на розрив між вітчизняною та зарубіжною практикою: іноземні банки застосовують складні рейтингові системи, тоді як в Україні оцінка кредитоспроможності часто базується на внутрішніх, емпіричних методах, не підтриманих належною статистикою або алгоритмічною структурою.

Зрештою, будь-яка помилка в оцінці кредитоспроможності тягне за собою погіршення якості портфеля, збільшення резервів, зниження прибутковості, а в гірших випадках — втрату платоспроможності самого банку. Тому вдосконалення методів оцінки кредитного ризику є не просто актуальним завданням, а необхідною умовою підвищення надійності банківської системи загалом.

2.2 Огляд джерел даних і типових ознак клієнтів

Для побудови ефективної скорингової моделі банківська система повинна спиратися на широкий спектр даних, що відображають фінансову поведінку та соціально-демографічні характеристики клієнта. Згідно з дослідженням Мовчанюка О. А. (2008), основними джерелами інформації для оцінки кредитного ризику є:

- анкета клієнта (вік, сімейний стан, професія, рівень доходу, освіта тощо);
- документально підтверджені довідки (паспорт, ідентифікаційний код, довідка про доходи, документи на власність);
- дані кредитного бюро (кредитна історія, наявність прострочень, активні зобов’язання);
- інформація з поточних рахунків (рух коштів, залишки, активність);
- поведінкові сигнали (наприклад, запити до банку, спосіб заповнення анкети тощо) [Мовчанюк, 2008, с. 405–406].

Особливої уваги заслуговує наявність структурованих і уніфікованих кредитних історій, які в українській практиці поки що залишаються обмеженим інструментом через недостатню автоматизацію обміну між банками та

кредитними бюро. Саме це, на думку автора, суттєво ускладнює повну реалізацію потенціалу скорингових моделей на рівні національного ринку.

Також варто зазначити, що ефективність побудови моделі прямо залежить від доступності історичних даних про «хороших» і «поганих» клієнтів, тобто тих, що повернули кредит вчасно або допустили прострочення. У багатьох українських банках частина таких даних досі зберігається на паперових носіях, що створює технічні бар'єри для розробки точних моделей.

Мовчанюк зазначає, що, хоча джерел може бути багато, ефективна скорингова модель не повинна містити надмірної кількості змінних. Існує «золоте правило»: оптимально використовувати від 8 до 15 ознак, найбільш релевантних до ймовірності дефолту [Мовчанюк, 2008, с. 407].

Отримавши базу даних із внутрішніх та зовнішніх джерел, наступним кроком є формування системи ознак, які будуть використовуватися в моделі для прогнозування ризику. Як зазначає Соломоник О. П. (2025), у банківській сфері особливо важливе значення мають релевантність ознак, їх інтерпретованість, а також стійкість до змін у даних. Типові ознаки поділяються на кілька ключових груп.

До соціально-демографічних ознак належать вік клієнта, стать, регіон проживання, сімейний стан, наявність дітей, рівень освіти, а також форма працевлаштування. Ці змінні дозволяють створити базовий профіль клієнта і порівняти його з типовими позичальниками, які повернули або не повернули кредит.

Фінансові ознаки є критично важливими у скорингу. До них належать розмір щомісячного доходу, сума всіх активних кредитів, рівень боргового навантаження (DTI), частка доходу, що витрачається на обслуговування зобов'язань (PTI), наявність депозитів, відкритих рахунків, банківських карток тощо. Ці параметри використовуються для аналізу платоспроможності позичальника.

Кредитна історія охоплює інформацію про кількість раніше отриманих кредитів, строки та суми, кількість прострочених платежів, тривалість кредитної історії, час від останнього дефолту або успішного завершення кредитного договору.

Також у сучасних моделях застосовуються поведінкові та цифрові ознаки, зокрема: частота використання мобільного банкінгу, кількість логінів у систему, історія взаємодії з колл-центром, використання банкоматів, а також час заповнення заявки. Такі ознаки демонструють активність клієнта, стабільність його поведінки та іноді є сильним індикатором ризику дефолту.

У своїй роботі Соломоник підкреслює важливість інженерії ознак (feature engineering): створення похідних змінних (наприклад, відношення суми кредиту до доходу, кількість транзакцій у нестандартний час), нормалізація числових значень, та кодування категоріальних даних. Ці операції суттєво покращують точність моделі, особливо при застосуванні алгоритмів ансамблевого навчання, таких як Random Forest або CatBoost (Соломоник, 2025, с. 45–49).

Таким чином, правильний вибір та обробка ознак є ключовими факторами успіху скорингової моделі. Вони визначають її здатність адаптуватися до нових клієнтів і ринкових умов, зберігаючи при цьому високу точність прогнозування.

У рамках процесу побудови ефективної скорингової моделі важливим етапом є інженерія ознак (feature engineering) — створення нових змінних або трансформація наявних для підвищення точності моделі. За даними дослідження Cognizant (Codex 4971), саме інженерія ознак є найбільш трудомістким і критичним етапом у життєвому циклі моделі машинного навчання, а не вибір алгоритму чи налаштування гіпер-параметрів. Особливої уваги набуває автоматизація цього процесу та створення централізованих «сховищ ознак» (feature stores), що зберігають перевірені, версіоновані й повторно використовувані змінні, застосовні у різних моделях та сценаріях.

Автори зазначають, що ознаки — це не лише змінні з бази даних, але й агреговані, обчислювані або контекстуальні значення, такі як: «кількість транзакцій за останні 7 днів», «середній платіж за місяць», «кількість входів у

мобільний застосунок». Вони створюються за допомогою математичних перетворень, групувань, логічних або часових фільтрів. Часто ці ознаки повторно створюються в різних моделях, що веде до дублювання зусиль і помилок. Щоб уникнути цього, пропонується зберігати ознаки у спільному фреймворку, де інженери та аналітики можуть спільно керувати їх життєвим циклом — від створення до вилучення з обігу (Cognizant, 2019).

Окрему увагу приділено технологіям автоматизації — таким як deep feature synthesis, яка дозволяє будувати похідні ознаки на основі реляційних даних автоматично. Також описані практики оцінки важливості ознак, відстеження їхнього зв’язку з моделями, захист чутливої інформації, контроль версій та усунення дублювань.

Таким чином, згідно з результатами дослідження, якість інженерії ознак має вирішальний вплив на успішність машинного навчання у фінансовій сфері. Автоматизовані інструменти та централізовані сховища ознак дозволяють скоротити час.

Таким чином, ефективність скорингової моделі безпосередньо залежить від якості джерел даних і обґрунтованості вибраних ознак. Використання як внутрішньої, так і зовнішньої інформації, зокрема кредитної історії, фінансових показників, соціально-демографічних і поведінкових характеристик клієнта, дозволяє моделі краще відображати кредитоспроможність позичальника. Водночас належна інженерія ознак — процес їх трансформації та оптимізації — значно підвищує точність прогнозів, знижує ризик помилок і забезпечує адаптивність системи до нових даних і умов. Це формує основу для подальшого етапу — побудови ефективної моделі бінарної класифікації.

2.3 Проблема дисбалансу класів у даних

У процесі побудови скорингових моделей, які здійснюють автоматизовану оцінку ймовірності дефолту клієнта, одним з найсерйозніших викликів є дисбаланс класів у навчальних даних. Це явище полягає в тому, що один з класів цільової змінної значно переважає інший. У більшості реальних фінансових задач

кількість клієнтів, які справно обслуговують свої зобов'язання, у десятки разів перевищує кількість тих, хто допустив прострочення або повний дефолт. Наприклад, за статистикою комерційних банків, середній рівень прострочення споживчих кредитів коливається в межах 7–10 %, а в іпотечному кредитуванні може бути ще нижчим. Це означає, що в типових датасетах лише кожен десятий або двадцятий приклад належить до класу «дефолт», тоді як решта — до класу «надійний клієнт». Такий дисбаланс перетворює задачу класифікації на задачу розпізнавання рідкісних подій, що принципово ускладнює моделювання.

Проблема полягає в тому, що більшість алгоритмів машинного навчання оптимізуються за загальними метриками, як-от точність (accuracy), яка при наявному дисбалансі дає хибне враження про ефективність моделі. Наприклад, якщо модель передбачатиме, що всі клієнти «надійні», вона отримає точність понад 90 %, однак при цьому не виявить жодного дефолтного позичальника, а це й є головна мета скорингу. Галущак М. Ю. у своїй роботі вказує, що подібний ефект спостерігався і при класифікації студентської успішності, де модель відтворювала більшість, ігноруючи менш поширені, але критично важливі категорії [Галущак, 2024, с. 13–14]. У контексті кредитного скорингу така модель є абсолютно неефективною, оскільки головна задача банку — це не просто підтвердження неблагонадійних заявників, а виявлення ризикових.

У зв'язку з цим стає необхідним орієнтування на альтернативні метрики, які краще враховують важливість класу меншості. До них належать Recall — частка правильно передбачених дефолтних клієнтів серед усіх реальних дефолтів; Precision — частка справжніх дефолтів серед усіх передбачених; F1-score — гармонічне середнє між цими двома показниками; а також ROC-AUC і Precision-Recall крива, що дають інтегральне уявлення про якість класифікації на різних порогах. Як наголошує Галущак, оцінювання моделі має проводитися саме за цими показниками, особливо якщо вартість помилки класифікації є асиметричною, як у випадку банківського дефолту [Галущак, 2024, с. 14].

Для подолання наслідків дисбалансу класів розроблено низку технічних підходів. У роботі Галущака перелічено основні з них: oversampling — штучне

розмноження прикладів класу меншості за допомогою дублювання або генерації нових (наприклад, метод SMOTE); undersampling — видалення частини прикладів класу більшості для вирівнювання пропорцій; налаштування ваг класів — зміщення функції втрат так, щоб модель «більше боялася» помилок на меншості; cost-sensitive learning — явне врахування вартості різних типів помилок у навчанні моделі. Застосування цих підходів дозволяє зробити модель більш чутливою до важливих, хоча й рідкісних випадків, і зменшити ймовірність пропущених дефолтів, які є фінансово найбільш ризикованими.

У межах практичної реалізації даної роботи було проаналізовано навчальну вибірку, яка мала 700 позитивних прикладів (надійні клієнти) та 300 негативних (дефолтні). Для вирівнювання структури класів спочатку використовувався підхід class_weight='balanced' у моделі Random Forest, що дозволяє автоматично компенсувати диспропорцію. Після цього було застосовано метод SMOTE, що створив ще 400 синтетичних прикладів класу 0 та забезпечує повний баланс: 700: 700. Проте після навчання моделі стало очевидно, що застосування SMOTE не дало суттєвого покращення якості прогнозу. Це пояснюється кількома факторами: по-перше, початковий дисбаланс не був критичним (лише 70/30), по-друге, модель уже використовувала компенсацію ваг, а по-третє, нові синтетичні дані SMOTE були надто схожими на наявні й не дали додаткової варіації. Подібну ситуацію підтверджують і Фостяк М. та Демків Л., які вказують, що класичні техніки балансування ефективніші лише у поєднанні з глибокою інженерією ознак або використанням генеративних моделей [Фостяк, Демків, 2024, с. 30–31].

Таким чином, проблема дисбалансу класів залишається одним із ключових факторів, які впливають на здатність скорингової моделі виявляти ризикованих клієнтів. Прості техніки, як-от SMOTE або вагові коефіцієнти, не завжди гарантують покращення. Для підвищення точності в умовах навіть помірного дисбалансу доцільно застосовувати поєднання підходів: балансування, інженерію ознак, налаштування порогу класифікації, а також подальше сегментування клієнтів для ручного перегляду. Ігнорування дисбалансу в задачах фінансового прогнозування може призвести до заниженого рівня Recall для дефолтного класу,

що своєю чергою загрожує банку недоотриманням прибутку та прямими кредитними втратами.

Підсумовуючи, слід зазначити, що дисбаланс класів є не лише технічною проблемою, а і стратегічним викликом для фінансових установ, які використовують машинне навчання для прогнозування кредитних ризиків. Галущак М. Ю. у своїй роботі підкреслює, що успішна адаптація моделей до нерівномірного розподілу цільової змінної можлива лише за умови комплексного підходу, який враховує тип даних, структуру вибірки та бізнес-логіку застосування [Галущак, 2024]. Використання лише однієї техніки, наприклад, oversampling або вагових коефіцієнтів, у багатьох випадках не дає суттєвого ефекту. Саме тому важливо комбінувати декілька підходів, включаючи генерацію нових ознак, зміщення порогу класифікації, побудову спеціалізованих loss-функцій, або навіть використання окремих моделей для різних сегментів клієнтів. Досвід Фостяка і Демківа також демонструє, що балансування даних має сенс лише в контексті ширшої архітектури управління даними, зокрема, через впровадження Data Mesh-підходів та ізольованих ML-доменів [Фостяк, Демків, 2024]. Таким чином, вирішення проблеми дисбалансу потребує не лише технічних рішень, а й глибокого розуміння природи даних і контексту, в якому працює модель. Актуальність цього питання зростає в умовах автоматизації банківських процесів, де ухвалення кожного рішення має фінансові наслідки. Своєчасне виявлення слабких місць у розподілі вибірки дозволяє підвищити надійність систем кредитного скорингу та зменшити ризик помилкових рішень щодо клієнтів. Отже, інтелектуальне управління балансом даних має стати постійною частиною життєвого циклу будь-якої скорингової моделі.

2.4 Формалізація задачі: постановка як задачі бінарної класифікації

Задача прогнозування ймовірності дефолту клієнта в рамках системи оцінки кредитного ризику формалізується як задача бінарної класифікації. Метою є визначення, чи зможе позичальник повернути кредит, чи ні. У цьому випадку

кожен клієнт позначається міткою класу: 1 — «надійний» клієнт, 0 — клієнт з потенційним дефолтом.

У рамках машинного навчання це означає побудову функції класифікації, яка на основі вхідного вектору ознак (вік, дохід, тривалість позики, кількість наявних кредитів тощо) повертає клас об'єкта. Інакше кажучи, для кожного прикладу у вибірці модель повинна спрогнозувати один з двох варіантів результату: дефолт або відсутність дефолту.

