

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.89

До захисту допущено
Завідувач кафедри ММСА
_____ Оксана ТИМОЩУК
«___» _____ 2025 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Системний аналіз фінансового ринку»
за спеціальністю 124 «Системний аналіз»
на тему: «Інтелектуальна система підтримки прийняття рішень для аналізу
економічних ризиків»

Виконав:

Студент 2 курсу, групи КА-42мп
Мельник Микола Сергійович _____

Науковий керівник:

старший викладач кафедри ММСА, PhD
Левенчук Людмила Борисівна _____

Консультант з нормоконтролю:

Доцент, к.ф.-м.н.,
Статкевич Віталій Михайлович _____

Рецензент:

Доцент кафедри ШІ, PhD
Гуськова Віра Геннадіївна _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань
Студент: _____

Київ – 2025

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)

Спеціальність — 124 «Системний аналіз»

Освітньо-професійною програмою «Системний аналіз фінансового ринку»

ЗАТВЕРДЖУЮ

Завідувач кафедри ММСА

_____ Оксана ТИМОЩУК

« ____ » _____ 2025 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Мельнику Миколі Сергійовичу

1. Тема дисертації: «Інтелектуальна система підтримки прийняття рішень для аналізу економічних ризиків», науковий керівник дисертації старший викладач кафедри ММСА, PhD Левенчук Людмила Борисівна, затверджені наказом по університету від 06 листопада 2025 року №4837-с.
2. Строк подання студентом дисертації: 10.12.2025 р.
3. Об'єкт дослідження: процес аналізу економічних ризиків підприємств та механізми оцінювання їхньої фінансової стійкості.
4. Предмет дослідження: методи, моделі та алгоритми машинного навчання, а також програмні засоби їх застосування в інтелектуальній системі підтримки прийняття рішень для прогнозування ризику банкрутства.
5. Перелік завдань, які потрібно зробити:
 - 1) проаналізувати проблемну область і вимоги;
 - 2) підготувати фінансові дані;
 - 3) збалансувати вибірку;
 - 4) виконати відбір ознак;
 - 5) навчити моделі ML;
 - 6) порівняти моделі за метриками;

- 7) інтегрувати SHAP для інтерпретації;
 - 8) спроектувати архітектуру ІСППР;
 - 9) реалізувати веб-систему;
 - 10) провести тестування та валідацію;
 - 11) проаналізувати результати та зробити висновки.
6. Перелік графічного (ілюстративного) матеріалу:
- 1) зразок вхідних даних (рис. 3.1);
 - 2) співвідношення банкрутів у даних (рис. 3.2);
 - 3) кореляційна матриця (рис. 3.3);
 - 4) графік викидів (рис. 3.4);
 - 5) обрані ознаки для моделей (рис. 3.5 – 3.7);
 - 6) результати навчання окремих моделей (рис. 3.8 – 3.19);
 - 7) порівняння результатів навчання моделей (рис. 3.20 – 3.21);
 - 8) функціональна схема ІСППР (рис. 3.22);
 - 9) огляд інтерфейсу та функціоналу ІСППР (рис. 3.23 – 3.35)
7. Орієнтовний перелік публікацій:
- 1) Мельник М.С., Левенчук Л.Б. Інтелектуальна система підтримки прийняття рішень для аналізу економічних ризиків. Системні науки та інформатика: збірник доповідей IV Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 01-05 грудня 2025 року, Київ. К., НН ІПСА КПІ ім. Ігоря Сікорського, 2025, 6 с.
8. Дата видачі завдання: 01 вересня 2025 року

Календарний план

№ з/п	Назва етапів виконання Магістерської дисертації (МД)	Строк виконання етапів магістерської дисертації	Примітка
1	Затвердження теми МД. Ознайомлення зі структурою МД згідно з Положенням про державну атестацію студентів НТУУ «КПІ ім. І. Сікорського». Ознайомлення з ДСТУ 3008:2015.	02.09.2025-10.09.2025	виконано
2	Робота над першим розділом МД. Дослідження предметної області. Огляд літератури за обраною тематикою. Узагальнення знайденої інформації.	11.09.2025-18.09.2025	виконано
3	Робота над другим розділом МД. Вибір методів реалізації поставленої задачі МД, аналіз їх математичних основ.	19.09.2025-01.10.2025	виконано
4	Робота над третім розділом МД. Розробка програмного продукту для розв'язання поставленої задачі МД.	02.10.2025-14.10.2025	виконано
5	Робота над третім розділом МД. Застосування розробленого програмного продукту для оцінки ефективності його роботи, аналіз отриманих результатів.	15.10.2025-30.10.2025	виконано
6	Робота над четвертим розділом МД. Розробка власного стартап-проекту.	31.10.2025-11.11.2025	виконано
7	Формування висновків з огляду на результати виконаної роботи.	12.11.2025-20.11.2025	виконано
8	Оформлення магістерської дисертації.	21.11.2025-01.12.2025	виконано

Студент

Науковий керівник дисертації

Микола МЕЛЬНИК

Людмила ЛЕВЕНЧУК

РЕФЕРАТ

Магістерська дисертація: 138 с., 35 рис., 22 табл., 1 дод., 29 джерел.

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ,
АНАЛІЗ ЕКОНОМІЧНИХ РИЗИКІВ, ПРОГНОЗУВАННЯ БАНКРУТСТВА,
МАШИННЕ НАВЧАННЯ, XGBOOST, SHAP, PYTHON.

Мета дослідження – розробка інтелектуальної системи підтримки прийняття рішень (ІСППР) для аналізу економічних ризиків, яка на основі сучасних методів машинного навчання забезпечує якісне та інтерпретоване прогнозування банкрутства підприємств.

Об’єкт дослідження – процес аналізу економічних ризиків підприємств та механізми оцінювання їхньої фінансової стійкості.

Предмет дослідження – методи, моделі та алгоритми машинного навчання, а також програмні засоби їх застосування в інтелектуальній системі підтримки прийняття рішень для прогнозування ризику банкрутства.

В дисертації було проведено системний аналіз проблеми оцінки економічних ризиків та огляд методів машинного навчання для прогнозування банкрутства. Обґрунтовано використання ансамблевих моделей (зокрема XGBoost) у поєднанні з методами балансування даних (SMOTE) та інтерпретації результатів (SHAP). Розроблено архітектуру та програмну реалізацію веб-орієнтованої ІСППР «RiskSense AI», що дозволяє виконувати експрес-діагностику фінансового стану підприємств та отримувати пояснення факторів ризику.

В додатку А наведено лістинг розробленої програми мовою Python з використанням бібліотек Pandas, Scikit-Learn, XGBoost, SHAP, Streamlit та інших.

ABSTRACT

Master's Thesis: 138 p., 35 figs., 22 tables, 1 appendix, 29 references.

INTELLIGENT DECISION SUPPORT SYSTEM, ECONOMIC RISK ANALYSIS, BANKRUPTCY PREDICTION, MACHINE LEARNING, XGBOOST, SHAP, PYTHON.

The aim of the study is to develop an intelligent decision support system (IDSS) for analyzing economic risks, which provides high-quality and interpretable bankruptcy prediction of enterprises based on modern machine learning methods.

The object of the study is the process of analyzing economic risks of enterprises and mechanisms for assessing their financial stability.

The subject of the study covers models, methods, and algorithms of machine learning, as well as software tools for their application in an intelligent decision support system for bankruptcy prediction.

The thesis conducts a system analysis of the economic risk assessment problem and reviews machine learning methods for bankruptcy prediction. The use of ensemble models (specifically XGBoost) in combination with data balancing methods (SMOTE) and result interpretation (SHAP) is substantiated. The architecture and software implementation of the web-oriented IDSS «RiskSense AI» have been developed, allowing for rapid diagnostics of an enterprise's financial state and providing explanations of risk factors.

Appendix A contains the code listing of the developed program in Python using libraires such as Pandas, Scikit-Learn, XGBoost, SHAP, Streamlit and others.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	9
ВСТУП	10
РОЗДІЛ 1 ТЕОРЕТИЧНІ ЗАСАДИ АНАЛІЗУ ЕКОНОМІЧНИХ РИЗИКІВ ТА ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ	13
1.1 Актуальність та стан наукової проблеми	13
1.2 Аналіз джерел і сучасних досліджень	15
1.2.1 Класичні моделі оцінки банкрутства	16
1.2.2 Еволюція кредитного скорингу та машинного навчання	16
1.2.3 Українські дослідження та практики	18
1.3 Підходи та інструменти машинного навчання для аналізу ризиків	20
1.3.1 Лінійні та узагальнені лінійні моделі	20
1.3.2 Дерева рішень та ансамблеві методи	21
1.3.3 Методи роботи з незбалансованими даними	22
1.3.4 Інтерпретація моделей та Explainable AI.....	23
1.3.5 Порівняльна оцінка підходів	23
1.4 Проблематика та формулювання наукової задачі дослідження	25
1.5 Постановка задачі	26
1.6 Висновки до розділу 1	28
РОЗДІЛ 2 ОГЛЯД МОДЕЛЕЙ ТА МЕТОДІВ ЇХ ОЦІНЮВАННЯ.....	30
2.1 Попередня обробка та аналіз даних	30
2.1.1 Методологія балансування класів: Алгоритм SMOTE	30
2.1.2 Рекурсивне виключення ознак (RFE)	32
2.2 Огляд моделей машинного навчання.....	32
2.2.1 Дерево рішень (Decision Tree)	33
2.2.2 Випадковий ліс (Random Forest)	34
2.2.3 Метод опорних векторів (Support Vector Machine — SVM)	35
2.2.4 Градієнтний бустинг (Gradient Boosting)	36
2.2.5 Екстремальний градієнтний бустинг (XGBoost)	37

2.3 Інтелектуальна система підтримки прийняття рішень	40
2.3.1 Аналіз природи проблемної області	41
2.3.2 Підсистема управління даними	42
2.3.3 Підсистема управління моделями	43
2.3.4 Підсистема знань та інтелектуального аналізу	44
2.3.5 Підсистема інтерфейсу користувача	45
2.3.6 Критеріальний аналіз інтелектуальності системи	45
2.4 Метрики оцінювання адекватності побудованих моделей	48
2.4.1 Матриця помилок та похідні метрики	48
2.4.2 Інтегральні метрики: F1 та AUC-ROC	49
2.5 Інтерпретація результатів моделей (SHAP)	49
2.6 Висновки до розділу 2	50
РОЗДІЛ 3 ПОБУДОВА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ТА РОЗРОБКА ІСППР	52
3.1 Аналіз та підготовка вхідних даних	52
3.2 Відбір значущих характеристик	57
3.3 Побудова моделей та порівняння результатів	63
3.4 Розробка ІСППР	74
3.5 Висновки до розділу 3	85
РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП ПРОЕКТУ	88
4.1 Опис ідеї стартап-проекту	89
4.2 Технологічний аудит ідеї проекту	91
4.3 Аналіз ринкових можливостей запуску стартап-проекту	95
4.4 Розроблення ринкової стратегії стартап-проекту	104
4.5 Розроблення маркетингової програми стартап-проекту	107
4.6 Висновки до розділу 4	109
ВИСНОВКИ	110
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	112
ДОДАТОК А. ЛІСТИНГУ КОДУ	116

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

БД	База даних
ІСППР	Інтелектуальна система підтримки прийняття рішень
МН	Машинне навчання (Machine Learning)
НБУ	Національний банк України
ПЗ	Програмне забезпечення
СППР	Система підтримки прийняття рішень
API	Application Programming Interface (Інтерфейс прикладного програмування)
AUC	Area Under Curve (Площа під кривою)
FN	False Negative (Помилка другого роду / пропуск події)
FP	False Positive (Помилка першого роду / хибне спрацьовування)
IQR	Interquartile Range (Міжквартильний розмах)
NPL	Non-performing loans (Непрацюючі кредити)
RFE	Recursive Feature Elimination (Рекурсивне виключення ознак)
ROC	Receiver Operating Characteristic (Робоча характеристика приймача)
SaaS	Software as a Service (Програмне забезпечення як послуга)
SHAP	SHapley Additive exPlanations (Адитивні пояснення Шеплі)
SMOTE	Synthetic Minority Over-sampling Technique (Техніка синтетичного надсемплінгу меншості)
SVM	Support Vector Machine (Метод опорних векторів)
TN	True Negative (Істинно негативний результат)
TP	True Positive (Істинно позитивний результат)
XGBoost	eXtreme Gradient Boosting (Екстремальний градієнтний бустинг)

ВСТУП

Актуальність теми. Сучасна економіка України функціонує в умовах екстремальної невизначеності, спричиненої воєнними діями та геополітичною напруженістю. Забезпечення фінансової стійкості підприємств вимагає надійних інструментів раннього попередження криз. Традиційні методи діагностики банкрутства (моделі Альтмана, Таффлера), що базуються на лінійних залежностях, втрачають прогностичну ефективність в умовах високої волатильності та структурних змін ринку.