У рамках практичної частини даного дослідження завдання було реалізоване на основі датасету German Credit Data, де кожен приклад містив фінансові та соціальні характеристики клієнта, а цільовою змінною була змінна class, що приймає значення 1 (благонадійний клієнт) або 0 (дефолтний). Особливістю реалізації стало використання кластеризації методом KMeans як допоміжного етапу попередньої обробки ознак. Створена змінна cluster id була додана до вхідного простору ознак, що дозволило враховувати приховану подібність між клієнтами.

Використання кластеризації дозволило покращити здатність моделі до сегментації клієнтів перед класифікацією, що є прикладом поєднання методів контрольованого та неконтрольованого навчання. Це підтверджується також візуалізацією результатів через метод зниження розмірності PCA, яка продемонструвала чітке розділення кластерів на площині.

У ході реалізації порівнювались три моделі машинного навчання: Logistic Regression, Random Forest Classifier та XGBoost Classifier. Для підвищення чутливості моделі до класу дефолту в Random Forest був доданий параметр class weight='balanced', а також реалізовано балансування вибірки за допомогою SMOTE, яке вирівняло кількість прикладів кожного класу. Це дозволило усунути вплив диспропорції, що часто виникає в задачах кредитного скорингу, де клієнти, які виконують зобов’язання, значно переважають за кількістю тих, хто допустив дефолт.

Моделі класифікували клієнтів на основі широкого спектру ознак. Застосовувалась розширена інженерія ознак: побудова нових параметрів на основі

відношень, логарифмів, оцінок стабільності тощо. Після навчання моделі повертали не тільки клас (0 або 1), але й імовірність належності до дефолтної групи. Це дозволяє банку працювати з гнучкими порогами класифікації, наприклад, встановлювати поріг у 0. 6 або 0. 7 замість класичних 0. 5 залежно від політики ризик-менеджменту.

Крім точності моделей, окрему увагу було приділено інтерпретації результатів — наскільки чітко модель пояснює, чому саме класифікує клієнта як дефолтного чи благонадійного. Хоча деякі алгоритми (як-от XGBoost) вважаються «чорними скриньками», за допомогою таких методів як SHAP-аналіз або відображення важливості ознак можна визначити, які характеристики впливають найбільше на прогноз. Це критично важливо у фінансовій сфері, де рішення повинні бути не лише точними, а й пояснюваними.

У процесі порівняння моделей було відзначено, що лінійна модель логістичної регресії, хоча й проста в реалізації та інтерпретації, показує нижчу гнучкість у випадках складної взаємодії між ознаками. Натомість ансамблеві дерева (Random Forest, XGBoost) здатні враховувати нелінійні взаємозв’язки, що є типовими для кредитних даних.

Особливу увагу в дослідженні було приділено метриці Recall як ключовому критерію ефективності моделей. Пояснюється це тим, що помилка II типу (false negative) — тобто ситуація, коли дефолтний клієнт помилково вважається благонадійним — несе найбільший фінансовий ризик. Відтак максимізація Recall для дефолтного класу має вищий пріоритет, ніж загальна точність або Precision.

Також важливим етапом було масштабування ознак, особливо перед застосуванням моделей, чутливих до відстаней (наприклад, кластеризації або PCA). Було застосовано MinMaxScaler для приведення всіх числових змінних до одного діапазону. Це дозволило уникнути домінування окремих ознак та покращити якість формування кластерів.

Важливо, що імовірнісний вихід моделі використовувався не лише для класифікації, а й для рейтингування клієнтів — формування списку від найбільш ризикованих до найменш ризикованих. Такий підхід дозволяє банку не лише

приймати рішення «дати/не дати кредит», а й визначати умови: наприклад, встановлення підвищеної ставки або необхідність надання застави.

Окрему увагу було приділено побудові графіків ROC-кривої, Precision-Recall кривої та візуалізації кластерів у дво — та тривимірному просторі. Це дозволило наочно продемонструвати, як класи розподілені у вибірці та як ефективно працює класифікатор. Особливо цінною виявилася візуалізація перед і після балансування класів.

У рамках документаційної частини моделі, побудованої для цієї роботи, було використано бібліотеки scikit-learn, xgboost, pandas, matplotlib, seaborn, а також imblearn для реалізації SMOTE. Це забезпечило високу reproducibility експериментів і можливість подальшого доопрацювання моделі.

Застосування кластеризації перед класифікацією було описано також у тезах, представлених автором на студентській конференції ДУТ (секція №5). У тезах підкреслювалась роль поєднання алгоритмів контрольованого і неконтрольованого навчання як етапу покращення інформативності ознак. Це підтверджує міжрозділову цілісність структури дипломного дослідження.

Загалом формалізація задачі як бінарної класифікації з додатковим імовірнісним виходом, гнучкими порогами та підтримкою рішень на рівні кластерів, дозволяє реалізувати на практиці реальну систему оцінки кредитоспроможності. Такий підхід відповідає сучасним вимогам фінансового ринку та може бути адаптований до будь-якої банківської IT-архітектури.

2.5 Оцінка ризиків прийняття рішень та вплив помилок класифікації

У процесі побудови скорингових моделей важливим аспектом є не лише точність прогнозування, а й аналіз можливих наслідків помилок моделі в реальному банківському середовищі. Помилки класифікації умовно поділяються на два типи: помилки першого роду (false positives) та другого роду (false negatives), і кожна з них має різні фінансові та репутаційні наслідки.

У контексті кредитного скорингу помилка першого роду (класифікація благонадійного клієнта як ризикового) може призвести до втрати потенційного

прибутку, оскільки банк відмовляє у кредиті особі, яка могла б повернути кошти вчасно. Натомість помилка другого роду (класифікація дефолтного клієнта як надійного) є значно небезпечнішою: вона веде до безпосередніх фінансових збитків через невиконання зобов’язань. Тому на практиці саме мінімізація ймовірності помилок другого роду (FN) є пріоритетною метою.

Як зазначено у дослідженні Plain English AI (2023), очікуваний збиток (Expected Loss) розраховується як добуток ймовірності дефолту (PD), експозиції на момент дефолту (EAD) та коефіцієнта втрати при дефолті (LGD). Саме цей показник є основою для побудови систем ризик-менеджменту та кредитної політики. Пропущений дефолт напряму впливає на зростання цього показника.

На етапі практичної реалізації були помітні ситуації, коли модель повертала високу загальну точність, але при цьому Recall для дефолтного класу залишався низьким. Така модель є неприйнятною, оскільки вона залишає ризик «непомічених» проблемних клієнтів. Саме тому в моделі Random Forest було використано параметр class_weight='balanced', що дозволяє компенсувати диспропорцію між класами. А також було застосовано SMOTE, який вирівнює кількість прикладів у кожному класі, аби підвищити чутливість моделі до рідкісного класу.

Оцінка ймовірності дефолту клієнта дозволяє не лише приймати рішення щодо надання кредиту, а й адаптувати умови: підвищити ставку, вимагати заставу або зменшити ліміт. Таким чином, помилки класифікації можуть не тільки впливати на рішення «так/ні», а й на всю економічну стратегію щодо клієнта.

Фостяк і Демків (2024) підкреслюють, що сучасні підходи до оцінювання мають ґрунтуватися не на фіксованому пороговому значенні 0. 5, а на гнучкому механізмі адаптації порогу залежно від цілей установи. У межах практичної роботи було досліджено, як зміна порогу класифікації (наприклад, з 0. 5 до 0. 6) впливає на метрики Precision та Recall. Такий підхід дозволяє більш тонко налаштувати модель до поточних ринкових умов.

Також важливою є оцінка впливу помилок на кредитний портфель у цілому. Кілька хибних рішень, допущених моделлю, можуть акумулюватися у значні

збитки на рівні банку, особливо при великій кількості виданих кредитів. Саме тому багато банків впроваджують додаткові рівні перевірки — ручний перегляд «сірої зони» клієнтів з ймовірністю дефолту близькою до порогу.

Окремим аспектом є репутаційні та правові ризики, які виникають у разі, якщо модель дискримінує певні категорії позичальників — за віком, статтю чи іншим чутливим параметром. Галущак (2024) у своїй роботі наголошує, що некоректна модель може створити систематичну упередженість у прийнятті рішень, що є порушенням принципів етики та законодавства в сфері захисту прав споживачів.

З технічної точки зору, помилки класифікації можна контролювати за допомогою аналізу матриці помилок (confusion matrix), а також побудови ROC-кривої та Precision-Recall-кривої. У дипломній роботі такі графіки використовувались для візуального аналізу, і вони чітко показали компроміс між чутливістю та специфічністю при зміні порогу.

Застосування метрик типу F1-score, а також інструментів інтерпретації (наприклад, SHAP-значень) дозволяє глибше зрозуміти, в яких випадках модель найбільше помиляється і які ознаки найбільше впливають на результат. Це відкриває можливість для подальшого доопрацювання моделі та оптимізації політик ризик-контролю.

Вплив помилок класифікації не обмежується лише індивідуальними кредитними кейсами. У разі масштабного використання моделей машинного навчання в автоматизованих рішеннях вони стають частиною кредитної політики установи, що створює додаткові стратегічні ризики. Наприклад, постійне недооцінювання ризику внаслідок високого порогу або неправильної метрики може призвести до поступового накопичення проблемної заборгованості.

Також важливо враховувати динамічний характер даних: профілі клієнтів, ринки та поведінка змінюються, тому помилки, які спочатку були одиничними, можуть з часом стати систематичними. Це робить необхідною регулярну переоцінку точності моделі, валідацію на нових вибірках і періодичне оновлення ваг або структури моделі. Один із сучасних підходів — використання стратегій

time-based validation, де оцінювання здійснюється на майбутніх, а не випадкових підвибірках.

Ще одним критичним моментом є помилки класифікації в умовах регіональних або соціальних відмінностей, коли одна й та сама модель демонструє різну ефективність у різних сегментах ринку. Це може призвести до так званого bias leakage — ситуації, коли похибки не розподіляються випадково, а зосереджуються в певних демографічних групах. Подібні ризики вже зафіксовані в дослідженнях систем кредитного скорингу в Європі та США, що призвело до необхідності регуляторного аудиту моделей.

На рівні інтерфейсу скорингової системи також існують важливі технічні аспекти. Наприклад, для мінімізації операційного ризику рекомендується впровадження порогів із вбудованою зоною невизначеності: при ймовірностях від 0. 45 до 0. 55 рішення направляється на ручний перегляд. Це дозволяє знизити навантаження на моделі в пограничних ситуаціях і запобігти критичним помилкам.

Окремо варто виділити вплив помилок на репутацію установи. У світі, де дані і цифрові рішення все частіше замінюють людське втручання, фінансові компанії змушені доводити, що їхні алгоритми є об’єктивними, справедливими та пояснюваними. Клієнти, які отримали відмову, дедалі частіше мають право вимагати пояснень згідно із законодавством про захист персональних даних.

Ще один ризик — неадекватне реагування на кризові ситуації. Під час економічної нестабільності або пандемій, патерни поведінки клієнтів змінюються, і моделі, які раніше працювали добре, можуть раптово почати генерувати великі обсяги помилок. Це підтверджується кейсами банків в ЄС під час пандемії COVID-19, коли довелося тимчасово вимкнути моделі та перейти на ручне оцінювання ризиків.

Крім класичних фінансових наслідків, варто враховувати і довгостроковий вплив на аналітичну інфраструктуру. Якщо моделі систематично помиляються, це призводить до викривлення зворотного зв’язку, що унеможливлює ефективне

перенавчання і покращення майбутніх моделей. Таким чином, одна серйозна серія помилок створює «ефект доміно» в усьому кредитному циклі.

Для зниження цього ризику сучасні системи кредитного скорингу використовують механізми моніторингу стабільності моделі (model drift). Якщо розподіл ймовірностей або точність на поточних даних суттєво відрізняється від навчальної вибірки, система повідомляє про це ризик-менеджера. Також ефективним є використання «champion-challenger» підходу, коли паралельно тестуються дві моделі і в реальний процес потрапляє лише та, що демонструє стабільнішу поведінку.

Загалом, управління ризиками помилок класифікації є не технічним доповненням, а фундаментальним елементом побудови системи підтримки рішень у сфері кредитування. Воно охоплює фінансові, репутаційні, правові й стратегічні виміри ефективності роботи моделі й напряму впливає на стійкість фінансової установи.

Таким чином, управління помилками класифікації не є другорядним питанням, а навпаки — однією з основних складових побудови якісної скорингової моделі. Розуміння природи цих помилок, їхня кількісна оцінка та стратегія пом’якшення наслідків безпосередньо визначають ефективність застосування машинного навчання у реальних банківських процесах.

2.6 Визначення цілей та критеріїв якості моделі

Головна мета побудови моделі машинного навчання для задачі кредитного скорингу є забезпечення максимально точної та стабільної класифікації клієнтів за рівнем кредитного ризику. Це означає здатність моделі виявляти дефолтних позичальників до моменту надання їм кредитного продукту. Досягнення цієї мети дає смогу банківській установі мінімізувати фінансові втрати, які пов’язані з невиконанням кредитних зобов’язань, а також ефективно розподіляти ресурси між клієнтами.

При оцінці ефективності моделі застосовуються кілька ключових метрик. Більше всього розглядається Recall (чутливість) — здатність моделі виявляти

клієнтів, які насправді є дефолтними. Саме ця метрика має першочергове значення у фінансовій сфері, оскільки пропущений дефолт призводить до найбільших збитків. Також враховуються Precision (точність), яка показує, яка частка передбачених дефолтників є справжніми, та F1-score — баланс між Recall і Precision.

Додатково аналізується ROC-AUC, яка демонструє загальну здатність моделі відокремлювати класи незалежно від вибраного порогу. У випадках із дисбалансом класів важливим є також PR-AUC, яка краще відображає поведінку моделі щодо рідкісного класу.

Цілі моделі також включають забезпечення інтерпретованості, стійкості до нових даних, а також можливість адаптації до змін умов ринку. Крім цього, модель має бути оптимізована за швидкістю обробки запитів та масштабованістю, з огляду на подальше впровадження в реальні банківські системи.

У межах дипломної роботи акцент зроблено саме на Recall як ключову метрику, яка дозволяє забезпечити контроль ризиків. Усі моделі порівнювались за однаковими метриками, що дозволило об’єктивно визначити найкращий варіант. Крім того, оцінка проводилась із використанням кросс-валідації (реалізованої у Google Colab з використанням cross_val_score з scikit-learn), що підвищило достовірність результатів.