Водночас, розвиток методів машинного навчання відкриває нові можливості для аналізу великих масивів фінансових даних. Проте широке впровадження ML-алгоритмів у ризик-менеджмент стримується проблемою їх низької інтерпретованості («black box»). Тому актуальним науково-прикладним завданням є розробка інтелектуальної системи підтримки прийняття рішень (ІСППР), яка б поєднувала високу точність ансамблевих методів (XGBoost) з прозорістю результатів завдяки технологіям пояснюваного штучного інтелекту (Explainable AI), забезпечуючи довіру до автоматизованих рішень.

Мета і завдання дослідження – розробити інтелектуальну систему підтримки прийняття рішень для аналізу економічних ризиків, яка на основі сучасних методів машинного навчання забезпечує якісне та інтерпретоване прогнозування банкрутства підприємств.

Для досягнення мети поставлено такі завдання:

- 1) проаналізувати проблемну область і вимоги;
- 2) підготувати фінансові дані;
- 3) збалансувати вибірку;
- 4) виконати відбір ознак;
- 5) навчити моделі ML;
- 6) порівняти моделі за метриками;

- 7) інтегрувати SHAP для інтерпретації;
- 8) спроектувати архітектуру ІСППР;
- 9) реалізувати веб-систему;
- 10) провести тестування та валідацію;
- 11) проаналізувати результати та зробити висновки.

Об’єкт дослідження – процес аналізу економічних ризиків підприємств та механізми оцінювання їхньої фінансової стійкості.

Предмет дослідження – методи, моделі та алгоритми машинного навчання, а також програмні засоби їх застосування в інтелектуальних системах підтримки прийняття рішень для прогнозування ризику банкрутства.

Методи дослідження. У роботі використано: системний аналіз – для проєктування архітектури системи; методи машинного навчання (ансамблеві методи, SVM) – для побудови прогностичних моделей; статистичні методи (SMOTE, RFE) – для обробки даних та усунення дисбалансу; методи Explainable AI (SHAP) – для інтерпретації результатів моделювання.

Наукова новизна одержаних результатів:

1. Набуло подальшого розвитку застосування методу градієнтного бустингу (XGBoost) для фінансового скорингу, що, на відміну від існуючих підходів, передбачає комплексну стратегію балансування даних (SMOTE), забезпечуючи високу точність (AUC ~ 0.85) на незбалансованих вибірках.

2. Удосконалено структуру СППР шляхом інтеграції модуля пояснень Шеплі (SHAP), що дозволяє кількісно оцінити вплив кожного фінансового показника на рівень ризику, забезпечуючи прозорість прийняття рішень.

Практичне значення одержаних результатів – розроблено діючий прототип веб-орієнтованої системи «RiskSense AI» для експрес-діагностики ймовірності банкрутства. Система рекомендована для використання у відділах ризик-менеджменту фінансових установ та аудиторських компаніях.

У **першому розділі** висвітлено теоретичні засади аналізу економічних ризиків та функціонування систем підтримки прийняття рішень. Проведено аналіз еволюції методів оцінювання фінансової стійкості від класичних

моделей до сучасних алгоритмів машинного навчання, а також обґрунтовано доцільність використання ансамблевих методів та технологій Explainable AI для розв'язання поставленої наукової задачі.

Другий розділ присвячено методології дослідження та огляду математичного апарату. Детально розглянуто методи попередньої обробки даних, зокрема алгоритми балансування класів та відбору ознак. Описано принципи роботи обраних моделей машинного навчання та визначено метрики для оцінювання їхньої ефективності й інтерпретованості.

У **третьому розділі** викладено результати практичної реалізації дослідження. Проведено підготовку даних, навчання та порівняльний аналіз моделей, за результатами якого обґрунтовано вибір алгоритму XGBoost. Описано архітектуру та функціональні можливості розробленої інтелектуальної системи підтримки прийняття рішень, яка реалізована у вигляді веб-застосунку з модулем візуалізації пояснень SHAP.

Четвертий розділ містить розробку власного стартап-проєкту «RiskSense AI». Здійснено маркетинговий аналіз ринку та конкурентного середовища, проведено технологічний аудит ідеї, а також сформовано ринкову стратегію та обґрунтовано економічну доцільність комерціалізації розробленого продукту.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ЗАСАДИ АНАЛІЗУ ЕКОНОМІЧНИХ РИЗИКІВ ТА ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

1.1 Актуальність та стан наукової проблеми

Економічні ризики є невід’ємною характеристикою функціонування підприємств і фінансових інститутів у сучасній глобальній економіці. Воєнні дії, структурні зрушення в енергетичному секторі, посилення геополітичної напруженості, цикли ужорсткішення монетарної політики провідних центробанків створюють середовище підвищеної невизначеності, у якому класичні інструменти ризик-менеджменту дедалі частіше виявляються недостатніми. В умовах високої невизначеності критично важливим є застосування методів попередньої обробки даних та ймовірнісного прогнозування, що дозволяють зменшити вплив стохастичної природи економічних процесів на якість управлінських рішень [27]. Згідно із Звітом про фінансову стабільність НБУ, ключовими загрозами для української фінансової системи у 2023–2024 рр. залишалися кредитний ризик, концентрація портфелів та висока чутливість до зовнішніх шоків, що вимагає від банківських установ поглибленого аналізу платоспроможності позичальників та якості активів [1].

Паралельно відбувається якісна трансформація фінансових ринків. Цифровізація призводить до появи нових бізнес-моделей (фінтех-платформи, р2р-кредитування, краудфандинг), збільшення швидкості обігу капіталу та зростання обсягів даних, що генеруються в режимі реального часу. Фінансові інститути оперують гігантськими масивами транзакційних, поведінкових та нефінансових даних. Це створює як нові можливості для якіснішого вимірювання ризиків, так і нові виклики — зокрема, необхідність

автоматизації аналітичних процесів та забезпечення прозорості прийнятих рішень.

Традиційні методи аналізу ризиків — дисперсійно-коваріаційні моделі, лінійні регресії, класичні скорингові карти — історично забезпечили основу для формування системи ризик-менеджменту. Прикладом є дискримінантні моделі оцінювання ймовірності банкрутства на кшталт Z-показника Альтмана [2] та підходи, систематизовані в роботі Чибісової, де виділено чотири групи методів: експертні, економіко-математичні, штучні інтелектуальні системи та методи оцінки фінансового стану [3].

Разом з тим, у сучасних умовах такі моделі мають низку фундаментальних обмежень:

1. Лінійність та жорсткі припущення. Більшість класичних моделей передбачає лінійні залежності між фінансовими показниками та ймовірністю дефолту, нормальність розподілів і стаціонарність процесів. На емпіричних даних фінансових ринків ці припущення часто не виконуються, що призводить до зміщених оцінок ризику.

2. Чутливість до вибору ознак і параметрів. Дискримінантні та регресійні моделі мають обмежену здатність працювати з великою кількістю корельованих або нелінійно пов'язаних факторів; відбір змінних у них значною мірою спирається на експертні судження [2–4].

3. Складність роботи з незбалансованими даними. Для задач прогнозування банкрутства чи дефолту частка «проблемних» позичальників у вибірці, як правило, невелика. Статистичні моделі, оптимізовані за загальною якістю, схильні ігнорувати рідкісний клас ризикових об'єктів [4,5].

4. Низька гнучкість щодо оновлення. У середовищі, де профілі клієнтів і макроекономічні умови швидко змінюються, моделі, побудовані на «статичних» уявленнях про ризик, втрачають адекватність.

Окремим викликом є регуляторні вимоги до прозорості та інтерпретованості моделей оцінки кредитного ризику. Банки повинні пояснювати причини відмови в кредитуванні, обґрунтовувати рівень резервів

і демонструвати стабільність моделей за різних сценаріїв стрес-тестування. Це ускладнює використання складних «чорних скриньок», які хоча й забезпечують вищу якість прогнозування, але є малозрозумілими для регуляторів та менеджменту.

В українському контексті додаткову актуальність проблеми зумовлюють:

- 1) структурні зрушення в реальному секторі внаслідок війни та релокації виробництв;
- 2) швидке зростання частки цифрових каналів обслуговування та дистанційних продуктів, для яких історичні дані обмежені;
- 3) посилення вимог НБУ до процедур оцінки кредитоспроможності та стрес-тестування банківських портфелів [1,7].

У цих умовах виникає потреба в інтелектуальних системах підтримки прийняття рішень (ІСППР), здатних поєднати:

- 1) високу прогнозну здатність сучасних алгоритмів машинного навчання;
- 2) механізми пояснення рішень моделі (Explainable AI);
- 3) інтеграцію у бізнес-процеси ризик-менеджменту банків та небанківських фінансових установ.

Такі системи мають не лише оцінювати ймовірність негативних подій (банкрутство, дефолт, різка втрата ліквідності), а й надавати користувачам структуровану інформацію про фактори, що призвели до конкретного прогнозу, підтримуючи прозорість та відтворюваність управлінських рішень.

1.2 Аналіз джерел і сучасних досліджень

Наукові дослідження, присвячені аналізу економічних ризиків та застосуванню інтелектуальних систем для їх оцінки, можна умовно поділити на кілька груп: класичні моделі банкрутства, еволюція кредитного скорингу, сучасні машинні методи в оцінці ризику та українські дослідження й практики.

1.2.1 Класичні моделі оцінки банкрутства

Початок кількісного аналізу ризику неплатоспроможності пов'язують із роботою Альтмана [2], який запропонував багатофакторну дискримінантну модель на основі фінансових коефіцієнтів. Незважаючи на обмеження (лінійність, чутливість до галузевих особливостей), Z-показник став еталонним інструментом для раннього виявлення фінансових проблем підприємств і активно використовується в дослідженнях та практиці.

Чибісова узагальнює підходи до оцінки й прогнозування банкрутства, виділяючи експертні методи, економіко-математичні моделі, штучні інтелектуальні системи та процедури аналізу фінансового стану [3]. Авторка показує, що жодна з існуючих методик не є універсальною, а ефективна діагностика потребує комбінування кількох підходів, зокрема використання нейронних мереж та експертних систем для виявлення складних нелінійних залежностей.