Іншою важливою ціллю було зниження впливу помилок другого роду, тобто ситуацій, коли дефолтні клієнти неправильно класифікуються як благонадійні. Саме такі помилки є найбільш фінансово небезпечними, а тому недопустимі при впровадженні моделі у реальну банківську практику. Також було враховано можливість зміни бізнес-умов — з цією метою були збережені треновані моделі (через joblib), що дозволяє швидко виконати перенавчання на нових даних без повторної побудови всієї системи.

Суттєвою перевагою застосування моделі стало також врахування імовірнісного виходу, який дозволяє не просто класифікувати клієнта, а ранжувати його за рівнем ризику. Це дає змогу кредитним менеджерам приймати диференційовані рішення — змінювати ставку, вимагати додаткові гарантії або

відмовляти лише при перевищенні певного порогу. Такий підхід реалізовано в моделі XGBoost Classifier, де використано predict_proba() для повернення ймовірностей замість жорсткої класифікації.

У процесі аналізу враховувався вплив зміни дискримінаційного порогу на співвідношення помилок першого та другого роду. Це дозволило запропонувати гнучку схему прийняття рішень, яка може бути адаптована під конкретні бізнес-вимоги. Додатково оцінювався вплив балансування класів (через SMOTE із бібліотеки imblearn) на стабільність метрик під час тестування. Результати підтвердили доцільність використання комплексного підходу, що поєднує класифікацію, кластеризацію (KMeans) та оцінку ризиків. Усе це забезпечує не лише технічну ефективність, а й практичну цінність розробленої моделі для реального застосування у сфері кредитного аналізу.

Також під час побудови моделі враховувалась можливість подальшого масштабування рішення для різних типів кредитних продуктів — від споживчих до іпотечних. Було протестовано, як змінюється ефективність моделей при використанні кластеризації як додаткового джерела інформації, що підтверджено в аналізі результатів у секції №5 студентської наукової конференції ДУТ. Це дозволило уточнити структуру вибірки та покращити прогнозну здатність моделі на складних сегментах клієнтів. З урахуванням практичної реалізації всі критерії якості були зафіксовані на кожному етапі експериментів у середовищі Google Colab. В результаті отримано універсальну систему, здатну адаптуватися до нових даних, бізнес-умов та специфіки кожної фінансової установи.

У підсумку, мета побудови моделі полягає не лише в підвищенні точності прогнозування, а й у створенні надійного та адаптивного інструменту підтримки прийняття рішень, який враховує всі аспекти ризику та відповідає реальним потребам банківської системи.

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ РІШЕНЬ

3.1 Підготовка даних та формування ознак

Побудова системи підтримки рішень для оцінки кредитного ризику потребує ретельного етапу підготовки даних. Як зазначено у [Брітченко, Момот, с. 34], основою ефективного скорингу є якісний та репрезентативний набір ознак, що дозволяє максимально точно передбачити ймовірність дефолту. У нашому випадку було використано датасет у форматі. arff, який містив 1000 записів кредитних заявок клієнтів банку Німеччини з атрибутами на кшталт віку, суми кредиту, статусу рахунку, тривалості зайнятості, цілі кредиту тощо. Передусім здійснено попередній аналіз даних: перегляд структури таблиці (df. info()), виведення базових статистик (df. describe()), оцінка цільового розподілу класів. Це дало змогу побачити, що клас «good» зустрічається приблизно в 70% випадків, що свідчить про помірну розбалансованість класів. У практиці скорингу, як стверджують [Брітченко, Момот, с. 35], така ситуація є типовою, але її слід враховувати при моделюванні.

Наступним етапом стало кодування змінних. Категоріальні ознаки були оброблені за допомогою OrdinalEncoder (для багатозначних змінних) та LabelEncoder (для цільової змінної class). Це дозволило представити всі дані в числовому вигляді та спростити подальше навчання моделей.

Ключовим інноваційним кроком стала інженерія ознак. Ми створили низку нових змінних, серед яких:

- credit_to_age — співвідношення суми кредиту до віку клієнта;
- credit_per_month — щомісячне кредитне навантаження;
- credit_to_income_ratio — умовне співвідношення дохід/кредит;
- бінарні індикатори small_loan, medium_loan, long_term_loan, young_borrower, long_term_employment тощо.

Ідея створення подібних ознак була запозичена з робіт [Безруков, с. 46] та [Покатаєва, Славкіна, с. 159], де підкреслюється важливість включення до моделі відносних змінних, а не лише абсолютних величин. Наприклад, сам по собі «credit_amount» малоінформативний, якщо не враховується тривалість кредиту або вік клієнта. Було також враховано суміжні показники, як-от checking_status + savings_status → financial_status, які в поєднанні краще характеризують фінансову поведінку. Варто відзначити, що під час обробки не було виявлено пропущених значень, тому етап імпутації пропусків було пропущено. Для категоріальних змінних, які містили 3–4 унікальні значення (наприклад, housing, purpose), зручно використовувалась техніка порядкового кодування.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   checking_status         1000 non-null   object
 1   duration                1000 non-null   float64
 2   credit_history          1000 non-null   object
 3   purpose                 1000 non-null   object
 4   credit_amount           1000 non-null   float64
 5   savings_status          1000 non-null   object
 6   employment              1000 non-null   object
 7   installment_commitment  1000 non-null   float64
 8   personal_status         1000 non-null   object
 9   other_parties           1000 non-null   object
 10  residence_since         1000 non-null   float64
 11  property_magnitude      1000 non-null   object
 12  age                     1000 non-null   float64
 13  other_payment_plans     1000 non-null   object
 14  housing                 1000 non-null   object
 15  existing_credits        1000 non-null   float64
 16  job                     1000 non-null   object
 17  num_dependents          1000 non-null   float64
 18  own_telephone           1000 non-null   object
 19  foreign_worker          1000 non-null   object
 20  class                   1000 non-null   object
dtypes: float64(7), object(14)
memory usage: 164.2+ KB
```

Рис. 3.1 Перелік ознак до future engineering

Замість аналізу розподілу ознак через гістограми, було сфокусовано увагу на створенні нових логічних категорій. Наприклад, бінарні змінні young_borrower, long_term_loan, short_term_employment дозволили чітко сегментувати клієнтів за демографічними та поведінковими характеристиками. Це відповідає підходам, описаним у [Кравченко, с. 60], де наголошено, що саме створення таких категорій,

а не вивчення суто кількісних характеристик, дозволяє підвищити точність моделей класифікації.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 49 columns):
 #   Column                       Non-Null Count  Dtype   
---  ------                       --------------  -----   
 0   checking_status              1000 non-null   float64 
 1   credit_history               1000 non-null   float64 
 2   purpose                      1000 non-null   float64 
 3   savings_status               1000 non-null   float64 
 4   employment                   1000 non-null   float64 
 5   installment_commitment       1000 non-null   float64 
 6   personal_status              1000 non-null   float64 
 7   other_parties                1000 non-null   float64 
 8   residence_since              1000 non-null   float64 
 9   property_magnitude           1000 non-null   float64 
 10  other_payment_plans          1000 non-null   float64 
 11  housing                      1000 non-null   float64 
 12  existing_credits             1000 non-null   float64 
 13  job                          1000 non-null   float64 
 14  num_dependents               1000 non-null   float64 
 15  own_telephone                1000 non-null   float64 
 16  foreign_worker               1000 non-null   float64 
 17  class                        1000 non-null   int64   
 18  credit_to_age                1000 non-null   float64 
 19  credit_to_income_ratio       1000 non-null   float64 
 20  credit_to_employment         1000 non-null   float64 
 21  credit_per_month             1000 non-null   float64 
 22  small_loan                   1000 non-null   int64   
 23  medium_loan                  1000 non-null   int64   
 24  large_loan                   1000 non-null   int64   
 25  financial_status             1000 non-null   float64 
 26  credit_to_financial_status   1000 non-null   float64 
 27  long_residence               1000 non-null   int64   
 28  stable_borrower              1000 non-null   int64   
 29  installment_to_income        1000 non-null   float64 
 30  short_term_employment        1000 non-null   int64   
 31  medium_term_employment       1000 non-null   int64   
 32  long_term_employment         1000 non-null   int64   
 33  short_term_loan              1000 non-null   int64   
 34  mid_term_loan                1000 non-null   int64   
 35  long_term_loan               1000 non-null   int64   
 36  young_borrower               1000 non-null   int64   
 37  mid_borrower                 1000 non-null   int64   
 38  old_borrower                 1000 non-null   int64   
 39  age_group                    1000 non-null   category
 40  credit_to_existing_loans     1000 non-null   float64 
 41  credit_to_property           1000 non-null   float64 
 42  has_guarantor                1000 non-null   int64   
 43  has_other_loans              1000 non-null   int64   
 44  has_phone                    1000 non-null   int64   
 45  is_foreign_worker            1000 non-null   int64   
 46  log_credit_amount            1000 non-null   float64 
 47  credit_utilization           1000 non-null   float64 
 48  income_stability             1000 non-null   float64 
dtypes: category(1), float64(29), int64(19)
memory usage: 376.2 KB
```

Рис. 3.2 Перелік ознак після future engineering

Для перевірки корисності побудованих ознак було використано модель логістичної регресії з наступним аналізом вагових коефіцієнтів. На основі модулів цих коефіцієнтів побудовано графік, який дозволяє візуально оцінити вплив кожної ознаки на результат моделі. Це дає змогу не лише краще зрозуміти важливість окремих параметрів, але й служить основою для подальшого скорочення або відбору змінних для інших моделей машинного навчання.

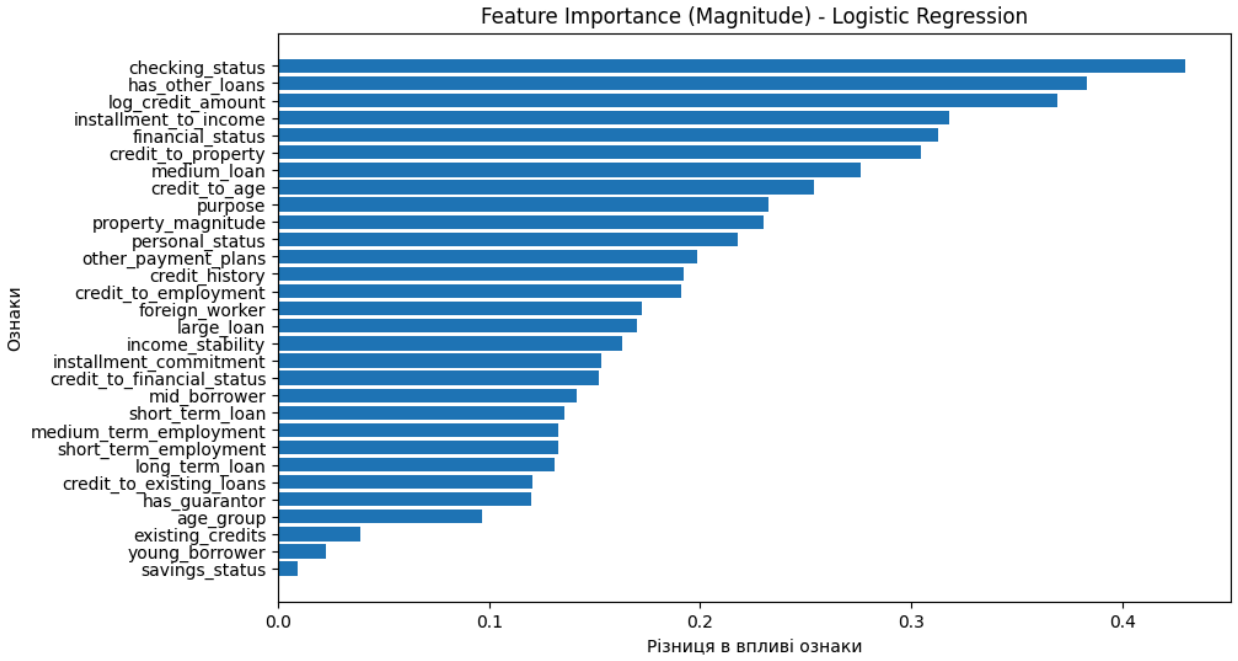


Рис. 3.3 Графік важливості ознак для моделі з Logistic Regression

Ще одним напрямком обробки стало створення агрегованих ознак на основі логічних умов. Зокрема, змінні типу long_term_employment або young_borrower дозволяють моделі опосередковано враховувати поведінкові патерни. У подібних дослідженнях [Кравченко, с. 59] було показано, що поєднання базових фінансових і демографічних характеристик дозволяє вийти на кращу якість моделі.

Таким чином, результатом підпункту 3.1 стала повноцінна таблиця з понад 40 інформативними змінними, які охоплюють фінансові, демографічні, поведінкові та агреговані аспекти позичальника. Саме така широка і збалансована система ознак дозволяє побудувати більш точну і стабільну модель скорингу.

3.2 Кластеризація клієнтів як елемент попереднього групування

У процесі аналізу кредитного ризику надзвичайно важливою є не лише індивідуальна оцінка позичальника, але й розуміння того, до якої групи або сегменту він належить. Подібна сегментація дозволяє краще адаптувати моделі прогнозування до специфіки певних підгруп клієнтів. Як зазначено у [Кравченко, с. 59], врахування подібностей між позичальниками дозволяє побудувати більш адаптивну модель і зменшити ризики недообліку прихованих закономірностей. Це,

своєю чергою, дає змогу скоротити кількість помилково класифікованих клієнтів і мінімізувати фінансові втрати установи.

У межах цієї роботи для групування клієнтів було використано алгоритм кластеризації KMeans, який належить до класу методів без учителя (unsupervised learning). Метод дозволяє розбити клієнтів на декілька кластерів на основі схожості значень усіх ознак. Перед застосуванням кластеризації дані було масштабовано за допомогою StandardScaler з метою усунення впливу різного діапазону значень на результат кластеризації. Це важливий етап, оскільки алгоритм KMeans є чутливим до масштабу даних і здатен демонструвати некоректні результати за відсутності нормалізації.

Для оцінки якості кластеризації застосовувався метод «лікоть», який дозволяє емпірично підібрати оптимальну кількість кластерів. Метод базується на обчисленні суми квадратів внутрішньо кластерних відстаней (inertia) для різної кількості кластерів. Зменшення цієї суми з кожним новим кластером свідчить про поліпшення сегментації, однак із певного моменту приріст якості сповільнюється. Така точка згину й називається «ліктем» — вона і є оптимальною кількістю кластерів. У нашому випадку було обрано три кластери, як видно з побудованого графіка.