Ці роботи заклали теоретичну основу для подальшого розвитку інтелектуальних систем аналізу ризику, однак не враховують особливостей великих, гетерогенних масивів даних та вимог до інтерпретованості сучасних моделей.

1.2.2 Еволюція кредитного скорингу та машинного навчання

Систематичний аналіз класифікаційних алгоритмів у кредитному скорингу подано в дослідженні Лессмана із співавторами [4]. На основі восьми реальних наборів даних автори порівняли 41 алгоритм (логістична регресія, дерева рішень, нейронні мережі, ансамблеві методи тощо) за шістьма показниками якості. Показано, що гетерогенні ансамблеві моделі (зокрема,

градієнтний бустинг і випадкові ліси) статистично значимо перевищують за якістю традиційну логістичну регресію, яка залишається промисловим стандартом. Це дослідження продемонструвало доцільність переходу від класичних статистичних до машинних моделей у задачах кредитного скорингу.

Барбоза, Кімура та Альтман [5] фокусуються на задачі прогнозування банкрутства компаній за ринковими й бухгалтерськими даними. Автори порівнюють SVM, bagging, boosting, random forest з дискримінантним аналізом і логістичною регресією, показуючи суттєве зростання якості за рахунок ансамблевих методів. Цікаво, що у низці експериментів саме бустингові алгоритми демонструють найкращий баланс помилки I та II роду, що є критичним для кредиторів та інвесторів.

У більш нових роботах акцент зміщується з простої якості до інтерпретованості моделей. Нгуєн, Вівіані та Джабер [9] пропонують комплексний підхід до прогнозування банкрутства французьких компаній з горизонтом від одного до п'яти років, поєднуючи ансамблеві моделі (Random Forest, LightGBM, XGBoost, NGBoost) з методом Shapley Additive Explanations (SHAP) для пояснення прогнозів. Отримано, що XGBoost і NGBoost забезпечують найвищу якість, а SHAP дозволяє ідентифікувати ключові драйвери ризику на рівні окремих компаній.

Біттето та співавтори [8] досліджують кредитний ризик малих та середніх підприємств, порівнюючи традиційний впорядкований пробіт-моделі із історичним random forest. Показано, що динамічна модель Historical Random Forest істотно поліпшує прогнозну здатність та краще враховує часову структуру даних. Важливим результатом роботи є використання значень Шеплі для інтерпретації внеску кожної змінної в оцінку ризику, що наближає машинні моделі до вимог регуляторів щодо прозорості.

1.2.3 Українські дослідження та практики

Українські наукові праці фіксують необхідність адаптації сучасних методів машинного навчання до специфіки національної фінансової системи. Мірошніченко та Івлієва [6] розглядають застосування алгоритмів машинного навчання для оцінювання кредитного ризику на даних українського банку. Автори демонструють, що використання методів відбору ознак (інформаційна цінність, коефіцієнт V Крамера) та логістичної регресії з попередньою категоризацією змінних дозволяє побудувати скорингову модель з високою прогностичною здатністю, при цьому зберігаючи інтерпретованість параметрів.

Коломієць і Кочорба [7] пропонують концептуальну модель оцінки кредитного ризику в системі ризик-менеджменту банківських структур, що поєднує етапи формування інформаційного простору, скорингової оцінки позичальника та прийняття рішення. У роботі демонструється застосування методів машинного навчання (логістичної регресії, дерева рішень, ансамблевих моделей) для скорингу клієнтів банку та розробляється механізм вибору порогових значень скорингового бала з урахуванням співвідношення доходів і втрат.

Вагомий внесок у розвиток методології моделювання ризиків зроблено в роботах Л. Б. Левенчук [28, 29]. У цих дослідженнях обґрунтовано принципи побудови систем підтримки прийняття рішень для аналізу операційних та фінансових ризиків, а також запропоновано підходи до ймовірнісного моделювання, які дозволяють враховувати приховані залежності у складних системах.

На макрорівні НБУ у Звітах про фінансову стабільність [1] підкреслює необхідність посилення методології оцінки кредитного ризику, зокрема для портфелів, чутливих до шоків, та гармонізації українського регулювання з нормами ЄС (стрес-тестування, буфери капіталу, моделі внутрішніх

рейтингів). Це створює запит на інструменти, які можуть бути інтегровані в процеси регуляторної звітності та внутрішніх моделей банків.

1.2.4 Методи та інструменти машинного навчання

Важливе місце в сучасних дослідженнях посідають роботи, присвячені алгоритмічним аспектам. Чен і Гуестрін [10] запропонували XGBoost — високоефективну реалізацію градієнтного бустингу над деревами рішень, орієнтовану на обробку великих табличних наборів даних із пропущеними значеннями та складними взаємодіями між ознаками. Саме XGBoost став одним із де-факто стандартів для задач кредитного скорингу і прогнозування банкрутства [4,5,8].

Лундберг і Лі [11] представили підхід SHAP, що базується на значеннях Шеплі з теорії кооперативних ігор і дає змогу одержати як глобальні, так і локальні пояснення рішень складних моделей. Поєднання градієнтного бустингу з SHAP сьогодні розглядається як одна з найперспективніших стратегій побудови інтерпретованих моделей кредитного ризику [8,9].

Нарешті, Леметр, Ногейра та Арідас [12] у статті про бібліотеку `imbalanced-learn` демонструють, як методи надсемплування (зокрема SMOTE та його модифікації) дозволяють підвищити чутливість класифікаторів до рідкісного класу, що має принципове значення для задач виявлення дефолтів і банкрутств.

Загалом аналіз літератури показує, що:

1. Поєднання tree-based ансамблевих моделей (Random Forest, XGBoost) з методами балансування класів та інтерпретації рішень (SHAP) забезпечує найкраще співвідношення якості та пояснюваності.

2. У більшості робіт досліджується або кредитний ризик (скоринг фізичних осіб, МСП), або банкрутство підприємств; інтегровані системи, що

дозволяють конфігурувати модель під різні типи економічних ризиків у єдиному середовищі, залишаються недостатньо опрацьованими;

3. Питання перенесення результатів на український ринок (особливості звітності, вплив воєнного фактору, регуляторні обмеження) висвітлено фрагментарно [1,6,7].

Ці прогалини обґрунтовують необхідність розробки інтелектуальної системи підтримки прийняття рішень, яка поєднувала б сучасні машинні методи, інструменти інтерпретації та адаптацію до умов українського фінансового сектору.

1.3 Підходи та інструменти машинного навчання для аналізу ризиків

Задачі аналізу економічних ризиків у більшості випадків формуються як задачі класифікації (наприклад, віднесення позичальника до класів «надійний / ризиковий», підприємства — до «банкрут / небанкрут») або регресії (оцінка ймовірності дефолту, очікуваних збитків). У цьому підрозділі узагальнено основні підходи машинного навчання, орієнтовані на роботу з табличними фінансовими даними.

1.3.1 Лінійні та узагальнені лінійні моделі

Логістична регресія залишається базовим інструментом кредитного скорингу завдяки простоті реалізації, інтерпретованості коефіцієнтів та відносній стабільності [4,6]. Вона дозволяє безпосередньо оцінювати ймовірність дефолту, а монотонність зв'язку між коефіцієнтами та ризиком полегшує бізнес-інтерпретацію. Проте лінійність зв'язку між логіт-функцією

та ознаками обмежує здатність моделі відображати складні нелінійні залежності, а чутливість до мультиколінеарності потребує попереднього відбору ознак та/або їх трансформації [3,6].

Методи регуляризації (L1/L2, elastic net) частково пом'якшують проблему перенавчання й дозволяють автоматично відсікати несуттєві змінні. Однак у задачах із сильно вираженою нелінійністю та взаємодією ознак (наприклад, коли ризик істотно залежить від поєднання галузі, структури капіталу і динаміки грошових потоків) лінійні моделі програють нелінійним алгоритмам [4,5].

1.3.2 Древа рішень та ансамблеві методи

Древа рішень (Decision Tree) є одним із найпоширеніших інструментів завдяки інтуїтивній зрозумілості: шлях від кореня до листка можна трактувати як набір логічних правил. Вони добре працюють з різнотипними ознаками й не потребують масштабування даних. Основні недоліки — схильність до перенавчання та нестабільність щодо невеликих змін у вибірці.

Ансамблеві методи покликані подолати ці обмеження. Random Forest агрегує велику кількість слабо корельованих дерев, побудованих на різних підвибірках ознак та спостережень, що знижує дисперсію оцінок. Градієнтний бустинг (Gradient Boosting Machines) послідовно навчає дерева на залишкових помилках попередніх моделей, концентруючись на «важких» об'єктах.

Дослідження Лесман [4] та Барбоза [5] показують, що ансамблі дерев рішень у більшості кредитних задач суттєво перевищують за якістю як логістичну регресію, так і окремі дерева, особливо у випадку багатовимірних даних.

XGBoost [10] є високопродуктивною реалізацією градієнтного бустингу, яка:

- 1) реалізує регуляризовану цільову функцію, що зменшує перенавчання;
- 2) підтримує обробку пропусків та різні типи втрат;
- 3) оптимізована для розподілених та паралельних обчислень.

Завдяки цим властивостям XGBoost став одним із ключових алгоритмів у практичних системах скорингу й прогнозування банкрутства [5,8,9].

1.3.3 Методи роботи з незбалансованими даними

Незбалансованість даних (коли частка дефолтів/банкрутств у вибірці мала) призводить до того, що моделі, оптимізовані за загальною якістю, можуть досягати високих значень асигасу, практично ігноруючи клас «ризик». Для вирішення цієї проблеми використовують:

- 1) методи ресемплінгу: надсемпсування міноритарного класу (SMOTE та його варіації), undersampling мажоритарного класу, комбіновані підходи;
- 2) змінені функції втрат: cost-sensitive learning, де помилки на міноритарному класі штрафуються сильніше;
- 3) спеціальні метрики якості: ROC AUC, F1-міра, Precision-Recall AUC.

Леметр [12] демонструє, що застосування методів типу SMOTE у поєднанні з ансамблевими алгоритмами дозволяє істотно підвищити чутливість до рідкісних подій без критичного зростання кількості хибних спрацьовувань. Це особливо важливо в задачах, що розглядаються в даній дисертації, де клас «банкрут» суттєво менший за клас «стабільні компанії».

1.3.4 Інтерпретація моделей та Explainable AI

Одним із ключових аргументів проти використання складних алгоритмів у фінансах тривалий час була їхня «непрозорість». Сучасна парадигма Explainable AI частково розв’язує цю проблему. Метод SHAP [11] надає:

1) глобальні пояснення: ранжування ознак за їхнім середнім внеском у передбачення моделі;

2) локальні пояснення: розклад прогнозу для конкретного об’єкта на суму внесків окремих ознак, що дозволяє пояснити, чому, наприклад, підприємство віднесено до високоризикової групи.

У роботах Нгуєн [9] та Біттето [8] показано, що поєднання XGBoost із SHAP дозволяє одночасно досягти високої якості прогнозу й забезпечити пояснюваність на рівні, достатньому для прийняття управлінських рішень та спілкування з регуляторами.

1.3.5 Порівняльна оцінка підходів

На основі аналізу джерел та властивостей алгоритмів можна виокремити такі узагальнені висновки, наведені у таблиці 1.1.