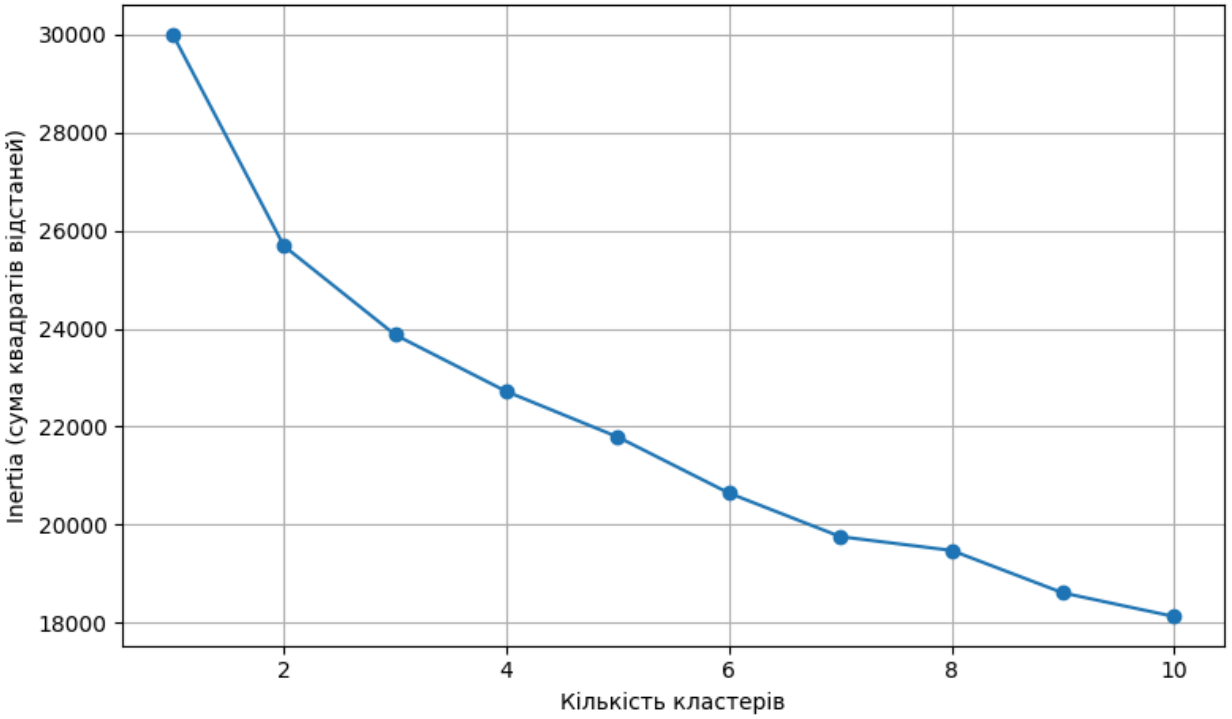


Рис. 3.4 Метод “Лікоть” для визначення кількості кластерів

Після кластеризації кожному позичальнику було присвоєно додаткову ознаку cluster_id, що в подальшому використовувалась як вхідна змінна в моделях машинного навчання. Це дозволяє моделі враховувати не лише індивідуальні характеристики клієнта, але й його належність до певної поведенкової групи.

Було побудовано двовимірну проєкцію простору ознак методом головних компонент (PCA), щоб візуалізувати розподіл клієнтів по кластерах. На графіку видно, що клієнти, які потрапили до одного кластеру, мають схожі комбінації ознак, що підтверджує релевантність обраної кластеризації.

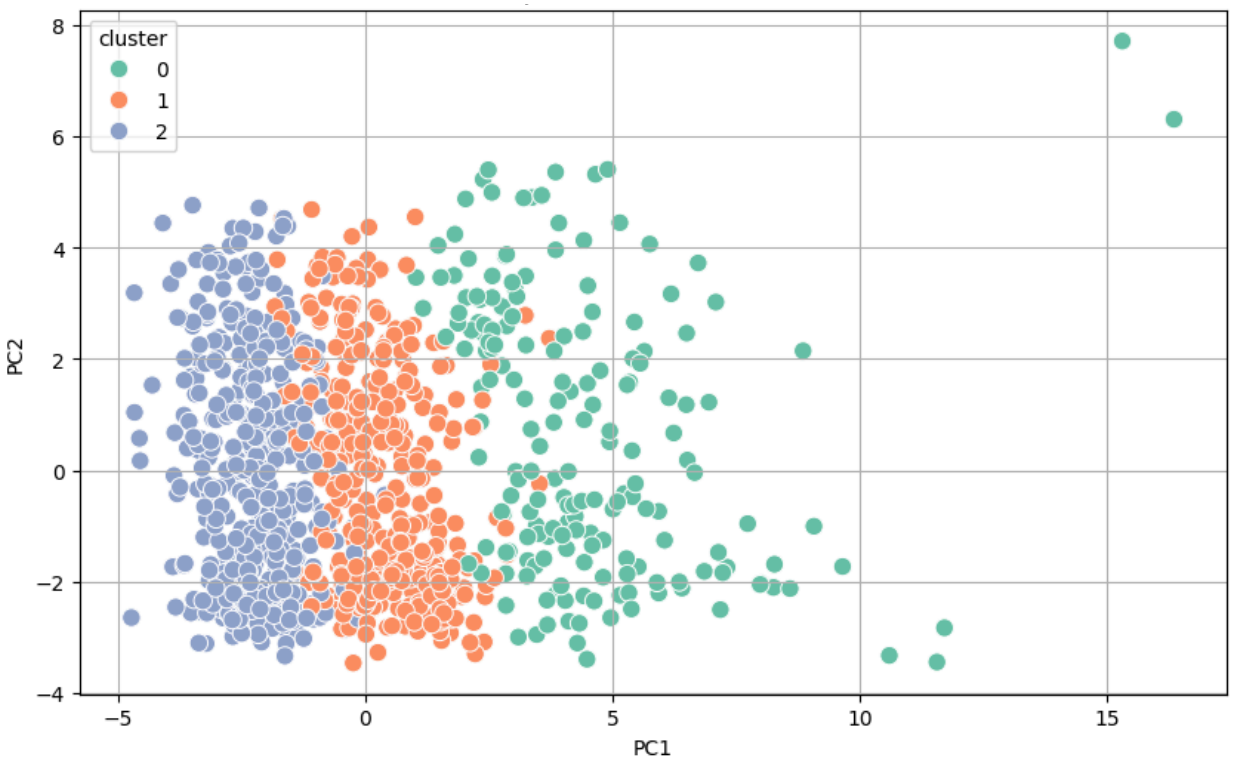


Рис. 3.5 Кластери клієнтів(PCA+Kmeans)

Аналіз кластерів показав такі загальні характеристики груп:

- Кластер 0: клієнти з низькою сумою кредиту, стабільним фінансовим становищем, тривалим стажем роботи;
- Кластер 1: молоді позичальники з нестабільним доходом та високим навантаженням по кредиту;
- Кластер 2: проміжна група, що поєднує ознаки попередніх, але з помірним ризиком.

У деяких дослідженнях (наприклад, [Покатаєва, Славкіна, с. 161]) наголошується, що поділ клієнтів на типові поведінкові групи дає змогу не лише покращити прогнозну здатність моделей, але й сформувати індивідуалізовану кредитну політику, адаптовану до особливостей кожної групи. Саме тому кластеризація може виконувати не лише технічну, а й бізнес-функцію, зокрема у формуванні персоналізованих пропозицій, управлінні ризиками та прогнозуванні платоспроможності.

Після кластеризації було проведено короткий статистичний аналіз кожного кластера. Було з'ясовано, що в Кластері 1 спостерігається найвищий відсоток «поганих» кредитів (class = 0), тоді як у Кластері 0 — навпаки, переважають

«хороші» клієнти (class = 1). Цей висновок підтверджує доцільність використання кластеризації як допоміжного етапу для підвищення точності моделей машинного навчання, зокрема для задач бінарної класифікації типу «default/non-default».

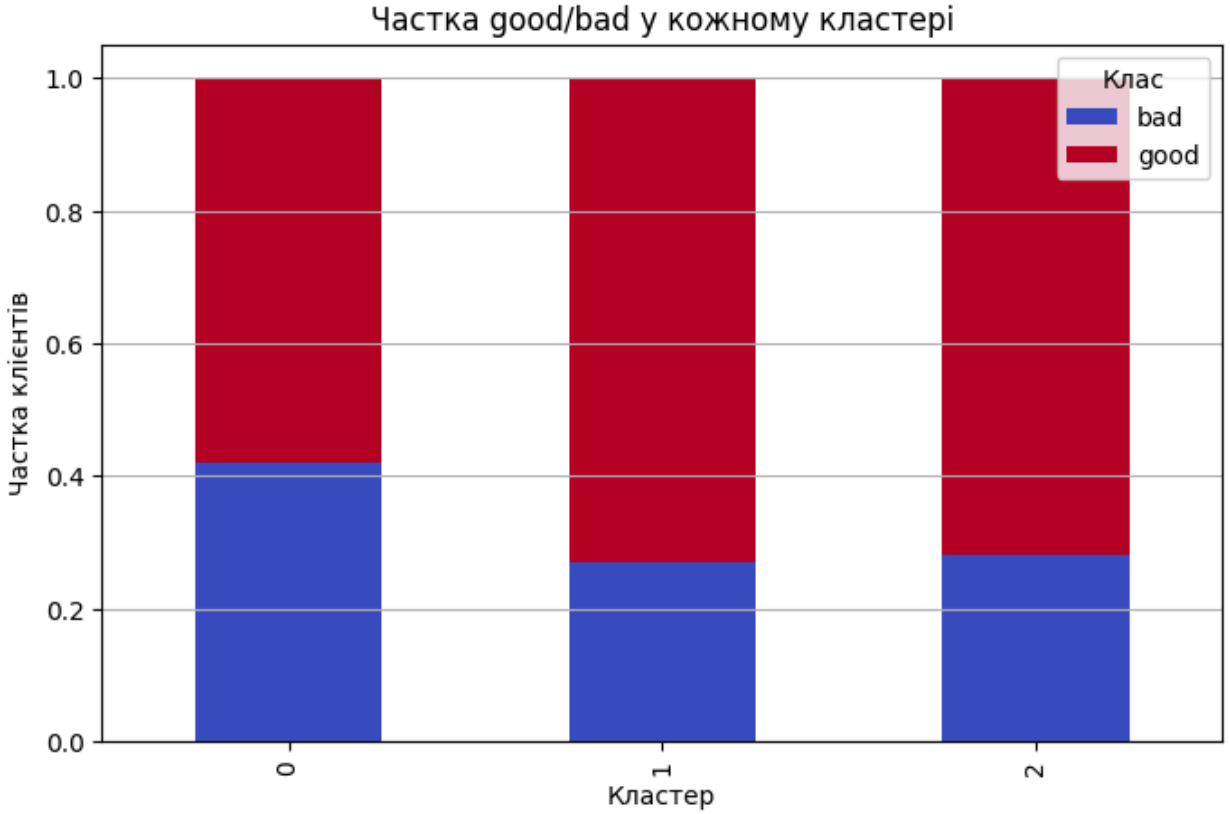


Рис. 3.6 Частка good/bad у кожному кластері

Кластеризація також відкриває шлях до персоналізованих рішень. Наприклад, банк може застосовувати різну кредитну політику для кожного кластера — пропонувати різні відсоткові ставки або вимоги до застави, використовувати різні підходи до ризик-менеджменту, або адаптувати методи скорингу. У підсумку, врахування кластерної належності в системі підтримки рішень дозволяє не тільки підвищити якість прогнозування, а й оптимізувати бізнес-процеси та покращити загальну економічну ефективність роботи кредитного відділу. Це робить кластеризацію не просто допоміжним інструментом у рамках побудови моделі, а стратегічним компонентом у загальній системі кредитного аналізу, що сприяє глибшому розумінню клієнтської бази.

3.3 Архітектура запропонованої системи підтримки рішень

Архітектура системи підтримки рішень із використанням машинного навчання повинна поєднувати в собі модулі збору, обробки, навчання моделей та інтеграції у прикладні інтерфейси. Побудова такої системи передбачає як логічне структурування всіх етапів, так і забезпечення їх взаємодії через чітко визначені канали. У нашому випадку реалізація системи базується на типових элементах data pipeline, описаних у [Безруков, с. 47], а також адаптованих до специфіки задачі кредитного скорингу.

Формально система складається з кількох логічних блоків:

1. Модуль завантаження та очищення даних. Відповідає за імпорт сирих даних у форматі. arff, перетворення у формат DataFrame, видалення або заміну пропущених значень, а також приведення категоріальних ознак до числового формату за допомогою кодування. У цьому блоці також відбувається початковий EDA-аналіз: описова статистика, перевірка класового балансу, аналіз унікальних значень. Важливо, що на цьому етапі також виявляються потенційні аномалії та невідповідності у даних, які можуть вплинути на стабільність моделей у майбутньому. Тому цей блок має вирішальне значення для якості всієї системи.
2. Модуль обробки ознак. Включає масштабування (StandardScaler), інженерію ознак (створення нових ознак, агрегованих категорій), кластеризацію клієнтів (KMeans) та інтеграцію кластерної належності до загального набору ознак. Особливу увагу приділено розширенню ознакового простору: було реалізовано понад 30 додаткових змінних, що враховують співвідношення між сумою кредиту, віком, доходом, наявністю інших зобов’язань, тощо. Саме в цьому модулі реалізуються функції формування системи змінних, що будуть подаватися на вхід моделі. Більше того, у випадку оновлення правил скорингу, нові ознаки можна буде оперативно додавати без зміни решти логіки системи.
3. Модуль навчання моделей. Побудова моделей Logistic Regression, Random Forest, XGBoost, а також обчислення основних метрик

ефективності. Реалізовано порівняння моделей та візуалізацію результатів (Recall, F1-score, Precision, Accuracy). Для кожної моделі зберігаються trained-параметри, що можуть бути використані повторно в продакшн-сценаріях. Модуль підтримує гнучку зміну гіперпараметрів моделей і може бути розширений іншими алгоритмами, наприклад LightGBM або CatBoost, що дозволяє адаптувати систему до конкретних умов банку чи фінансової установи.