Таблиця 1.1. Порівняння алгоритмів

Алгоритм	Переваги	Недоліки
Логістична регресія	Простота, інтерпретованість, можливість калібрування ймовірностей, добра інтеграція в існуючі процеси банків.	Слабке відображення нелінійних взаємодій, чутливість до мультиколінеарності, потреба у ретельній підготовці ознак.
Дерева рішень	Наочність, здатність працювати з різнотипними змінними, відсутність вимог до нормальності розподілів.	Нестабільність, нижча якість порівняно з ансамблями.
Random Forest, Gradient Boosting, XGBoost	Висока якість, здатність моделювати складні нелінійні залежності, відносна стійкість до викидів та мультиколінеарності.	Більш висока обчислювальна складність, потреба у налаштуванні гіперпараметрів, обмежена «прозорість» без додаткових засобів інтерпретації.

Методи балансування (SMOTE, cost-sensitive learning) критично важливі для задач з рідкісними подіями, дозволяють уникнути ситуації, коли модель «навчається» завжди передбачати відсутність ризику.

З огляду на це, для розробки інтелектуальної системи підтримки прийняття рішень, орієнтованої на аналіз економічних ризиків, доцільно використовувати tree-based ансамблеві моделі (насамперед XGBoost) у поєднанні з методами балансування класів і пояснювальними техніками SHAP. Саме така комбінація є фокусом подальших розділів дисертації.

1.4 Проблематика та формулювання наукової задачі дослідження

Проведений аналіз дозволяє сформулювати низку невирішених проблем, які визначають наукову задачу даної магістерської дисертації.

Фрагментарність застосування машинного навчання. Більшість робіт зосереджується або на кредитному скорингу фізичних осіб, або на прогнозуванні банкрутства юридичних осіб [4–6,8,9]. Відсутні інтегровані рішення, які дозволяли б у єдиному середовищі налаштовувати моделі для різних типів економічних ризиків, використовуючи уніфіковану архітектуру даних і спільний інтерфейс для ризик-менеджерів.

Проблеми незбалансованості та якості даних. Реальні набори фінансових даних характеризуються відсутніми значеннями, шумом, «довгими хвостами» розподілів, а класи «ризик» є значно меншими за клас «норми». Хоча існують методи ресемплінгу [12], їх інтеграція у виробничі системи потребує врахування обмежень щодо стабільності моделей та можливого викривлення розподілу ризиків.

Інтерпретованість і прозорість рішень. Використання складних ансамблевих моделей (XGBoost, Random Forest) без механізмів пояснення суперечить вимогам до прозорості кредитних рішень і може викликати недовіру з боку менеджменту, регуляторів та клієнтів. Потрібні архітектурні рішення, які «вбудовують» explainable AI (зокрема SHAP) безпосередньо у процес прийняття рішень [8,9,11].

Відсутність інтегрованої ІСППР на рівні програмно-технологічної реалізації. У багатьох роботах акцент робиться на побудові окремих моделей, а не на створенні завершеної системи, яка забезпечує повний цикл: завантаження й перевірка даних, попередня обробка, балансування класів, навчання й валідація моделей, візуалізація ризиків та генерація рекомендацій для користувача.

З урахуванням наведеного, наукову задачу дослідження можна сформулювати так: Розробити концептуальні та алгоритмічні засади інтелектуальної системи підтримки прийняття рішень для аналізу економічних ризиків, яка на основі сучасних методів машинного навчання (зокрема ансамблевих моделей градієнтного бустингу) та інструментів Explainable AI забезпечує високоякісне, інтерпретоване й стійке до незбалансованості даних прогнозування ризику банкрутства підприємств.

Для вирішення цієї задачі в подальших розділах дисертації буде:

- 1) сформовано вимоги до інформаційної структури даних та функціональної архітектури ІСППР;
- 2) обґрунтовано вибір алгоритмів машинного навчання та методів балансування класів;
- 3) розроблено підхід до інтеграції SHAP-пояснень у інтерфейс користувача й процедури ризик-менеджменту;
- 4) проведено експериментальне порівняння моделей на реальних або квазіреальних фінансових даних з оцінкою якості прогнозів та інтерпретованості результатів.

1.5 Постановка задачі

Виходячи з аналізу стану наукової проблеми та сформульованої мети роботи, для розробки ефективної інтелектуальної системи підтримки прийняття рішень (ІСППР) необхідно вирішити комплекс взаємопов'язаних завдань, що охоплюють етапи від підготовки даних до програмної реалізації та валідації результатів.

Загальну задачу дослідження декомпозовано на такі підетапи:

1. **Проаналізувати проблемну область і вимоги.** Здійснити формалізацію вимог до системи на основі аналізу специфіки оцінювання

фінансових ризиків в умовах економічної нестабільності, визначити ключові функціональні та нефункціональні вимоги до ІСППР.

2. Підготувати фінансові дані. Виконати збір, очищення та попередню обробку історичних фінансових даних підприємств, включаючи обробку пропущених значень, нормалізацію та стандартизацію вхідних показників (X1–X18).

3. Збалансувати вибірку. Розв'язати проблему дисбалансу класів («alive»/«failed») шляхом застосування методів синтетичного семплінгу (зокрема, алгоритму SMOTE) для підвищення чутливості моделей до виявлення рідкісних кризових подій.

4. Виконати відбір ознак. Зменшити розмірність простору ознак та усунути мультиколінеарність за допомогою методів рекурсивного виключення (RFE) та аналізу важливості ознак для підвищення узагальнюючої здатності моделей.

5. Навчити моделі машинного навчання. Побудувати та навчити низку моделей класифікації, включаючи деревоподібні ансамблеві методи (Random Forest, Gradient Boosting, XGBoost) та метод опорних векторів (SVM).

6. Порівняти моделі за метриками. Виконати порівняльний аналіз ефективності побудованих моделей з використанням метрик, стійких до дисбалансу класів (Precision, Recall, F1-score, ROC AUC) , та обрати оптимальний алгоритм для подальшої імплементації.

7. Інтегрувати SHAP для інтерпретації. Реалізувати модуль пояснюваного штучного інтелекту (Explainable AI) на основі значень Шеплі (SHAP) для забезпечення прозорості прийняття рішень та визначення внеску кожного фінансового показника у прогнозований ризик.

8. Спроекувати архітектуру ІСППР. Розробити функціональну схему та архітектуру системи, що включає модулі керування даними, моделювання, візуалізації та інтерфейсу користувача.

9. Реалізувати веб-систему. Створити програмний прототип ІСППР у вигляді веб-застосунку з використанням мови Python та відповідних бібліотек.

10. Провести тестування та валідацію. Здійснити перевірку працездатності розробленої системи на тестових даних, оцінити адекватність роботи інтерфейсу та коректність генерації прогнозів і пояснень.

11. Проаналізувати результати та зробити висновки. Узагальнити отримані теоретичні та практичні результати, оцінити наукову новизну та практичну цінність розробленої системи для завдань ризик-менеджменту.

Реалізація наведених етапів дозволить створити цілісний інструмент, здатний забезпечити високу точність прогнозування банкрутства у поєднанні з інтерпретованістю результатів, що є критичною вимогою для фінансового сектору.

1.6 Висновки до розділу 1

Обґрунтовано актуальність теми дисертації. Показано, що зростання невизначеності та волатильності фінансового середовища, цифровізація ринків і посилення регуляторних вимог роблять завдання високоякісного й прозорого аналізу економічних ризиків критично важливим для банківських та небанківських інститутів, особливо в умовах української економіки після початку повномасштабної війни.

Проаналізовано класичні та сучасні наукові підходи. Розглянуто еволюцію від дискримінантних моделей банкрутства до сучасних машинних алгоритмів кредитного скорингу та банкрутства, а також українські дослідження, які адаптують ці підходи до національної специфіки.

Систематизовано інструменти машинного навчання для аналізу ризиків. Узагальнено переваги й недоліки лінійних моделей, дерев рішень та ансамблевих методів; обґрунтовано доцільність використання XGBoost у поєднанні з методами боротьби з незбалансованістю (SMOTE, cost-sensitive learning) та інструментами Explainable AI (SHAP).

Виявлено наукові прогалини. Сформульовано проблеми, пов'язані з фрагментарним застосуванням машинного навчання, недостатньою адаптацією моделей до українського контексту, труднощами роботи з незбалансованими даними та необхідністю інтерпретованості моделей. Підкреслено відсутність інтегрованих інтелектуальних систем підтримки прийняття рішень, орієнтованих на повний цикл аналізу економічних ризиків.

Сформульовано наукову задачу дисертації. Вона полягає у розробці інтелектуальної системи підтримки прийняття рішень для аналізу економічних ризиків, яка поєднує сучасні ансамблеві методи машинного навчання, засоби балансування класів і модулі інтерпретації, забезпечуючи практичну придатність для ризик-менеджменту українських фінансових установ.

Деталізовано постановку задачі дослідження шляхом її декомпозиції на послідовні етапи реалізації: від попередньої обробки та балансування фінансових даних до програмної імплементації та валідації веб-системи. Визначено, що досягнення мети роботи потребує комплексного підходу, який поєднує глибокий інжиніринг ознак, порівняльний аналіз ML-моделей за спеціалізованими метриками та обов'язкову інтеграцію модуля пояснень (SHAP). Такий алгоритм дій забезпечує створення завершеного продукту (ІСППР), здатного не лише прогнозувати банкрутство, а й надавати інтерпретовану аналітичну підтримку для прийняття управлінських рішень.

Отримані в цьому розділі результати створюють методологічну основу для подальшого проєктування структури даних, архітектури ІСППР та експериментальної перевірки її ефективності.

РОЗДІЛ 2 ОГЛЯД МОДЕЛЕЙ ТА МЕТОДІВ ЇХ ОЦІНЮВАННЯ

2.1 Попередня обробка та аналіз даних

Розробка інтелектуальної системи підтримки прийняття рішень (ІСППР) для аналізу економічних ризиків та прогнозування банкрутства підприємств вимагає фундаментального підходу до роботи з даними. У сучасній науці про дані етап попередньої обробки та розвідувального аналізу (Exploratory Data Analysis — EDA) визнається не допоміжним, а визначальним фактором, що встановлює верхню межу ефективності будь-якої прогностичної моделі [4,5]. Як зазначається у фундаментальних працях з системного аналізу [26], якість вхідних даних, їхня інформативність та репрезентативність є критичними умовами для побудови адекватної математичної моделі, незалежно від обраного методу машинного навчання. Специфіка фінансових даних, які характеризуються високою волатильністю, різномірністю шкал вимірювання, наявністю латентних зв'язків та критичним дисбалансом класів, зумовлює необхідність застосування складних алгоритмічних процедур ще до моменту навчання моделей.

2.1.1 Методологія балансування класів: Алгоритм SMOTE

Для вирішення проблеми дисбалансу класів у дослідженні застосовано метод SMOTE (Synthetic Minority Over-sampling Technique) [13]. Цей алгоритм став «золотим стандартом» у галузі навчання на незбалансованих даних.

На відміну від примітивного методу випадкового пересемплювання (Random Oversampling), який просто дублює існуючі приклади меншості, SMOTE генерує нові, синтетичні екземпляри. Просте дублювання часто

призводить до звуження областей прийняття рішень навколо конкретних точок, що викликає перенавчання. SMOTE ж працює в просторі ознак, створюючи нові точки вздовж лінійних сегментів, що з'єднують будь-який/всі k найближчих сусідів класу меншості [12,13].

Алгоритм можна описати наступною послідовністю кроків:

1. Для кожного вектору x_i з класу меншості знаходяться k найближчих сусідів серед об'єктів того ж класу, використовуючи евклідову відстань.
2. Обирається випадковим чином один із знайдених сусідів, позначимо його $\hat{x}_i$.
3. Обчислюється різниця між вектором ознак поточного об'єкта та сусіда: $\delta = \hat{x}_i - x_i$.
4. Генерується новий синтетичний приклад x_{new} шляхом додавання до x_i цієї різниці, помноженої на випадкове число λ в інтервалі (0. 1):

$$x_{new} = x_i + \lambda (\hat{x}_i - x_i)$$

Цей процес дозволяє «заповнити» простір між існуючими прикладами позитивного класу, роблячи границі рішень більш гладкими та узагальненими. У дослідженні використання SMOTE дозволило штучно збільшити представництво класу «failed», забезпечивши моделям (особливо чутливим до дисбалансу, таким як SVM) достатню кількість інформації для формування коректних розділяючих гіперплощин. Важливість цього кроку підкреслюється тим, що без застосування SMOTE моделі демонстрували б зміщення у бік мажоритарного класу, що неприпустимо для задач прогнозування кризових явищ. Використання бібліотеки imblearn (imbalanced-learn), зазначеної у переліку залежностей проекту, підтверджує технічну реалізацію цього підходу.

2.1.2 Рекурсивне виключення ознак (RFE)

Зважаючи на виявлену мультиколінеарність, для оптимізації набору предикторів було обрано метод Recursive Feature Elimination (RFE). Цей підхід належить до категорії методів-обгорток (wrapper methods) [14].

RFE базується на ідеї ітеративного видалення найменш важливих ознак. Процес виглядає наступним чином [14]:

1. Навчається модель (estimator) на повному наборі ознак.
2. Для кожної ознаки обчислюється показник важливості. Для лінійних моделей (наприклад, Linear SVM) це може бути квадрат ваги ознаки, для дерев рішень — показник зменшення домішок (Gini importance).
3. Ознака (або група ознак) з найменшим показником важливості видаляється з набору даних.
4. Процедура повторюється на зменшеному наборі ознак до тих пір, поки не буде досягнуто бажаної кількості змінних.

Використання RFE дозволяє не просто відкинути корельовані ознаки, а знайти таку підмножину змінних, яка максимізує прогностичну здатність конкретної моделі. Це відрізняє RFE від методів фільтрації, які оцінюють ознаки ізольовано від моделі. У контексті даної роботи RFE дозволив зменшити розмірність простору ознак, усунувши «шумові» змінні та ті, що дублюють інформацію, залишаючи лише найбільш значущі індикатори фінансового стану.

2.2 Огляд моделей машинного навчання

Вибір архітектури моделей є ключовим етапом у проектуванні ІСППР. У дослідженні проведено порівняльний аналіз п'яти сучасних алгоритмів:

Дерево рішень (Decision Tree), Випадковий ліс (Random Forest), Метод опорних векторів (SVM), Градієнтний бустинг (Gradient Boosting) та XGBoost. Кожен з цих методів має унікальні теоретичні властивості, переваги та обмеження, які детально розглядаються нижче.

2.2.1 Дерево рішень (Decision Tree)

Дерево рішень (Decision Tree) є фундаментальним алгоритмом, який моделює процес прийняття рішень у вигляді деревоподібної структури графів [15].

Метод CART (Classification and Regression Trees) будує бінарні дерева шляхом рекурсивного розбиття простору ознак на прямокутні області. На кожному кроці алгоритм жадібно обирає ознаку x_j та поріг t , які забезпечують найкращий поділ множини даних на дві підмножини ($x_j < t$ та $x_j > t$). Критерієм оптимальності розбиття в задачах класифікації виступає мінімізація невизначеності (домішок) [15].

У даному дослідженні використовується Індекс Джині (Gini Impurity) як критерій розбиття, що є стандартом для алгоритму CART (на відміну від C4.5, що використовує ентропію). Для вузла t індекс Джині визначається як ймовірність некоректної класифікації випадково обраного елемента з цього вузла, якщо його мітка обирається згідно з розподілом класів у вузлі [15]:

$$Gini = 1 - \sum_{i=1}^n p_i^2$$

де p_i — ймовірність класу i . Алгоритм прагне знайти розбиття, яке мінімізує зважену суму індексів Джині дочірніх вузлів.

Альтернативою є використання Ентропії Шеннона:

$$Entropy = - \sum_{i=1}^n p_i \log_2(p_i)$$

Ентропія досягає максимуму при рівномірному розподілі класів і мінімуму (0), коли вузол містить об'єкти лише одного класу. Максимізація приросту інформації еквівалентна мінімізації ентропії.

Головною перевагою дерев рішень є їх інтерпретованість. Модель можна візуалізувати, а правила («Якщо Поточні активи < X І Чистий прибуток < Y, ТО Банкрут») зрозумілі фінансовим аналітикам. Проте цей метод має суттєвий недолік — схильність до перенавчання. Поодинокі дерева, особливо глибокі, мають високу дисперсію і будують надмірно складні границі рішень, підлаштовуючись під шум у навчальній вибірці.

2.2.2 Випадковий ліс (Random Forest)

Для вирішення проблеми перенавчання та зниження дисперсії поодиноких дерев був винайдений метод Випадкового лісу (Random Forest). Цей алгоритм базується на концепції беггінгу (Bagging — Bootstrap Aggregating) [16].

Суть методу полягає у створенні ансамблю з B дерев рішень $\{T_1, T_2, \dots, T_B\}$. Щоб забезпечити незалежність (декореляцію) дерев, Random Forest вносить два елементи випадковості:

1. Row Sampling (Bootstrap): кожне дерево T_B навчається на власній бутстреп-вибірці D_B , яка формується шляхом вибору N прикладів з початкового набору даних з поверненням. При цьому близько 36.8% даних не

потрапляють у вибірку для конкретного дерева (Out-of-Bag samples, OOB) і можуть використовуватися для внутрішньої валідації.

2. Feature Sampling (Random Subspace Method): при побудові кожного розбиття в дереві розглядається не повний набір з M ознак, а лише випадкова підмножина з $m < M$ ознак (зазвичай $m \approx \sqrt{M}$). Це запобігає ситуації, коли одна-дві сильні ознаки домінують у всіх деревах, роблячи їх схожими одне на одне.

Фінальний прогноз ансамблю для класифікації отримується шляхом голосування більшості. Таке усереднення дозволяє значно знизити дисперсію моделі без суттєвого збільшення зміщення.

2.2.3 Метод опорних векторів (Support Vector Machine — SVM)

Метод опорних векторів (SVM) представляє собою принципово інший підхід, що базується на геометричній інтерпретації даних. Основна ідея SVM полягає у знаходженні гіперплощини, яка розділяє класи з максимально можливим зазором (margin) [17].

Нехай навчальна вибірка складається з пар (x_i, y_i) , де $y_i \in \{-1, 1\}$. Лінійна розділяюча функція має вигляд $f(x) = w^T x + b$. Задача навчання SVM формулюється як задача квадратичного програмування: знайти w та b , які мінімізують норму ваг (максимізують зазор $\frac{2}{\|w\|}$) при умові правильної класифікації:

$$\min_{w, b, \xi} \|w\|_2^2 + C \sum_{i=1}^n \xi_i$$

при обмеженнях:

$$y_i(w^T x_i - b) \geq 1 - \xi_i, \xi_i \geq 0 \quad \forall i \in \{1, \dots, n\}$$

Тут ξ_i — змінні слабкості (slack variables), які дозволяють допускати певні помилки класифікації (soft margin) у випадку лінійно нероздільних даних, а C — параметр регуляризації, що керує компромісом між шириною зазору та сумою помилок.

Для вирішення нелінійних задач (що є типовим для економічних даних) застосовується «ядровий трюк» (Kernel Trick). Замість явного перетворення вхідних векторів у простір високої розмірності $\phi(x)$, алгоритм використовує функцію ядра $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, яка обчислює скалярний добуток у просторі ознак без явного відображення. Це дозволяє будувати складні нелінійні границі рішень.

2.2.4 Градієнтний бустинг (Gradient Boosting)

Градієнтний бустинг представляє собою клас алгоритмів, які будують ансамбль послідовно, а не паралельно. На відміну від Random Forest, де дерева незалежні, у бустингу кожна нова модель намагається виправити помилки попередньої комбінації моделей [18].

Нехай маємо датасет $D = \{(x_i, y_i)\}_{i=1}^n$, де $x_i \in X$ — вектор ознак, $y_i \in Y$ — ціль. Задача — побудувати функцію $F(x)$ (предиктор), що мінімізує очікуване значення функції втрат $L(y, F(x))$. Мережна (адитивна) модель градієнтного бустингу задається як послідовне додавання базових моделей $h_m(x)$, зазвичай слабких регресійних дерев:

$$F(x) = F_0(x) + \sum_{m=1}^M \gamma_m h_m(x)$$

де $F_0(x)$ — початкова константна апроксимація (наприклад, аргумент мінімуму сумарної втрати по всіх y_i), $\gamma_m \in R$ — коефіцієнти масштабування (можуть бути знайдені лінійним пошуком), а M — кількість ітерацій/дерев. Основна ідея — виконувати градієнтний спуск у просторі функцій шляхом апроксимації негативного градієнта функціоналу втрат за кожною ітерацією [18].

На ітерації m розглядаємо поточну модель F_{m-1} та обчислюємо для кожного прикладу псевдо-залишок:

$$r_{im} = - \left. \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right|_{F=F_{m-1}}, \quad i = 1, \dots, n$$

Далі знаходимо базовий учень $h_m(x)$ (наприклад, регресійне дерево), який апроксимує ці псевдо-залишки у вигляді наступної задачі регресії:

$$h_m \approx \arg \min_{h \in H} \sum_{i=1}^n (r_{im} - h(x_i))^2,$$

тобто підбираємо h_m для наближення напрямку найшвидшого спуску у функціональному просторі.

2.2.5 Екстремальний градієнтний бустинг (XGBoost)

Екстремальний градієнтний бустинг (XGBoost — eXtreme Gradient Boosting) є однією з найефективніших реалізацій градієнтного бустингу, яка поєднує алгоритмічні удосконалення та низку системних оптимізацій. На відміну від класичної реалізації бустингу, що була розглянута у попередньому

пункті, XGBoost доповнює базовий підхід механізмами регуляризації, ефективними структурами даних і оптимізованими процедурами побудови дерев, що підвищує його стійкість, швидкість та узагальнюючу здатність [10].

Одним з ключових удосконалень XGBoost є використання додаткового регуляризаційного члена у функціоналі оптимізації. Якщо класичний градієнтний бустинг мінімізує суму втрат, то XGBoost вводить штрафи на складність дерев, що формально можна подати як:

$$L^* = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{k=1}^m \Omega(f_k)$$

де

$$\Omega(f_k) = \gamma T_k + \frac{\lambda}{2} \sum_{j=1}^{T_k} w_j^2$$

T_k — кількість листків у дереві, w_j — ваги листків, а параметри γ та λ контролюють складність моделі. Таким чином, XGBoost безпосередньо запобігає перенавчанню, що є критично важливим при роботі з високимірними або зашумленими даними [10].

Ще однією принциповою відмінністю є використання другої похідної функції втрат (Hessian), що дозволяє переходити від звичайного градієнтного спуску до методу Ньютона в оптимізації. Це забезпечує більш точну апроксимацію оновлень моделі, підвищуючи швидкість збіжності та стабільність навчання.

Також XGBoost реалізує ефективний механізм роботи з пропущеними значеннями, що особливо актуально для фінансових даних. Алгоритм автоматично обчислює оптимальний напрямок розгалуження для записів із пропущеними атрибутами, не вимагаючи додаткової імпутації.