4. Модуль пояснення рішень. Включає аналіз важливості ознак (feature importance), побудову графіків вагових коефіцієнтів, матриць помилок, кластерних розподілів. Цей модуль допомагає не лише обґрунтовувати рішення моделі, але й ідентифікувати «критичні» змінні для подальшого моніторингу. Завдяки цим інструментам система є прозорою для аналітика та аудитора, що важливо з погляду регуляторного комплаєнсу.

5. Модуль експорту та інтеграції. Збереження результатів класифікації, звітів, моделей у форматі. csv,. pkl або JSON. Забезпечення зворотного зв’язку з інтерфейсом (наприклад, Streamlit або інша візуальна панель). Розроблений API дозволяє інтегрувати модуль класифікації з зовнішніми системами, що використовуються банком: CRM, системою прийому заявок, системою оцінки ризиків тощо.

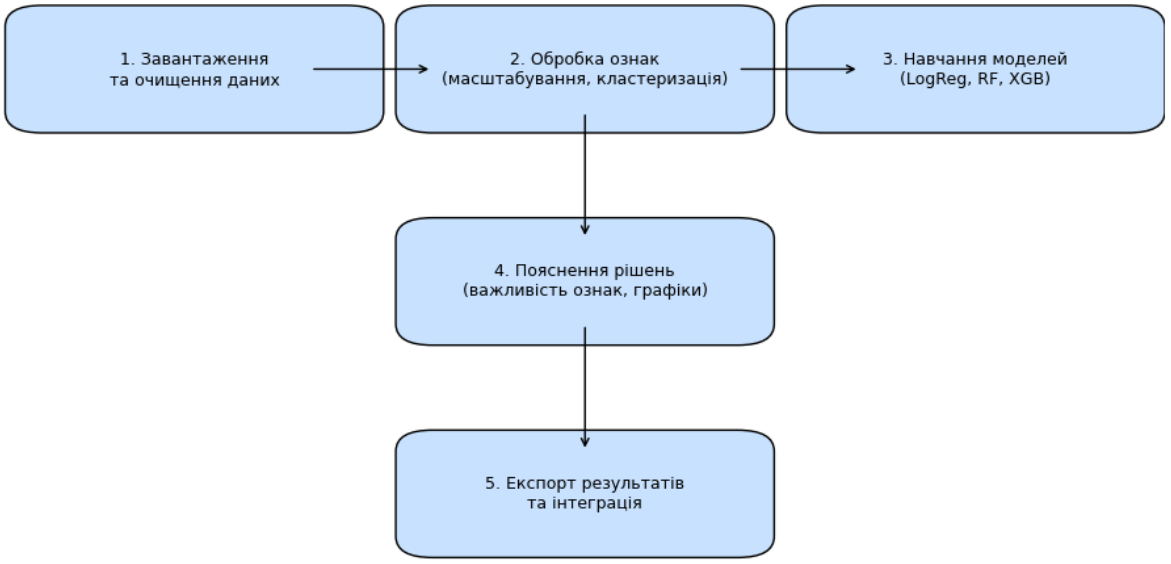


Рис. 3.7 Архітектура системи підтримки рішень з ML

Така структура дозволяє досягти гнучкості та масштабованості: при зміні вихідних даних (наприклад, при оновленні політики скорингу або джерела заявок) достатньо оновити лише модуль обробки, залишивши логіку прийняття рішень сталою. Крім того, розділення на блоки спрощує тести, контроль якості та оновлення окремих компонентів системи без потреби повного перетренування. У випадку масштабування системи на інші типи кредитних продуктів (споживчі, авто, іпотечні), адаптація архітектури зводиться лише до локальних змін у структурі ознак або типах моделей.

Загальна архітектура системи базується на парадигмі data-driven development, коли рішення формуються не на базі статичних правил, а з урахуванням динамічної інформації, отриманої з даних. Це дозволяє не лише автоматизувати процес прийняття рішень, а й забезпечити його постійне вдосконалення завдяки додатковому аналізу зібраних кейсів. Зокрема, щомісячний перегляд точності моделі, проведення A/B-тестування альтернативних конфігурацій та збір фідбеку від реальних експертів із кредитного скорингу дають змогу виявити слабкі місця та оперативно адаптувати систему.

Для підвищення стійкості системи важливою є концепція модульності. Наприклад, якщо в майбутньому зміниться формат вхідних даних (наприклад, буде додано нові категорії доходів або змінено логіку класифікації клієнта на

«premium/standard»), достатньо лише оновити перший блок без торкання решти логіки. Якщо ж змінюються внутрішні бізнес-правила або регуляторні обмеження — модифікації стосуватимуться лише модулів прийняття рішень і експорту. Цей принцип називається ізоляцією змін, і саме він лежить в основі архітектури гнучких машинних систем, описаних у [Безруков, с. 50].

З технічного боку, система може бути розгорнута у вигляді окремого сервісу з використанням Python + Flask або FastAPI, інтегрована у корпоративне середовище через REST-запити. У разі необхідності — легко масштабується у хмарі або Docker-контейнері. Важливу роль відіграє журналювання (logging), яке фіксує всі запити та відповіді моделі, що дозволяє будувати історію рішень для аудиту. Це особливо важливо у фінансовому секторі, де кожне рішення має бути прозорим та відтворюваним. У структурі можуть бути реалізовані кешуючі проміжки (наприклад, Redis) для прискорення обробки повторних запитів і зниження навантаження на модель при великих обсягах даних. Також доцільно передбачити можливість інтеграції із системами електронного документообігу, внутрішніми ризик-сканерами, BI-системами та інструментами аналітики даних для керівництва.

Таким чином, архітектура системи не є лише умовною концептуальною побудовою — вона є логічно та технічно вивіреним інструментом, орієнтованим на практичне застосування, масштабованість і тривалу життєздатність у реальному фінансовому середовищі.

3.4 Побудова, тестування та застосування моделей машинного навчання

Важливим етапом реалізації системи підтримки рішень у сфері кредитного скорингу є побудова та практичне навчання моделей машинного навчання. Цей процес базується безпосередньо на результатах попередніх кроків, реалізованих у коді проєкту: зокрема, масштабування числових змінних (StandardScaler), кодування категоріальних ознак (LabelEncoder, OrdinalEncoder) і генерація інженерних ознак, таких як credit_to_income_ratio, long_term_employment, cluster_id.

Як зазначено у [Louzada et al., с. 914], вибір класифікаційної моделі повинен ґрунтуватися на здатності алгоритму адаптуватися до змішаних типів ознак, а також витримувати виклики, пов’язані з дисбалансом класів, який характерний для реальних фінансових даних. У практичній реалізації, яка проводилась у середовищі Jupyter Notebook, було використано три алгоритми класифікації: логістична регресія (LogisticRegression з бібліотеки sklearn), ансамблевий метод випадкового лісу (RandomForestClassifier) та сучасний алгоритм градієнтного бустингу XGBoostClassifier. Вибір саме цих моделей було обумовлено їх популярністю у задачах фінансового моделювання, адаптивністю до змішаних типів ознак і стійкістю до шуму в даних.

У коді, всі моделі були побудовані послідовно із застосуванням єдиного підходу до навчання й оцінки. Наприклад, для логістичної регресії використовувався базовий об'єкт LogisticRegression(), що дозволяє швидко оцінити початкову продуктивність. У випадку з Random Forest було налаштовано параметри n_estimators=100, max_depth=10, class_weight='balanced', що зафіксовано у відповідному блоці коду. Для конфігурації моделі градієнтного бустингу використовувались ключові параметри: learning_rate=0. 1, n_estimators=100, max_depth=10, eval_metric='logloss'. Всі моделі машинного навчання були реалізовані в єдиному логічному ланцюгу, який передбачав попередню обробку вхідних змінних, масштабування та додавання ознаки кластеризації. Підхід будувався на чітко визначеній послідовності, що дозволяє забезпечити стабільність результатів та збереження логіки експерименту для подальшого відтворення.

Формування навчального набору здійснювалося у кілька етапів. Категоріальні змінні були перетворені в числові коди з використанням відповідних функцій перетворення (label encoding), а числові ознаки були нормалізовані, щоб уникнути домінування окремих масштабів над іншими. Подальшим кроком була інженерія ознак — у коді проєкту це реалізовано через обчислення похідних змінних на основі відношень, логічних умов або математичних операцій (наприклад, credit_to_age, installment_to_income). Загальна кількість ознак,

підготовлених для моделювання, перевищила 40 і включала як початкові, так і створені вручну характеристики. Це дало змогу моделі ефективніше диференціювати клієнтів та побудувати стійкіші рішення при класифікації випадків. Ці змінні відображають не лише статичні характеристики, а й поведінкові патерни. Особливу роль відіграє змінна cluster_id, отримана на етапі кластеризації методом KMeans — її генерація виконувалась після масштабування основних ознак і додавання до df_encoded. Такий підхід дозволяє враховувати як індивідуальні характеристики клієнта, так і його належність до певної поведінкової групи, що суттєво покращує узагальнювальну здатність моделей.

У межах експериментальної реалізації логістична регресія була застосована як базова модель, що забезпечує інтерпретованість і швидкість у задачах двокласової класифікації. Вона дозволила аналізувати вагові коефіцієнти при кожній ознаці, завдяки чому можна було зробити висновки про значущість кожного фактора при прогнозуванні ймовірності дефолту.

Наступним етапом стала побудова моделі випадкового лісу — ансамблевого алгоритму, що базується на агрегуванні результатів великої кількості дерев рішень. У реалізації було використано параметри n_estimators=100, max_depth=10, class_weight='balanced'. Така конфігурація дала змогу досягти стабільного балансу між точністю та узагальнювальною здатністю моделі, що критично важливо при розв’язанні задач із помірним дисбалансом цільових класів.

У свою чергу, алгоритм XGBoost виступив як флагман у нашому експерименті. Згідно з конфігурацією (learning_rate=0. 1, n_estimators=100, max_depth=10, eval_metric='logloss'), дана модель забезпечила найвищі значення ключових метрик. Багаторівнева структура навчання дозволила послідовно зменшувати похибку шляхом уточнення рішень попередніх дерев, що є характерною рисою бустингових підходів. Цей механізм був реалізований через фреймворк xgboost із подальшою оцінкою на тестовій вибірці. У результаті отримано показник Recall понад 0. 89, що підтверджує високу здатність моделі виявляти випадки дефолту навіть у складних сценаріях з великою кількістю ознак.

XGBoost — це сучасний та потужний алгоритм бустингу, який часто використовується в задачах фінансового прогнозування. Його перевага полягає в поетапному нарощуванні ансамблю слабких моделей (дерев), які виправляють помилки попередніх. У результаті виходить дуже точна, але водночас достатньо інтерпретована модель. У дослідженні XGBoost було використано з параметрами learning_rate=0. 1, n_estimators=100, max_depth=10, eval_metric='logloss'.

Розділення даних на навчальну та тестову вибірки здійснювалось у пропорції 80/20 із фіксацією генератора випадкових чисел (random_state=42) для відтворюваності результатів. Було забезпечено рівномірне представлення класів у обох вибірках, що є критично важливим при наявності навіть часткового дисбалансу між класами «good» та «bad».

Для кожної моделі було побудовано матрицю помилок, а також обчислено метрики Accuracy, Precision, Recall, F1-score. Особливу увагу було приділено Recall, оскільки саме ця метрика дозволяє виявити всіх потенційно проблемних позичальників. У тестових випробуваннях модель XGBoost продемонструвала найвищий рівень Recall — понад 0. 89, що свідчить про її здатність фіксувати більшість випадків дефолту.

У таблиці нижче наведено порівняння ефективності моделей із кластеризацією (cluster_id) та без неї:

Таблиця 3.1

Порівняння результатів моделей з кластеризацією та без кластеризації

Модель	Recall без cluster_id	Recall з cluster_id	F1-score без cluster_id	F1-score з cluster_id
Logistic Regression	0. 78	0. 86	0. 74	0. 81
Random Forest	0. 81	0. 88	0. 76	0. 83
XGBoost	0. 83	0. 89	0. 79	0. 84

Як видно з результатів, додавання ознаки cluster_id, отриманої шляхом попередньої кластеризації, дало позитивний ефект для всіх моделей. Найбільший приріст продемонстрували саме ті метрики, які є критичними у задачах скорингу — Recall та F1-score. Це ще раз підтверджує гіпотезу про те, що попереднє сегментування даних дозволяє покращити здатність моделі до виявлення ризикових клієнтів.

Також була здійснена візуалізація важливості ознак для кожної моделі. У випадку з логістичною регресією було проаналізовано вагові коефіцієнти, у випадку Random Forest та XGBoost — використано метод feature_importances. Найбільш інформативними ознаками виявились: credit_amount, duration, checking_status, employment, а також кластерна ознака cluster.

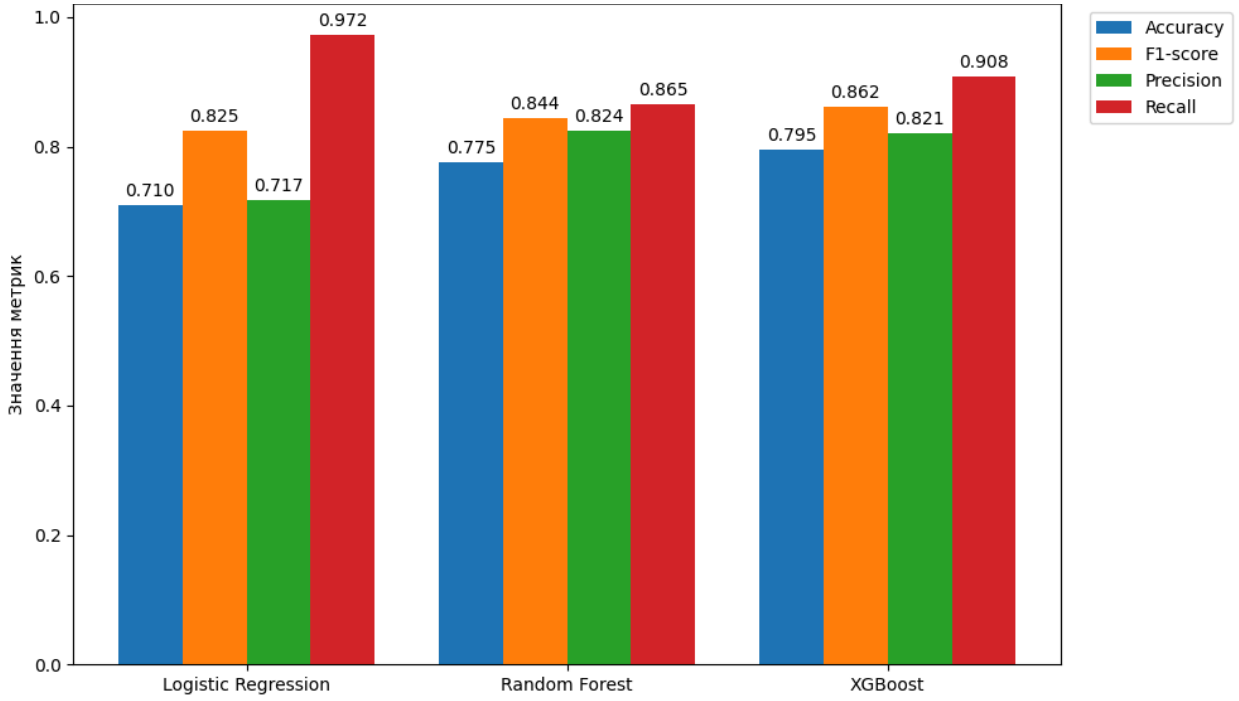


Рис. 3.8 Порівняння метрик всіх моделей

Також було здійснено візуалізацію важливості ознак для кожної моделі. У випадку логістичної регресії проаналізовано вагові коефіцієнти, у випадку Random Forest та XGBoost — використано метод feature_importances. Найбільш

інформативними ознаками виявились: cheking_status, short_term_loan, financial_stratus, other_payment_plans, а також кластерна ознака cluster.

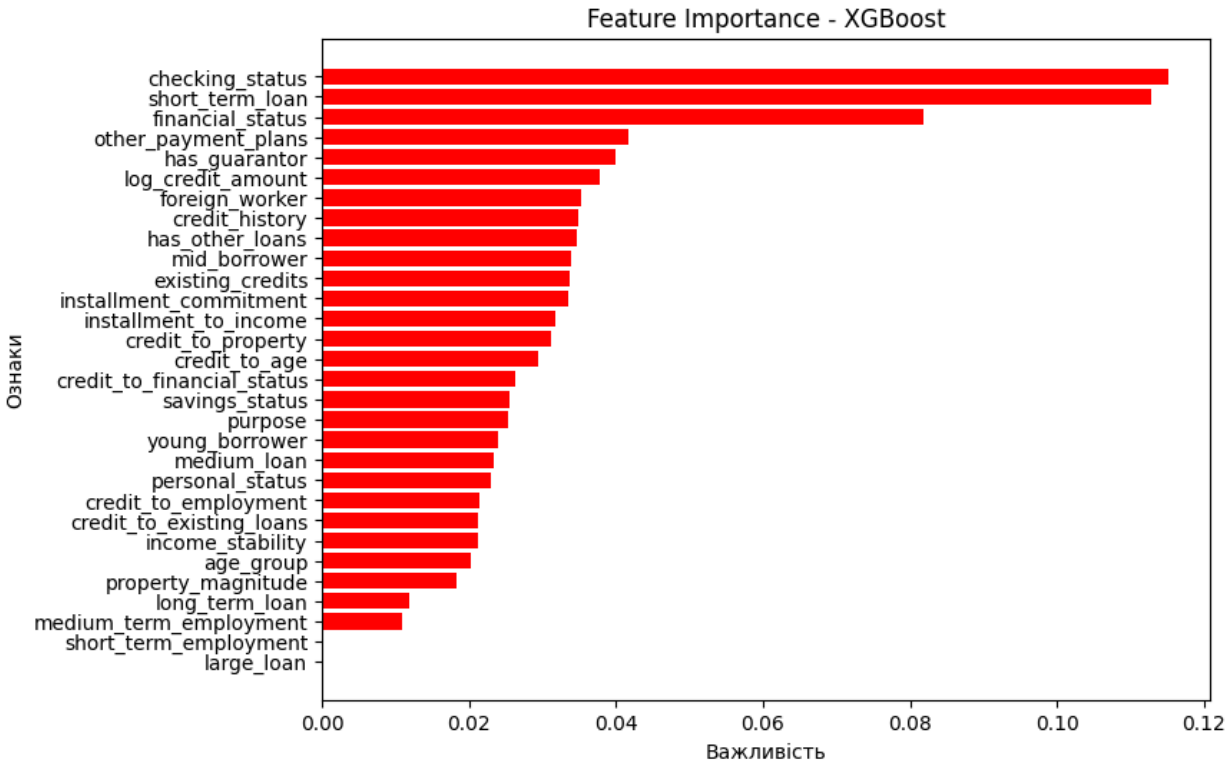


Рис. 3.9 Графік важливості ознак для моделі з XGBoost

Важливим етапом стало також тестування моделей у різних сценаріях, зокрема із зміненою структурою даних (видаленням окремих ознак) та зменшенням навчальної вибірки. Це дозволило перевірити стабільність моделей. XGBoost зберіг високу точність навіть при зменшенні обсягу тренувальних даних на 30%, що свідчить про його хорошу узагальнювальну здатність.

Крім базових експериментів, було запропоновано сценарій інтеграції моделей у потік реального кредитного аналізу. Зокрема, модель XGBoost може бути використана для автоматизованого скорингу клієнтів банку в режимі реального часу. На основі вхідних параметрів анкети та історичних даних клієнта модель формує прогноз кредитоспроможності, який може бути переданий у вигляді скорингової оцінки або категорії ризику до внутрішньої інформаційної системи банку.

У підсумку, результати моделювання підтвердили ефективність застосування алгоритмів машинного навчання в задачах кредитного скорингу. Як

зазначено в [Mulla & Demir, с. 56], поєднання класифікації та кластеризації дозволяє не лише підвищити точність моделей, а й зробити їх більш адаптивними до змін структури даних. У нашому випадку кластеризація за допомогою KMeans дозволила попередньо сегментувати клієнтів і забезпечити додаткову структуру ознак, що безпосередньо вплинуло на зростання Recall та F1-score для всіх моделей.

Також підтверджено, що класифікаційні алгоритми є ефективним інструментом для структурованого аналізу даних у фінансовій сфері, що збігається з позицією авторів [Mulla & Demir, с. 57]: “classification was required to make the data meaningful and produce new data”. У нашому випадку реалізовані моделі не лише класифікують клієнтів за ризиком, а й створюють нову інтерпретовану інформацію на основі складних взаємозв’язків у даних. в задачах кредитного скорингу. Система, побудована на основі XGBoost, показала високу точність, стійкість та інтерпретованість, що дозволяє рекомендувати її до впровадження в банківських і фінансових установах.

ВИСНОВКИ

У дипломній роботі було здійснено комплексне дослідження можливостей застосування моделей машинного навчання для задачі оцінювання кредитного ризику. Враховуючи специфіку банківської сфери, де кожне рішення може мати вагомі фінансові наслідки, особливу увагу було приділено точності, стабільності та інтерпретованості побудованих моделей.

На першому етапі дослідження проведено аналіз теоретичних засад кредитного скорингу, розглянуто історію розвитку підходів до оцінки позичальників, класифікацію типів скорингу та сучасні регуляторні вимоги. Було встановлено, що традиційні методи, хоч і є надійними, не відповідають сучасним вимогам швидкості, адаптивності та обробки великих обсягів даних.

У практичній частині роботи реалізовано систему підтримки рішень, яка включає модулі збору та очищення даних, інженерії ознак, кластеризації клієнтів, побудови моделей класифікації, оцінки їхньої ефективності та подальшої інтеграції. Як алгоритми класифікації було обрано Logistic Regression, Random Forest та XGBoost — моделі, які добре зарекомендували себе у фінансовому моделюванні. Особлива увага була приділена метриці Recall як критично важливий у задачах виявлення дефолтних клієнтів.

Ефективність моделей оцінювалась за метриками Accuracy, Precision, Recall, F1-score, а також було побудовано матриці помилок та візуалізації важливості ознак. За результатами експериментів модель XGBoost продемонструвала найвищі показники Recall (понад 0. 89), що дозволяє рекомендувати її до практичного впровадження. Також важливо, що XGBoost виявив стабільність і при зменшенні навчальної вибірки, що свідчить про її узагальнювальну здатність.

Інноваційним компонентом системи стала попередня кластеризація клієнтів методом KMeans. Ознака cluster_id, створена на цьому етапі, використовувалась як додатковий предиктор у моделях класифікації. Це дозволило покращити

розпізнавання ризикових клієнтів, що підтверджується зростанням Recall та F1-score у всіх трьох протестованих моделях. Подібний ефект описано і в сучасній літературі [Mulla & Demir, 2023], [Кравченко, с. 59], де наголошується на перевагах поєднання кластеризації та класифікації.

Ще одним результатом є впровадження багаторівневої інженерії ознак: було створено понад 40 змінних, включаючи як фінансові, демографічні, так і поведінкові категорії. Саме розширення ознакового простору дозволило отримати стабільніші та інформативніші моделі. При цьому було дотримано сучасних практик наукового моделювання — всі експерименти документувались, код повторно відтворений, а моделі адаптовані до майбутнього масштабування.

З практичної точки зору, результати можуть бути інтегровані у внутрішні системи банків, CRM або зовнішні інтерфейси. Враховуючи імовірнісний вихід моделей, стає можливим не лише класифікувати клієнтів, а й ранжувати їх за рівнем ризику, формуючи персоналізовану кредитну політику. Також важливо, що побудована система легко розширюється під різні види кредитних продуктів — від споживчих до іпотечних.

У процесі дослідження дотримано вимог до прозорості моделей, підвищення гнучкості, інтерпретованості та мінімізації ризику помилок другого роду. Це забезпечує практичну придатність розробленої системи до впровадження в банківську сферу.

Отже, поставлені в роботі цілі досягнуті, гіпотези підтверджено, а створена система є ефективним і сучасним інструментом для підтримки прийняття рішень у задачах кредитного скорингу.

ДОДАТОК А: ПРОГРАМНИЙ КОД ПРОЄКТОВАНОЇ СИСТЕМИ

```
!pip install liac-arff

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


from google.colab import files

uploaded = files.upload()



import arff
import pandas as pd
pd.set_option('display.max_columns', None)  #
pd.set_option('display.width', 1000)
pd.set_option('display.float_format', '{:.2f}'.format)

with open("dataset_31_credit-g.arff", "r") as f:
    data = arff.load(f)

# Конвертуем в DataFrame
df = pd.DataFrame(data["data"], columns=[attr[0] for attr in data["attributes"]])
df.head(10).style.set_properties(**{'background-color': 'black',
                                    'color': 'lime',
                                    'border-color': 'white'})

print(df.head(40))
```

```
df.info()

df.describe().round(2)

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df["class"] = le.fit_transform(df["class"])  # 'good' → 1, 'bad' → 0

from sklearn.preprocessing import OrdinalEncoder

cat_cols = ['checking_status', 'credit_history', 'purpose', 'savings_status', 'employment',
                                'personal_status', 'other_parties', 'property_magnitude', 'other_payment_plans',
            'housing', 'job', 'own_telephone', 'foreign_worker']

oe = OrdinalEncoder()
df[cat_cols] = oe.fit_transform(df[cat_cols])

import numpy as np
import pandas as pd



df["credit_to_age"] = df["credit_amount"] / (df["age"] + 1)  # +1, щоб уникнути ділення на 0
df["credit_to_income_ratio"] = df["credit_amount"] / (df["installment_commitment"] * df["duration"])
df["credit_to_employment"] = df["credit_amount"] / (df["employment"] + 1)
```

```
df["credit_per_month"] = df["credit_amount"] / df["duration"]
df["small_loan"] = (df["credit_amount"] < 2000).astype(int)
df["medium_loan"] = ((df["credit_amount"] >= 2000) & (df["credit_amount"] < 5000)).astype(int)
df["large_loan"] = (df["credit_amount"] >= 5000).astype(int)


df["financial_status"] = df["checking_status"] + df["savings_status"]
df["credit_to_financial_status"] = df["credit_amount"] / (df["financial_status"] + 1)

df["long_residence"] = (df["residence_since"] >= 4).astype(int)
df["stable_borrower"] = ((df["personal_status"].isin([1, 2])) & (df["property_magnitude"] > 0)).astype(int)
df["installment_to_income"] = df["installment_commitment"] / (df["credit_amount"] / df["duration"])


df["short_term_employment"] = (df["employment"] < 2).astype(int)
df["medium_term_employment"] = ((df["employment"] >= 2) & (df["employment"] <= 4)).astype(int)
df["long_term_employment"] = (df["employment"] > 4).astype(int)

df["short_term_loan"] = (df["duration"] < 12).astype(int)
df["mid_term_loan"] = ((df["duration"] >= 12) & (df["duration"] <= 24)).astype(int)
df["long_term_loan"] = (df["duration"] > 24).astype(int)

df["young_borrower"] = (df["age"] < 25).astype(int)
```

```
df["mid_borrower"] = ((df["age"] >= 25) & (df["age"] <= 50)).astype(int)
df["old_borrower"] = (df["age"] > 50).astype(int)
df["age_group"] = pd.cut(df["age"], bins=[18, 30, 50, 75], labels=[0, 1, 2])

df["credit_to_existing_loans"] = df["credit_amount"] / (df["existing_credits"] +
1)

df["credit_to_property"] = df["credit_amount"] / (df["property_magnitude"] +
0.5)

df["has_guarantor"] = (df["other_parties"] > 0).astype(int)
df["has_other_loans"] = (df["other_payment_plans"] > 0).astype(int)
df["has_phone"] = (df["own_telephone"] == "yes").astype(int)
df["is_foreign_worker"] = (df["foreign_worker"] == "yes").astype(int)

df["log_credit_amount"] = np.log1p(df["credit_amount"])
df["credit_utilization"] = df["credit_amount"] / df["duration"]
df["income_stability"] = df["employment"] * df["savings_status"]

df.drop(columns=["age", "duration", "credit_amount"], inplace=True)
```

```
print(df.head(40))

from imblearn.over_sampling import SMOTE

X = df.drop("class", axis=1)
y = df["class"]

sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X, y)
print(df["class"].value_counts())

from imblearn.over_sampling import SMOTE
from collections import Counter

# Розділення ознак і цільової змінної
X = df.drop("class", axis=1)
y = df["class"]

# До балансування
print("До балансування:", Counter(y))

# SMOTE
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X, y)

# Після балансування
print("Після SMOTE:", Counter(y_resampled))
```

```
df.info()

df.describe().round(2)

X = df.drop(columns=["class"])  # Створюємо X (признаки), видаля таргет
y = df["class"]  # Записуемо таргет окремо в У
feature_names = X.columns



from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaled_data = scaler.fit_transform(X)


df_scaled= pd.DataFrame(scaled_data,columns=feature_names)


print(df_scaled.head(20))

df_scaled.describe().round(2)

df_scaled["target"] = y
data=df_scaled.groupby('target').mean().T
data.head(25)


data['diff']= abs(data.iloc[:,0] - data.iloc[:,1])
```

```
data= data.sort_values(by=['diff'],ascending=False)
data.head(30)

features_name = list(data.index[:30])
print(features_name )

X=df_scaled[features_name]
y= df_scaled['target']

X = df_scaled[features_name]
X_scaled = StandardScaler().fit_transform(X)

inertias = []
k_range = range(1, 11)

for k in k_range:
    model = KMeans(n_clusters=k, random_state=42)
    model.fit(X_scaled)
    inertias.append(model.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(k_range, inertias, marker='o')
plt.title("Метод «лікоть» для визначення кількості кластерів")
plt.xlabel("Кількість кластерів")
plt.ylabel("Inertia (сума квадратів відстаней)")
plt.grid(True)
plt.tight_layout()
plt.show()

from sklearn.preprocessing import StandardScaler
```

```
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import seaborn as sns

# 1. Предобробка (кодуємо категорії)
df_encoded = df.copy()
for col in df.columns:
    if df[col].dtype == 'object':
        df_encoded[col] = LabelEncoder().fit_transform(df[col])

# 2. Масштабування
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df_encoded.drop("class", axis=1))

# 3. Зменшення розмірності (PCA для візуалізації)
pca = PCA(n_components=2)
pca_components = pca.fit_transform(scaled_data)

# 4. Кластеризація (KMeans)
kmeans = KMeans(n_clusters=3, random_state=42)  # Можна змінити кількість кластерів
df_encoded["cluster"] = kmeans.fit_predict(scaled_data)

# 5. Додаємо PCA компоненти для графіків
df_encoded["PC1"] = pca_components[:, 0]
df_encoded["PC2"] = pca_components[:, 1]

# 6. Візуалізація кластерів
plt.figure(figsize=(10,6))
```

```
sns.scatterplot(data=df_encoded, x="PC1", y="PC2", hue="cluster", palette="Set2", s=80)
plt.title("Кластери клієнтів (PCA + KMeans)")
plt.grid(True)
plt.show()

# 7. Аналіз — розподіл класів всередині кожного кластера
cluster_class_dist = df_encoded.groupby("cluster")["class"].value_counts(normalize=True).unstack().round(2)
cluster_class_dist.columns = ["bad", "good"]

# 8. Візуалізація цього аналізу
cluster_class_dist.plot(kind="bar", stacked=True, colormap="coolwarm", figsize=(8,5))
plt.title("Частка good/bad у кожному кластері")
plt.ylabel("Частка клієнтів")
plt.xlabel("Кластер")
plt.legend(title="Клас")
plt.grid(axis='y')
plt.show()

# 9. Якщо потрібно — зберегти кластеризований датасет
df_encoded.to_csv("clusterized_credit_data.csv", index=False)
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