Окремої уваги заслуговує стратегія побудови дерев за листками (leaf-wise growth). На відміну від традиційного підходу, де дерево зростає рівнями (level-wise), XGBoost розширює той листок, який дає найбільше зменшення функції втрат. Це призводить до глибших, але значно якісніших дерев. З метою контролю складності такі параметри, як `max_depth`, `min_child_weight` та `gamma`, дозволяють стримувати надмірне розростання дерева.

З погляду продуктивності XGBoost використовує:

- 1) колонкове (стовпчикове) представлення даних, що підвищує ефективність пошуку оптимальних розбиттів;
- 2) паралельну обробку при побудові дерев;
- 3) кеш-ефективні структури даних та стиснуті блоки;
- 4) алгоритм `weighted quantile sketch`, який забезпечує коректне ранжування навіть для даних зі зваженими прикладами, що є важливим при дисбалансі класів [10].

Крім того, XGBoost підтримує ранню зупинку, крос-валідацію на кожній ітерації, роботу з функціями втрат для класифікації, регресії та ранжування, а також можливість визначення власних `loss-функцій`.

Для задачі прогнозування банкрутства підприємств XGBoost є особливо корисним завдяки:

- 1) здатності обробляти вибірки зі значною кількістю нелінійних залежностей;
- 2) природній роботі з незбалансованими даними (параметр `scale_pos_weight`);
- 3) високій стабільності моделі при шумових даних;
- 4) підвищеній інтерпретованості завдяки можливості будувати глобальні й локальні важливості ознак (зокрема, у поєднанні з SHAP).

Таким чином, XGBoost є однією з найсучасніших та найефективніших реалізацій бустингу, яка поєднує математичну строгість, високу продуктивність та гнучкість використання.

2.3 Інтелектуальна система підтримки прийняття рішень

У сучасній теорії системного аналізу та інформаційних технологій коректна ідентифікація класу розроблюваного програмного забезпечення є фундаментальним етапом, що визначає не лише термінологічний апарат дослідження, але й методологію проектування, критерії оцінювання ефективності та сферу практичного застосування. Розроблена в межах магістерської дисертації система, основною метою якої є аналіз економічних ризиків та прогнозування банкрутства підприємств в умовах невизначеності, потребує чіткого позиціонування в ієрархії інформаційних систем. Базуючись на детальному аналізі функціональних можливостей, архітектури та алгоритмічного наповнення системи, зокрема використання ансамблевих методів машинного навчання (XGBoost, Random Forest) та методів пояснюваного штучного інтелекту (SHAP), можна стверджувати, що дана розробка виходить за межі традиційних інформаційно-довідкових систем (MIS) або систем обробки транзакцій (TPS). Вона демонструє сукупність ознак, характерних для класу Інтелектуальних систем підтримки прийняття рішень (ІСППР).

Системи підтримки прийняття рішень (СППР) історично еволюціонували як інструмент допомоги особам, що приймають рішення (ОПР), у вирішенні складних завдань, де неможливо застосувати повністю автоматизовані алгоритми через невизначеність або брак структурованої інформації. Додавання кваліфікатора «інтелектуальна» передбачає інтеграцію в класичну архітектуру СППР методів штучного інтелекту, які дозволяють системі не лише оперувати даними, але й генерувати нові знання, навчатися на основі досвіду (історичних даних) та пояснювати логіку своїх висновків. У контексті

даного дослідження, інтелектуалізація досягається шляхом імплементації моделей машинного навчання для виявлення латентних нелінійних залежностей у фінансовій звітності та використання технологій інтерпретації для забезпечення прозорості рішень.

2.3.1 Аналіз природи проблемної області

Першочерговим критерієм для класифікації системи як СППР є тип задачі, яку вона покликана вирішувати. Згідно з класифікацією Г. Саймона та подальшими розробками Горрі і Скотта-Мортон, задачі управління поділяються на структуровані, слабоструктуровані та неструктуровані [22, 23]. Аналіз предметної області дисертаційного дослідження — оцінки фінансової стійкості та прогнозування банкрутства — дозволяє віднести її до класу слабоструктурованих задач, що є прямою передумовою для використання СППР.

Специфіка економічного середовища України, яка характеризується екстремальною невизначеністю внаслідок воєнних дій, геополітичної напруженості та структурних зрушень, унеможливорює використання повністю детермінованих алгоритмів. Класичні моделі, такі як Z-рахунок Е. Альтмана або модель Таффлера, які базуються на жорстко фіксованих лінійних коефіцієнтах, втрачають свою прогностичну ефективність в умовах високої волатильності ринку. Вони не здатні адаптуватися до нових патернів кризових явищ без ручного перекалібрування.

Задача діагностики банкрутства характеризується такими ознаками слабоструктурованості, що вирішуються в рамках системи:

1. Багатофакторність та неповнота даних. Оцінка базується на широкому спектрі фінансових показників $X_1 - X_{18}$, що включають оборотні активи,

зобов'язання, операційні витрати тощо. При цьому реальні дані часто містять пропуски, шуми та викиди, що вимагає інтелектуальної попередньої обробки.

2. Нелінійність зв'язків. Вплив окремих факторів на ймовірність дефолту не є адитивним. Наприклад, зростання зобов'язань X17 може бути критичним для компанії з низькою оборотністю активів, але прийнятним для швидкозростаючого бізнесу. Виявлення таких контекстуальних залежностей вимагає застосування нелінійних методів, реалізованих через ансамблеві моделі.

3. Імовірнісний характер результату. Система не може гарантувати 100% якість прогнозу. Вона генерує оцінку ймовірності, яка слугує допоміжним аргументом для ОПР. Остаточне рішення про кредитування або санацію приймає людина, спираючись на рекомендації системи.

Таким чином, оскільки задача не піддається повній алгоритмізації і вимагає синергії обчислювальної потужності машини та евристичних здібностей людини, система обґрунтовано класифікується як система підтримки прийняття рішень.

Для формального доведення належності розробленої системи до класу ІСППР необхідно провести структурний аналіз її архітектури та зіставити його з канонічною моделлю СППР, що складається з трьох базових підсистем (управління даними, управління моделями, інтерфейс користувача) та четвертої — підсистеми знань (або інтелектуального компонента).

2.3.2 Підсистема управління даними

Інтелектуальна система повинна базуватися на якісному інформаційному фундаменті. Цей компонент реалізовано через модулі завантаження, валідації та глибокої попередньої обробки даних (ETL-процеси). Підсистема даних

виконує не лише пасивну функцію зберігання, але й активну інтелектуальну обробку.

Система забезпечує прийом вхідних даних у форматах CSV/Excel, що містять 18 фінансових індикаторів, та їх приведення до уніфікованого вигляду.

Реалізовано алгоритми виявлення аномалій та викидів (Outlier Detection) на основі методу міжквартильного розмаху (IQR). Це критично важливо для фінансових даних, де помилка введення або разова аномальна транзакція можуть викривити результати моделювання.

Унікальною характеристикою підсистеми даних, що наближає її до інтелектуальних систем, є інтеграція алгоритму SMOTE (Synthetic Minority Over-sampling Technique). Оскільки реальні дані про банкрутства є суттєво незбалансованими (клас «failed» є рідкісним порівняно з «alive»), звичайна статистична система ігнорувала б ризики. Розроблена система використовує SMOTE для генерації синтетичних прикладів у просторі ознак, фактично «уявляючи» можливі варіанти кризових станів, яких не було в історичній вибірці, але які є вірогідними. Це свідчить про наявність елементів індуктивного навчання вже на етапі підготовки даних.

2.3.3 Підсистема управління моделями

Підсистема моделей є ядром СППР, що забезпечує аналітичні можливості. Цей компонент трансформувався з банку статичних формул у динамічне середовище машинного навчання, що є головним аргументом на користь статусу ІСППР.

Система не обмежується одним алгоритмом, а надає користувачеві доступ до бібліотеки моделей: від класичних (Altman Z-Score) до передових методів машинного навчання (Decision Tree, Random Forest, SVM, Gradient Boosting,

XGBoost). Це дозволяє реалізувати сценарний аналіз та перехресну перевірку результатів.

Ключовою ознакою інтелектуальності є використання алгоритму XGBoost (Extreme Gradient Boosting). На відміну від статичних моделей, XGBoost є алгоритмом, здатним до ітеративного покращення своєї структури шляхом мінімізації функції втрат. Система здатна «довчатися» на нових даних, адаптуючись до зміни економічних умов, що неможливо для традиційних експертних систем з жорсткими правилами.

Підсистема включає механізми налаштування гіперпараметрів моделей та відбору ознак за допомогою методів RFE (Recursive Feature Elimination) та аналізу важливості ознак. Це забезпечує автоматичну адаптацію моделі до специфіки вхідного набору даних.

2.3.4 Підсистема знань та інтелектуального аналізу

У класичних СППР знання часто задаються експертами у вигляді правил. У розробленій системі знання видобуваються автоматично з даних та формалізуються у вигляді навчених ансамблів дерев рішень. Однак, «інтелектуальність» системи не обмежується лише точністю прогнозу. Критичним елементом є пояснюваність, яка реалізується через інтеграцію бібліотеки SHAP.

Модуль SHAP перетворює абстрактні ваги моделі XGBoost у зрозумілі для людини економічні чинники. Він визначає внесок кожного показника (наприклад, X8 — ринкова вартість або X15 — нерозподілений прибуток) у фінальну оцінку ризику для конкретного підприємства.

Завдяки SHAP система надає не просто число, а аргументоване пояснення («ризик високий через критичне падіння чистого прибутку X6 при зростанні довгострокового боргу X11»). Це дозволяє ОПР верифікувати логіку системи,

спираючись на власні професійні знання, що створює ефект синергії між штучним та природним інтелектом.

2.3.5 Підсистема інтерфейсу користувача

Інтерфейс системи розроблено як веб-орієнтоване середовище (SaaS), що забезпечує діалоговий режим взаємодії.

Інтерфейс трансформує багатовимірні масиви даних у інтуїтивно зрозумілі графічні образи: кореляційні матриці, графіки розподілу класів, діаграми важливості ознак та графіки SHAP.

Користувач має можливість керувати процесом моделювання: обирати набір ознак, змінювати алгоритм прогнозування, завантажувати нові дані для тестування. Це відповідає принципу «що-якщо» (what-if analysis), який є базовим для СППР.

2.3.6 Критеріальний аналіз інтелектуальності системи

Для остаточного підтвердження належності системи до класу ІСППР доцільно проаналізувати її відповідність специфічним критеріям інтелектуальності [24].

У таблиці 2.1 наведено зіставлення функціональних можливостей розробленої системи з вимогами до ІСППР.

Таблиця 2.1. Критерії інтелектуальності розробленої системи

Критерій інтелектуальності	Реалізація в системі	Рівень відповідності
Здатність до навчання	Система використовує алгоритми ML (XGBoost, RF), які навчаються на історичних даних, покращуючи точність прогнозування.	Високий
Робота з нечіткою / неповною інформацією	Алгоритми дерев рішень та бустингу здатні обробляти пропущені значення; використання імовірнісних оцінок замість бінарних.	Високий
Адаптивність	Можливість перенавчання моделей на нових даних; автоматичний відбір значущих ознак під конкретну вибірку.	Середній
Пояснюваність рішень	Інтеграція модуля SHAP для генерації локальних та глобальних пояснень впливу факторів на ризик банкрутства.	Високий
Виведення нових знань	Виявлення прихованих нелінійних патернів банкрутства, які не описуються класичними фінансовими теоріями.	Високий
Підтримка на всіх етапах прийняття рішень	Підтримка фаз Intelligence (моніторинг), Design (моделювання сценаріїв) та Choice (ранжування ризиків) за Г. Саймоном [22].	Високий

Аналіз таблиці підтверджує, що розроблена система задовольняє всі ключові вимоги до інтелектуальних систем. Особливо важливою є здатність системи працювати в умовах незбалансованості даних (завдяки SMOTE), що є типовою проблемою для реальних економічних задач, яку не вирішують стандартні статистичні пакети.