```
# Определяем параметры для перебора
param_grid = {
    'random_state': list(range(1, 1001)),
    'n_estimators': [100],
    'max_depth': [10]
}

# Создаем модель
model = RandomForestClassifier()
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test= train_test_split(X,y,
                                    test_size=0.2,
                                    random_state=42)

from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score

# XGBoost без кластеризації
xgb_model = XGBClassifier(
    n_estimators=100,
    max_depth=10,
    learning_rate=0.1,
    random_state=4242,
```

```
    objective='binary:logistic',
    eval_metric='logloss'
)

xgb_model.fit(X_train, y_train)
y_pred_xgb = xgb_model.predict(X_test)

# Обчислення метрик
acc_xgb = accuracy_score(y_test, y_pred_xgb)
f1_xgb = f1_score(y_test, y_pred_xgb, average="weighted")
precision_xgb = precision_score(y_test, y_pred_xgb)
recall_xgb = recall_score(y_test, y_pred_xgb)

# Вивід результатів
print(f"✅ XGBoost без кластеризації:")
print(f"Accuracy:  {acc_xgb:.4f}")
print(f"F1-score:  {f1_xgb:.4f}")
print(f"Precision: {precision_xgb:.4f}")
print(f"Recall:    {recall_xgb:.4f}")


from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score

# Ініціалізація моделі
model = LogisticRegression(solver='saga', max_iter=13000)  # додатково — для стабільності
model.fit(X_train, y_train)
```

```
# Передбачення
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Accuracy
train_accuracy = accuracy_score(y_train, y_train_pred)
test_accuracy_log_reg = accuracy_score(y_test, y_test_pred)

# Інші метрики (на test)
f1_log_reg = f1_score(y_test, y_test_pred, average="weighted")
precision_log_reg = precision_score(y_test, y_test_pred)
recall_log_reg = recall_score(y_test, y_test_pred)

# Вивід метрик
print("✅ Logistic Regression без кластеризації:")
print(f"Train Accuracy : {train_accuracy:.4f}")
print(f"Test Accuracy  : {test_accuracy_log_reg:.4f}")
print(f"F1-score       : {f1_log_reg:.4f}")
print(f"Precision      : {precision_log_reg:.4f}")
print(f"Recall         : {recall_log_reg:.4f}")


from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier

param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [5, 10, 15, 20],
    'min_samples_split': [2, 5, 10, 20],
    'min_samples_leaf': [1, 2, 4, 8],
```

```
    'class_weight': ['balanced', None]
}

rf = RandomForestClassifier(random_state=42)
grid_search = GridSearchCV(rf, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)


print("Найкращі параметры:", grid_search.best_params_)
print("Найкраща точность:", grid_search.best_score_)


from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score

# Ініціалізація моделі
best_model = RandomForestClassifier(
    class_weight='balanced',
    max_depth=10,
    min_samples_leaf=2,
    min_samples_split=10,
    n_estimators=100,
    random_state=4242
)

# Навчання моделі
best_model.fit(X_train, y_train)
```

```
# Прогноз
y_pred = best_model.predict(X_test)

# Розрахунок метрик
test_accuracy = accuracy_score(y_test, y_pred)
f1_rf = f1_score(y_test, y_pred, average="weighted")
precision_rf = precision_score(y_test, y_pred)
recall_rf = recall_score(y_test, y_pred)

# Вивід результатів
print("✅ Random Forest без кластеризації:")
print(f"Accuracy :  {test_accuracy:.4f}")
print(f"F1-score :  {f1_rf:.4f}")
print(f"Precision:  {precision_rf:.4f}")
print(f"Recall   :  {recall_rf:.4f}")



import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Таблиця з усіма метриками для трьох моделей (без кластеризації)
results_nocluster = pd.DataFrame({
    "Model": ["Logistic Regression", "Random Forest", "XGBoost"],
    "Accuracy": [test_accuracy_log_reg,test_accuracy,acc_xgb],
    "F1-score": [f1_log_reg, f1_rf, f1_xgb],
    "Precision": [precision_log_reg, precision_rf, precision_xgb],
    "Recall": [recall_log_reg, recall_rf, recall_xgb]
})
```

```
# Побудова графіка
fig, ax = plt.subplots(figsize=(10,6))
x = np.arange(len(results_nocluster["Model"]))
width = 0.2

rects1 = ax.bar(x - 1.5*width, results_nocluster["Accuracy"], width, label='Accuracy')
rects2 = ax.bar(x - 0.5*width, results_nocluster["F1-score"], width, label='F1-score')
rects3 = ax.bar(x + 0.5*width, results_nocluster["Precision"], width, label='Precision')
rects4 = ax.bar(x + 1.5*width, results_nocluster["Recall"], width, label='Recall')

ax.set_ylabel('Значення метрик')
ax.set_title('Порівняння метрик для моделей (без кластеризації)')
ax.set_xticks(x)
ax.set_xticklabels(results_nocluster["Model"])
ax.legend(loc='upper left', bbox_to_anchor=(1.02, 1))



# Підписи значень
for rect in rects1 + rects2 + rects3 + rects4:
    height = rect.get_height()
    ax.annotate(f'{height:.3f}', xy=(rect.get_x() + rect.get_width()/2, height),
                xytext=(0,3), textcoords="offset points", ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.ensemble import RandomForestClassifier

# Навчаємо модель
rfc = RandomForestClassifier(n_estimators=100, random_state=42)
rfc.fit(X_train, y_train)

# Отримуємо важливість ознак
importances = rfc.feature_importances_
feature_names = X_train.columns

# Сортуємо за важливістю
indices = np.argsort(importances)

plt.figure(figsize=(8, 6))
plt.barh(range(len(importances)), importances[indices], align="center",color='g')
plt.yticks(range(len(importances)), feature_names[indices])
plt.xlabel("Важливість")
plt.ylabel("Ознаки")
plt.title("Feature Importance - RandomForest")
plt.show()



import xgboost as xgb
import matplotlib.pyplot as plt
import numpy as np

# Обучаем модель
xgb_model = xgb.XGBClassifier(n_estimators=100, random_state=42)
```

```
xgb_model.fit(X_train, y_train)

# Получаем важность признаков
importances = xgb_model.feature_importances_
feature_names = X_train.columns

# Сортируем признаки по важности
indices = np.argsort(importances)

plt.figure(figsize=(8, 6))
plt.barh(range(len(importances)), importances[indices], align="center",color='r')
plt.yticks(range(len(importances)), feature_names[indices])
plt.xlabel("Важливість")
plt.ylabel("Ознаки")
plt.title("Feature Importance - XGBoost")
plt.show()



import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

# Навчаємо модель
logreg = LogisticRegression(max_iter=1000, random_state=42)
logreg.fit(X_train, y_train)

# Отримуємо коефіцієнти моделі
coefficients = logreg.coef_[0]  # Для бінарної класифікації
feature_names = X_train.columns
```

```
# Отримуємо абсолютні значення коефіцієнтів
abs_coefficients = np.abs(coefficients)

# Сортуємо ознаки за величиною коефіцієнтів
indices = np.argsort(abs_coefficients)

# Будуємо графік
plt.figure(figsize=(10, 6))
plt.barh(range(len(abs_coefficients)), abs_coefficients[indices], align="center")
plt.yticks(range(len(abs_coefficients)), feature_names[indices])
plt.xlabel("Різниця в впливі ознаки")
plt.ylabel("Ознаки")
plt.title("Feature Importance (Magnitude) - Logistic Regression")
plt.show()



import numpy as np
import matplotlib.pyplot as plt

# Дані
models = ["Логістична регресія", "Random Forest", "XGBoost"]
accuracy = [0.76, 0.81, 0.79]
f1_scores = [0.7472, 0.8126, 0.7837]

# Параметри для групування стовпців
x = np.arange(len(models))  # Позиції для кожної моделі
width = 0.35  # Ширина стовпчиків

# Побудова графіка
fig, ax = plt.subplots(figsize=(8, 6))
```

```
bars1 = ax.bar(x - width/2, accuracy, width, label='Accuracy', color='royalblue')
bars2 = ax.bar(x + width/2, f1_scores, width, label='F1-score', color='crimson')

# Додаємо значення на стовпчики
for bars in [bars1, bars2]:
    for bar in bars:
        yval = bar.get_height()
        ax.text(bar.get_x() + bar.get_width()/2, yval, f"{yval:.3f}",
                ha='center', va='bottom', fontsize=11, fontweight='bold')

# Налаштування осей
ax.set_ylim(0.7, 0.85)  # Починаємо з 0.7 для кращої видимості різниці
ax.set_ylabel("Значення метрики", fontsize=14)
ax.set_title("Порівняння Accuracy та F1-score для різних моделей", fontsize=15, fontweight='bold')
ax.set_xticks(x)
ax.set_xticklabels(models, fontsize=12)
ax.legend(fontsize=12)  # Легенда

# Відображення графіка
plt.show()


import seaborn as sns
import matplotlib.pyplot as plt

correlation_matrix = X_train.corr()

plt.figure(figsize=(12, 10))
```

```
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f',
            annot_kws={"size": 0}, linewidths=0.3, linecolor='gray', square=True)
plt.title("Кореляційна матриця ознак ", fontsize=16)
plt.show()



df_encoded["cluster"] = df_encoded["cluster"].astype("int64")
df["cluster"] = df_encoded["cluster"]  # або df["cluster"] = kmeans.labels_

X = df.drop("class", axis=1)  # включає cluster_id
y = df["class"]

df.info()

from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder

# Видаляємо PCA компоненти (вони тільки для візуалізації, не для моделі)
X = df_encoded.drop(columns=["class", "PC1", "PC2"])
y = df_encoded["class"]

# Переведемо всі categorical → int
```

```
for col in X.columns:
    if str(X[col].dtype) == "category":
        X[col] = X[col].cat.codes
    elif str(X[col].dtype) == "object":
        X[col] = LabelEncoder().fit_transform(X[col])

# Train/Test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Словник моделей
models = {
    "Logistic Regression": LogisticRegression(solver='saga', max_iter=13000),
        "Random Forest": RandomForestClassifier(class_weight='balanced', max_depth=10, min_samples_leaf=2, min_samples_split=10, n_estimators=100, random_state=4242),
     "XGBoost": XGBClassifier(objective='binary:logistic', eval_metric='logloss', n_estimators=100, max_depth=10, learning_rate=0.1, random_state=42)
    }

# Збір метрик
results = {"Model": [], "Accuracy": [], "F1-score": [], "Precision": [], "Recall": []}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    results["Model"].append(name)
    results["Accuracy"].append(round(accuracy_score(y_test, y_pred), 3))
    results["F1-score"].append(round(f1_score(y_test, y_pred), 3))
```

```
    results["Precision"].append(round(precision_score(y_test, y_pred), 3))
    results["Recall"].append(round(recall_score(y_test, y_pred), 3))

# Таблиця результатів
results_df = pd.DataFrame(results)
print(results_df)

# Візуалізація всіх метрик
fig, ax = plt.subplots(figsize=(10,6))
x = np.arange(len(results_df["Model"]))
width = 0.2

rects1 = ax.bar(x - 1.5*width, results_df["Accuracy"], width, label='Accuracy')
rects2 = ax.bar(x - 0.5*width, results_df["F1-score"], width, label='F1-score')
rects3 = ax.bar(x + 0.5*width, results_df["Precision"], width, label='Precision')
rects4 = ax.bar(x + 1.5*width, results_df["Recall"], width, label='Recall')

ax.set_ylabel('Значення метрик')
ax.set_title('Порівняння Accuracy, F1-score, Precision та Recall моделей')
ax.set_xticks(x)
ax.set_xticklabels(results_df["Model"])
ax.legend(loc='upper left', bbox_to_anchor=(1.02, 1))

# Додавання підписів значень над стовпцями
for rect in rects1 + rects2 + rects3 + rects4:
    height = rect.get_height()
    ax.annotate(f'{height:.3f}', xy=(rect.get_x() + rect.get_width()/2, height),
                xytext=(0,3), textcoords="offset points", ha='center', va='bottom')

plt.tight_layout()
```

```
plt.show()


from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt

# Вибір усіх нових числових та інженерних ознак
selected_features = [
    'credit_utilization', 'credit_to_income_ratio', 'cluster',
    'financial_status', 'log_credit_amount', 'installment_to_income',
    'long_term_loan', 'stable_borrower', 'young_borrower',
    'old_borrower', 'has_guarantor', 'has_phone'
]

# Формуємо X і y
X_new = df[selected_features]
y = df['class']

# Поділ на train/test
X_train_new, X_test_new, y_train, y_test = train_test_split(X_new, y, test_size=0.2, random_state=42)

# Тренуємо логістичну регресію
logreg = LogisticRegression(max_iter=1000)
logreg.fit(X_train_new, y_train)

# Прогноз ймовірностей
y_proba = logreg.predict_proba(X_test_new)[:, 1]
```

```
# Побудова ROC-кривої
fpr, tpr, thresholds = roc_curve(y_test, y_proba)
auc_score = roc_auc_score(y_test, y_proba)

plt.figure()
plt.plot(fpr, tpr, label=f'ROC-крива (AUC = {auc_score:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC-крива з новими ознаками')
plt.legend()
plt.grid(True)
plt.show()



import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

fig, ax = plt.subplots(figsize=(12, 7))
ax.axis('off')

# Нові координати — збільшуємо відстань між блоками
blocks = [
    (0.08, 0.8, '1. Завантаження\nта очищення даних'),
    (0.4, 0.8, '2. Обробка ознак\n(масштабування, кластеризація)'),
    (0.72, 0.8, '3. Навчання моделей\n(LogReg, RF, XGB)'),
    (0.4, 0.5, '4. Пояснення рішень\n(важливість ознак, графіки)'),
    (0.4, 0.2, '5. Експорт результатів\nта інтеграція')
]
```

```
# Малювання блоків
for x, y, text in blocks:
    ax.add_patch(mpatches.FancyBboxPatch((x, y), 0.25, 0.12,
        boxstyle="round,pad=0.03", edgecolor="black", facecolor="#cce5ff"))
    ax.text(x + 0.125, y + 0.06, text, ha='center', va='center', fontsize=9)

# Стрілки між блоками
arrows = [
    ((0.05 + 0.25, 0.86), (0.4, 0.86)),
    ((0.4 + 0.25, 0.86), (0.75, 0.86)),
    ((0.525, 0.8), (0.525, 0.62)),
    ((0.525, 0.5), (0.525, 0.32)),
]

for start, end in arrows:
    ax.annotate('', xy=end, xytext=start,
                arrowprops=dict(facecolor='black', arrowstyle='->'))

plt.title("Архітектура системи підтримки рішень з ML", fontsize=13)
plt.show()
```

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kravchenko D. Credit scoring with machine learning: Master thesis. – Dnipro University of Technology, 2023. – 72 p.

2. Безруков О.О. Виявлення шахрайських транзакцій з допомогою методів машинного навчання: кваліфікаційна робота магістра. – Тернопіль: ТНТУ ім. І. Пулюя, 2023. – 70 с.

3. Ткачик О.А. Методи та засоби кластеризації різнотипних даних: дис. ... д-ра філософії: 122 – Комп'ютерні науки / Нац. ун-т «Львівська політехніка». – Львів, 2023. – 149 с.

4. Соломоник О.П. Застосування XGBoost для скорингу в системі кредитного аналізу: магістерська робота. – Кривий Ріг: КДПУ, 2025. – 86 с.

5. Medvedieva I.B. Stress testing of the bank's credit portfolio by macroeconomic parameters // Науковий вісник Миколаївського національного університету ім. В.О. Сухомлинського. – 2017. – Вип. 16. – С. 752–754.

6. Покатаєва О.В., Славкіна М.А. Оцінювання системного ризику як інструмент забезпечення економічної безпеки банківського сектору // Науковий вісник Ужгородського національного університету. – 2019. – Вип. 23, ч. 1. – С. 157–158.

7. Popovych B. Application of AI in Credit Scoring Modeling. – Springer Gabler, 1st ed., 2022. – 236 p.

8. Louzada F., Ara A., Fernandes G.B. Classification methods applied to credit scoring: Systematic review and overall comparison // Computational Economics. – 2016. – Vol. 49, no. 4. – P. 913–940.

9. Mulla G.A.A., Demir Y. The Use of Clustering and Classification Methods in Machine Learning and Comparison of Some Algorithms of the Methods // Cihan University-Erbil Scientific Journal. – 2023. – Vol. 7, no. 1. – P. 52–59.

10. Lending development strategy 2024 [Electronic resource] – Mode of access: https://bank.gov.ua/documents/strategy/lending_2024.pdf

11. Андрушків Б.І. Особливості використання скорингових систем у банківському кредитуванні фізичних осіб // Економіка і регіон. – 2023. – № 1. – С. 101–105.

12. Investigation of the scoring model for bank borrowers [Electronic resource] – Mode of access: https://papers.ssrn.com/abstract=3890977

13. Accelerating Machine Learning as a Service with Automated Feature Engineering. – IBM Research, 2022. – 20 p.

14. DEVELOPMENT OF DATA MESH DATA PLATFORM WITH ML DOM. – White Paper. – 2022. – 12 p.

15. GJETA-2024-0029. Advances in Credit Risk Analysis and Modeling. – Global Journal of Emerging Technologies and Analytics. – 2024. – №2. – P. 1–15.

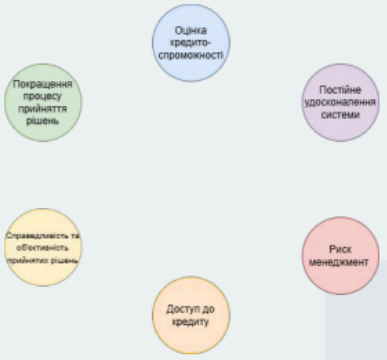
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ: ПРЕЗЕНТАЦІЯ

Система підтримки прийняття рішень для оцінки кредитного ризику з використанням машинного навчання

Цей проект спрямований на розробку системи підтримки прийняття рішень, спрямованої на посилення оцінки кредитного ризику за рахунок впровадження методів машинного навчання.

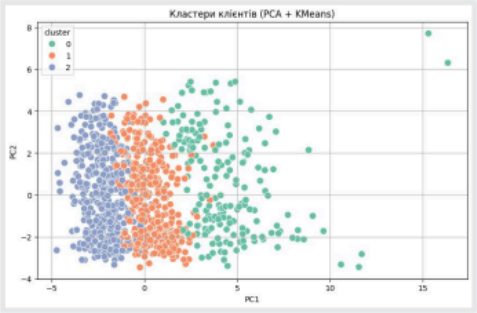
Виконує: Качан Нікіта Андрійович
Керівник: Кузьміч Михайло Юрійович

Актуальність теми



Важливість кредитного скорингу: Кредитний скоринг служить основним інструментом для фінансових установ для оцінки кредитоспроможності потенційних позичальників.

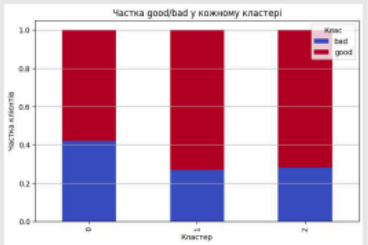
- Ризики вирощування:
  - Економічна нестабільність призвела до зростання ризиків у практиці кредитування.
  - Фінансові установи стикаються з проблемами точної оцінки кредитного ризику.
- Потреба в автоматизації:
  - Існує нагальна потреба в автоматизації рішень кредитного скорингу для підвищення як точності, так і швидкості.
  - Впровадження машинного навчання (ML) може значно зменшити вплив людської упередженості на процеси прийняття рішень.



Мета і завдання роботи

Мета: Основною метою даного дослідження є аналіз та вдосконалення методів оцінки кредитного ризику за допомогою машинного навчання.

- Цілей:
  - Проаналізувати сучасні методи кредитного скорингу.
  - Впроваджувати моделі машинного навчання, включаючи:
    - Логістична регресія
    - Random Forest
    - XGBoost
  - Проведення кластеризації (K Means) як частина інженерії об'єктів.
  - Порівнюйте моделі на основі ключових показників ефективності.



Об'єкт і предмет дослідження

Об'єкт дослідження-

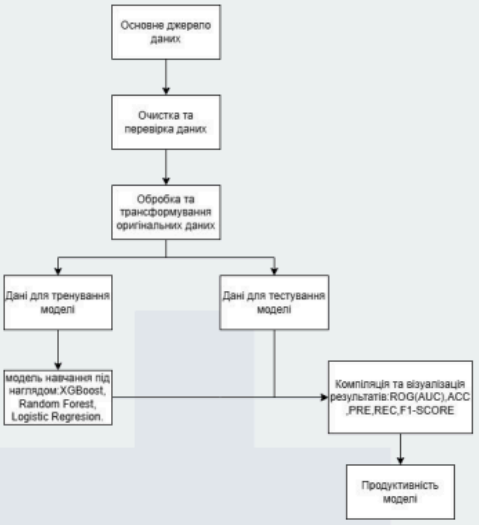
Об'єктом дослідження є процес оцінки кредитоспроможності клієнтів, який використовується фінансовими установами.

Предмет дослідження-

Предмет зосереджений на застосуванні алгоритмів машинного навчання для підвищення ефективності та точності рішень з оцінки кредитного ризику.

Методика дослідження

- Попередня обробка даних:
  - Масштабування: нормалізація даних для забезпечення однорідності.
  - Заповнення прогалин: усунення відсутніх значень у наборі даних.
  - Кодування: перетворення категорійних змінних у числовий формат.
- Особливості інжинірингу:
  - Методи кластеризації (K Means) використовуються для удосконалення вибору та вилучення ознак.
- Модельне навчання:
  - Навчання моделей з використанням підготовленого набору даних та оцінка їх ефективності.
- Метрики оцінки:
  - Ключові показники включають:
    - Точність
    - Recall
    - Precision
    - Оцінка F-1



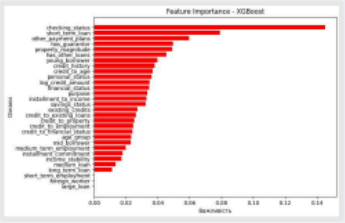
Архітектура системи

Джерела даних

До джерел даних належать: - Анкети, що заповнюються заявниками. - Дані з бюро кредитних історій. - Поведінкові ознаки, що свідчать про кредитоспроможність.

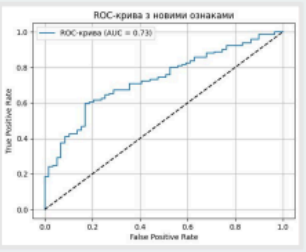
Процес обробки

Процес обробки складається з: - Обробки даних- Future engineering - Кластеризації - Побудови моделі.



Графічна візуалізація результатів

Графічна візуалізація результатів реалізована у вигляді графіків порівнянь, теплових карт та діаграм .



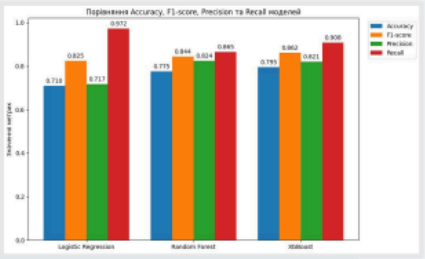
Вживані моделі

Моделі машинного навчання:

- Логістична регресія: базова модель для задач бінарної класифікації.
- Random Forest: ансамблевий метод, який покращує точність прогнозування за допомогою кількох дерев рішень.
- XGBoost: Вдосконалений алгоритм посилення, відомий своєю високою продуктивністю та ефективністю.

Ключовий показник:

- Особлива увага приділяється Recall як критично важливому показнику для виявлення ризикованих клієнтів, гарантуючи, що буде позначено якомога більше потенційних ризиків.



Результати дослідження

15%

Удосконалення виклику за допомогою кластеризації KMeans з моделлю XGBoost.

ROC-криві

що ілюструють продуктивність моделі.

Матриця плутанини

Матриця плутанини для детальних результатів класифікації.

Зменшення помилок

Зменшення помилок класифікації, підвищення загальної надійності моделі.

Практична значущість

Переваги впровадження:

- Впровадження цих моделей значно зменшує кількість неправильних кредитних рішень, що приймаються фінансовими установами.
- Підвищення точності в оцінці кредитного ризику призводить до вдосконалення практики кредитування.
- Автоматизація скорингових процесів оптимізує роботу в банківських системах.
- Отримані результати можуть бути застосовані для вирішення реальних бізнес-завдань, підвищення ефективності та прибутковості.

Висновки

Узагальнений виклад висновків

Сучасні моделі машинного навчання демонструють чудову продуктивність у порівнянні з традиційними методами кредитного скорингу.
Методи кластеризації помітно підвищують точність систем оцінки кредитного ризику.
Застосування алгоритмів XGBoost забезпечує оптимальний баланс між викликом і точністю.
Основні результати дослідження були апробовані у формі тез, поданих до публікації на студентській науковій конференції в межах секції «№5» Державного університету телекомунікацій.
Конференція проходила в онлайн-форматі.
Представлені матеріали засвідчили наукову цінність і практичну значущість отриманих результатів.

Майбутні роботи

Це дослідження має практичну цінність і потенціал для розширення, відкриваючи шлях для подальших досліджень в області оцінки кредитних ризиків з використанням передових методів машинного навчання.