Підсумовуючи викладене, можна зробити висновок, що розроблена система повною мірою відповідає класу Інтелектуальних систем підтримки прийняття рішень (ІСППР). Це твердження базується на наступних доказах:

1. Проблемна орієнтація. Система призначена для вирішення слабоструктурованої задачі прогнозування банкрутства в умовах високої невизначеності, де традиційні алгоритмічні підходи є неефективними.

2. Інтелектуальний інструментарій. Ядром системи є ансамблеві моделі машинного навчання (XGBoost, Random Forest), здатні до виявлення нелінійних закономірностей, у поєднанні з методами інтелектуальної обробки даних (SMOTE) та відбору ознак (RFE).

3. Архітектурна відповідність. Реалізована структура (дані-моделі-інтерфейс-знання) є ізоморфною до канонічної архітектури ІСППР.

4. Когнітивна підтримка. Інтеграція технології Explainable AI (SHAP) забезпечує інтерпретованість результатів, трансформуючи «чорну скриньку» ML-моделі у прозорий інструмент підтримки рішень, що є критичним фактором для довіри з боку користувачів-експертів.

Розроблена ІСППР дозволяє перейти від ретроспективного аналізу фінансового стану до проактивного управління ризиками, забезпечуючи ОПР науково обґрунтованими, точними та інтерпретованими рекомендаціями.

2.4 Метрики оцінювання адекватності побудованих моделей

Оцінювання ефективності моделей прогнозування банкрутства вимагає ретельного підбору метрик, оскільки стандартна «Влучність» (Accuracy) в умовах дисбалансу класів може вводити в оману. У дослідженні використано комплексний підхід, що включає аналіз матриці помилок, F1-міри, ROC-аналізу та Precision-Recall кривих.

2.4.1 Матриця помилок та похідні метрики

Базовим інструментом оцінки є матриця помилок (Confusion Matrix), яка для бінарної класифікації містить чотири компоненти [19]:

1. True Positive (TP): Правильно класифіковані банкрути.
2. True Negative (TN): Правильно класифіковані діючі компанії.
3. False Positive (FP): Помилка I роду (хибна тривога) — діюча компанія визнана банкрутом.
4. False Negative (FN): Помилка II роду (пропуск цілі) — банкрут визнаний діючою компанією.

На основі цих компонентів розраховуються:

1. Precision (Якість): $\frac{TP}{TP+FP}$. Показує, наскільки можна довіряти прогнозу «банкрут». Низька якість означає багато хибних тривог, що збільшує витрати на перевірку клієнтів.

2. Recall (Повнота, Sensitivity): $\frac{TP}{TP+FN}$. Показує, яку частку реальних банкрутів виявила система. У ризик-менеджменті ця метрика є критичною, оскільки пропуск банкрутства (FN) несе значні фінансові збитки.

2.4.2 Інтегральні метрики: F_1 та AUC-ROC

Для балансування між якістю та повнотою використовується F_1 -міра, яка є їхнім середнім гармонічним [19]:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Найбільш робастною метрикою для порівняння класифікаторів є площа під ROC-кривою — AUC-ROC (Area Under Receiver Operating Characteristic Curve). ROC-крива відображає залежність між TPR (Recall) та FPR ($\frac{FP}{FP+TN}$) при зміні порогу класифікації. AUC оцінює ймовірність того, що модель ранжує випадковий позитивний приклад вище за випадковий негативний.

Також аналізувалися Precision-Recall (PR) криві, які є більш інформативними при сильному дисбалансі, оскільки не залежать від кількості True Negatives [19,20].

2.5 Інтерпретація результатів моделей (SHAP)

Важливою вимогою до сучасних ІСППР є не лише якість прогнозу, але й його пояснюваність. «Чорні скриньки», якими є складні ансамблі, часто викликають недовіру у користувачів-аналітиків.

Для вирішення цієї проблеми у роботі запропоновано використання методу SHAP (SHapley Additive exPlanations). Цей метод базується на теорії кооперативних ігор і розраховує внесок кожної ознаки (значення Шеплі) у відхилення прогнозу для конкретного об'єкта від середнього прогнозу по вибірці [11].

SHAP дозволяє отримати як глобальну інтерпретацію (які фактори є найважливішими в цілому), так і локальну (чому саме ця компанія отримала високий ризик банкрутства). Впровадження SHAP у розроблювану веб-орієнтовану ІСППР забезпечує прозорість прийняття рішень, дозволяючи користувачу валідувати прогнози моделі на основі експертних знань, що становить наукову новизну та практичну цінність роботи.

2.6 Висновки до розділу 2

У другому розділі було здійснено комплексний огляд сучасних методів підготовки даних, моделей машинного навчання та метрик оцінювання, що становлять фундамент побудови ефективної системи прогнозування кредитних ризиків. Показано, що якість результатів моделювання значною мірою визначається саме етапами попередньої обробки: очищенням вибірки, роботою з викидами, масштабуванням характеристик та кодуванням категоріальних ознак. Застосування методів балансування класів, зокрема SMOTE, дозволило усунути проблему дисбалансу, характерну для фінансових даних, що суттєво підвищує стабільність та об'єктивність побудованих моделей.

У межах розділу було проаналізовано низку моделей машинного навчання, що репрезентують різні підходи до побудови ансамблів та класифікаційних алгоритмів. Беггінг продемонстрував здатність зменшувати дисперсію базових моделей, забезпечуючи їхню стійкість до варіацій у навчальній вибірці. Градієнтний бустинг, як послідовний ансамблевий підхід, дав змогу гнучко адаптувати процес навчання до структури похибок моделей, а вдосконалена його модифікація XGBoost завдяки регуляризації, ефективній реалізації побудови дерев та підтримці роботи з пропущеними значеннями показала особливу придатність до задачі прогнозування фінансових ризиків.

Обґрунтовано належність розроблюваного програмного комплексу до класу інтелектуальних систем підтримки прийняття рішень (ІСППР), що зумовлено слабоструктурованим характером задачі прогнозування банкрутства та необхідністю обробки нечіткої інформації в умовах невизначеності. Спроектовано архітектуру системи, яка базується на канонічній моделі (підсистеми даних, моделей, знань та інтерфейсу), але якісно відрізняється інтеграцією механізмів машинного навчання та модуля пояснюваного штучного інтелекту (Explainable AI). Доведено, що така структура забезпечує не лише високу точність прогнозування завдяки алгоритмам ансамблевого типу, а й інтерпретованість результатів, створюючи синергію між автоматизованим аналізом та експертним судженням особи, що приймає рішення.

Окрему увагу приділено метрикам оцінювання моделей. Показано, що традиційна влучність є недостатньою при дисбалансі класів, тоді як F1-міра, ROC- та Precision–Recall-криві надають глибше та інформативніше уявлення про ефективність моделей у задачах виявлення високоризикових позичальників. Аналіз кривих навчання дозволив оцінити поведінку моделей у контексті проблем недонавчання та перенавчання, що є важливим для побудови стійких і узагальнюючих рішень.

Таким чином, у межах розділу сформовано цілісну методологічну основу побудови моделей прогнозування кредитних ризиків, що охоплює як передобробку та відбір характеристик, так і вибір оптимальних алгоритмів машинного навчання та способів їх оцінювання. Отримані результати є необхідним підґрунтям для подальшої практичної реалізації, проведення обчислювальних експериментів та розробки системи підтримки прийняття рішень у наступному розділі.

РОЗДІЛ 3 ПОБУДОВА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ТА РОЗРОБКА ІСППР

3.1 Аналіз та підготовка вхідних даних

Експериментальна частина дослідження базується на використанні історичних фінансових даних компаній, що функціонують на ринку США [21]. У вибірці представлені як фінансово стійкі підприємства, так і компанії, що збанкрутували, що дає змогу оцінювати ризик банкрутства на основі ретроспективної інформації (рис. 3.1). Згідно з описом даних, у наборі містяться 18 числових фінансових показників (X1 – X18), серед яких, зокрема, поточні активи, чистий прибуток, ринкова вартість компанії та загальні активи, а також цільова змінна `status_label`, що набуває значень «alive» або «failed».

	company_name	status_label	year	X1	X2	X3	X4	X5	X6	X7	...	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18
0	C_1	alive	1999	511.267	833.107	18.373	89.031	336.018	35.163	128.348	...	1024.333	740.998	180.447	70.658	191.226	163.816	201.026	1024.333	401.483	935.302
1	C_1	alive	2000	485.856	713.811	18.577	64.367	320.590	18.531	115.187	...	874.255	701.854	179.987	45.790	160.444	125.392	204.065	874.255	361.642	809.888
2	C_1	alive	2001	436.656	526.477	22.496	27.207	286.588	-58.939	77.528	...	638.721	710.199	217.699	4.711	112.244	150.464	139.603	638.721	399.964	611.514
3	C_1	alive	2002	396.412	496.747	27.172	30.745	259.954	-12.410	66.322	...	606.337	686.621	164.658	3.573	109.590	203.575	124.106	606.337	391.633	575.592
4	C_1	alive	2003	432.204	523.302	26.680	47.491	247.245	3.504	104.661	...	651.958	709.292	248.666	20.811	128.656	131.261	131.884	651.958	407.608	604.467

5 rows × 21 columns

Рисунок 3.1 – Зразок вхідних даних

Вибірка налічує 78 682 входжень, з наступними показниками:

- 1) X1: оборотні активи;
- 2) X2: собівартість реалізованої продукції;
- 3) X3: амортизація та знос;
- 4) X4: EBITDA – прибуток до відсотків, податків, амортизації та зносу;
- 5) X5: запаси;
- 6) X6: чистий прибуток;
- 7) X7: дебіторська заборгованість;
- 8) X8: ринкова вартість;

- 9) X9: чисті продажі;
- 10) X10: загальні активи;
- 11) X11: довгостроковий борг;
- 12) X12: ЕВІТ – Прибуток до відсотків і податків.
- 13) X13: валовий прибуток;
- 14) X14: поточні зобов'язання;
- 15) X15: нерозподілений прибуток;
- 16) X16: загальний дохід;
- 17) X17: загальні зобов'язання;
- 18) X18: операційні витрати.

На рисунку 3.2 подано графічне відображення співвідношення підприємств, що продовжують діяльність «alive», та компаній, які припинили її унаслідок банкрутства «failed». Як видно з рисунка, у вибірці спостерігається виражений дисбаланс класів: кількість об'єктів класу «alive» суттєво перевищує обсяг представників класу «failed».

Такий розподіл відповідає реальній економічній ситуації, у якій банкрутства є рідкісною, але критично важливою для прогнозування подією. Водночас наявність значного дисбалансу ускладнює побудову ефективних моделей класифікації, оскільки без додаткових заходів алгоритми машинного навчання схильні орієнтуватися на домінуючий клас, знижуючи чутливість щодо виявлення підприємств із підвищеним ризиком дефолту.

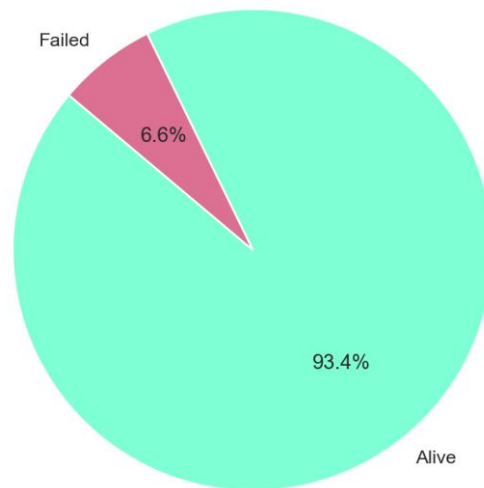


Рисунок 3.2 – Співвідношення банкрутів у даних

Також для аналізу датасету було побудовано кореляційну матрицю (рис. 3.3).

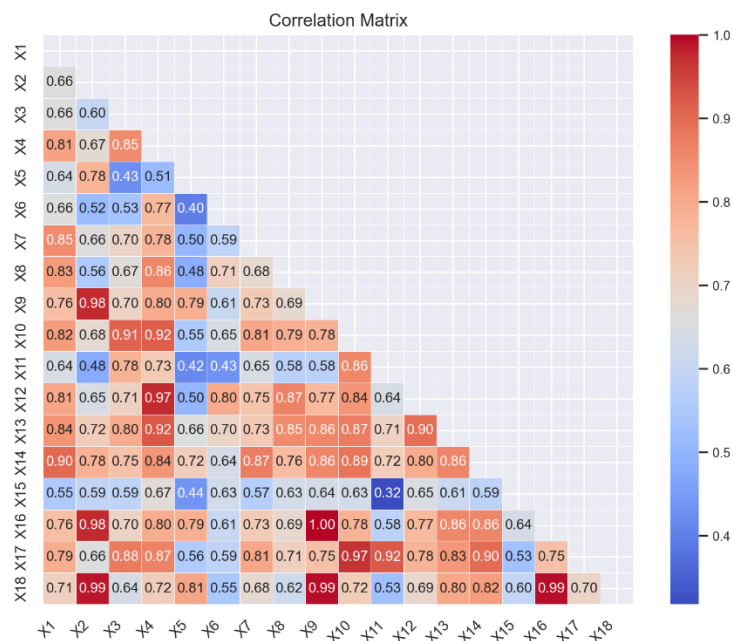


Рисунок 3.3 – Кореляційна матриця

Аналіз матриці виявив наступні залежності:

1. **Сильні позитивні кореляції.** Було виявлено зв'язки, що вказують на те, що змінні мають тенденцію зростати разом. Найсильніші з них спостерігаються між X3 та X16 (0.83), а також між X15 та X18 (0.78).

2. **Помірні кореляції.** Спостерігається значуща, але не абсолютна залежність, наприклад, між X_7 та X_{14} (0.75).

3. **Слабкі кореляції.** Деякі пари, як-от X_{13} та X_{14} (0.01), демонструють практично повну відсутність лінійного зв'язку.

4. **Негативні кореляції.** Матриця також показала зворотні зв'язки, де збільшення однієї змінної відповідає зменшенню іншої, наприклад, між X_1 та X_{15} (-0.67).

Ці результати вказують на потенційні проблеми мультиколінеарності, що може негативно вплинути на побудову прогностичних моделей. Для зменшення надмірності даних може знадобитися відбір ознак або зменшення розмірності.

Для ідентифікації викидів у наборі фінансових показників $X_1 - X_{18}$ застосовано метод міжквартильного розмаху (IQR) (рис. 3.4).

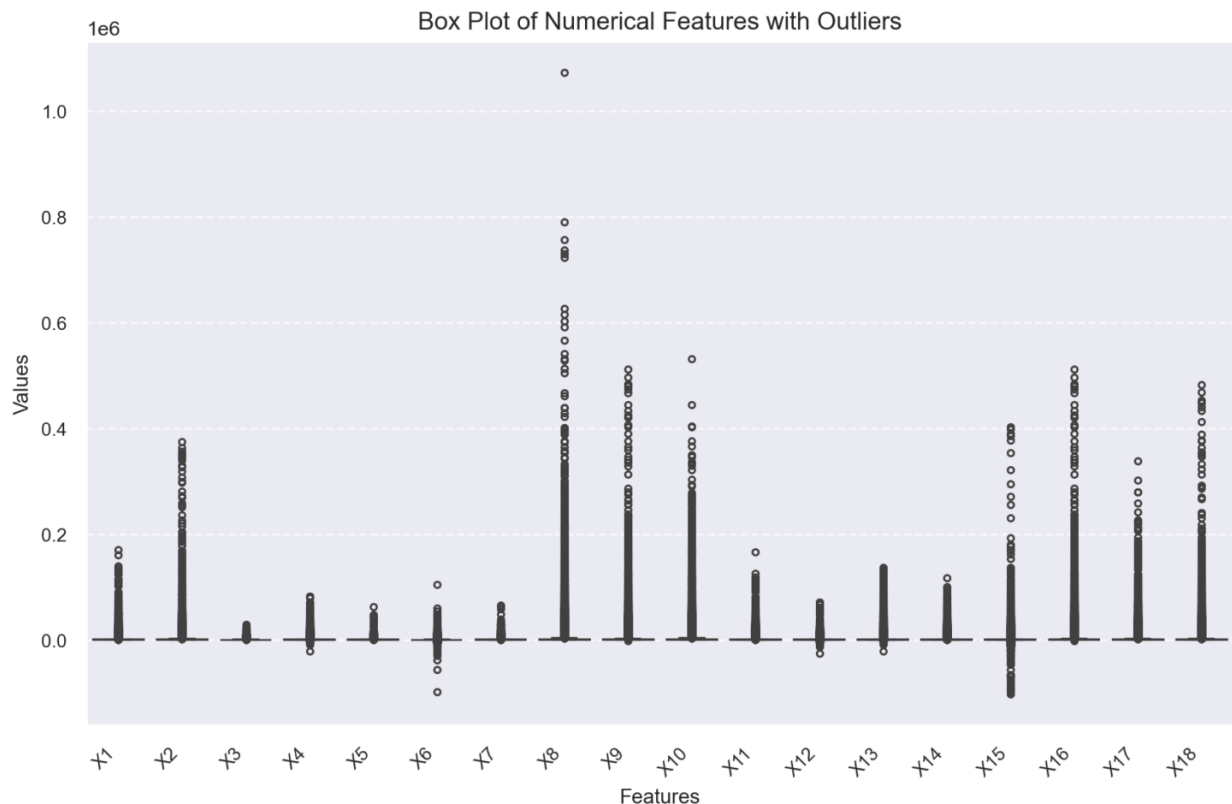


Рисунок 3.4 – Графік викидів

Процедура передбачає такі етапи:

1. Обчислення першого та третього кuartилів для кожної ознаки:

$$Q1 = x_{25} \quad Q3 = x_{75}$$

2. Визначення міжквартильного розмаху:

$$IQR = Q3 - Q1$$

3. Формування меж допустимого діапазону:

$$Lower = Q1 - 1.5 * IQR \quad Upper = Q3 + 1.5 * IQR$$

4. Класифікація значень як викидів у разі порушення умов:

$$X_{ij} < Lower \text{ або } X_{ij} > Upper$$

Побудований графік (рис. 3.4) дозволяє оцінити розподіл числових змінних та ідентифікувати наявні викиди. Для низки показників, зокрема X8, X9, X10, X15 та X16 спостерігається значна кількість значень, що виходять за межі міжквартильного розмаху, що підтверджує наявність екстремальних фінансових станів або значних відхилень у масштабах діяльності окремих підприємств. Вертикальна вісь графіка демонструє суттєву різницю у порядках величин між ознаками, що додатково підкреслює необхідність застосування процедур масштабування для забезпечення коректної роботи моделей машинного навчання.

Окрему увагу слід звернути на характер розподілу ознак: для більшості з них встановлено правобічну асиметрію, коли медіанні значення розміщуються у нижній частині діапазону, а основний обсяг варіації зосереджений у верхньому хвості розподілу. Така форма кривої є типовою для фінансових даних і свідчить про наявність значної кількості компаній із помірними показниками та порівняно невеликого числа підприємств із екстремально високими значеннями.

У роботі також застосовано метод SMOTE з метою балансування цільової змінної щодо належності підприємства до класу «alive» або «failed». Хоча використання цього методу сприяє підвищенню якості класифікації у контексті дисбалансу класів, він не усуває проблеми викидів, які зберігають вплив на структуру ознак. Це обумовлює необхідність подальшого опрацювання екстремальних значень або застосування додаткових

перетворень, таких як вінзоризація чи логарифмування. Розуміння природи цих викидів є важливим для подальшого моделювання, оскільки вони можуть містити релевантну інформацію щодо фінансової нестабільності, а ефективність різних підходів до їх обробки може істотно відрізнятись залежно від типу моделі.

Отже, первинний аналіз датасету показав наявність низки проблем, характерних для фінансових даних:

1. Незбалансованість класів. Кількість компаній, що залишилися платоспроможними («alive»), суттєво перевищує кількість збанкрутілих («failed») (рис. 3.2). Це призводить до того, що наївні класифікатори, орієнтовані на максимізацію загальної якості прогнозу, схильні «ігнорувати» рідкісний клас банкрутів, що є неприйнятним для задачі ризик-менеджменту.

2. Різномірність масштабів ознак. Фінансові показники суттєво відрізняються за порядком величин: від коефіцієнтів рентабельності до абсолютних значень активів. Така різниця може призводити до домінування окремих змінних у метриках відстані та процесі оптимізації параметрів моделей.

3. Мультиколінеарність. Попередній розвідувальний аналіз показав наявність значних кореляцій між частиною ознак, що відображають близькі за економічним змістом характеристики (наприклад, різні варіанти оцінки активів чи ринкової вартості) (рис. 3.3). Надлишкова корельована інформація ускладнює інтерпретацію та може сприяти перенавчанню моделей.

3.2 Відбір значущих характеристик

Вихідний набір фінансових показників $X_1 - X_{18}$ характеризується суттєвою мультиколінеарністю, різномірністю масштабів та наявністю викидів, що ускладнює побудову стійких прогностичних моделей. За таких

умов відбір інформативних характеристик є обов'язковим етапом побудови ІСППР, оскільки дозволяє зменшити розмірність простору ознак, усунути надлишкову інформацію, знизити ризик перенавчання та підвищити інтерпретованість результатів. Теоретичні засади методів відбору ознак, зокрема рекурсивного виключення ознак (RFE) та підходів на основі важливості ознак у деревоподібних моделях.

У рамках даного дослідження відбір характеристик здійснювався модельно-орієнтованим чином, тобто окремо для кожного з використаних алгоритмів машинного навчання. Такий підхід враховує специфіку того, як саме різні моделі «працюють» з ознаками: дерева рішень та ансамблі дерев природно формують міру важливості на основі зменшення домішок, градієнтний бустинг і LightGBM оперують власними вагами розбиттів, метод опорних векторів потребує попереднього RFE, а XGBoost у поєднанні з SHAP забезпечує інтерпретовану оцінку внеску кожної змінної в прогноз. Додатково для порівняння було використано класичну скорингову модель Altman Z-Score, для якої відбір фактично реалізовано в самій конструкції фінансових коефіцієнтів.

Для кожної моделі, навченій на попередньо підготовлених даних (масштабування ознак, балансування класів за допомогою SMOTE), обчислювалися показники важливості ознак або застосовувався спеціалізований алгоритм відбору (RFE, SHAP-аналіз). Далі для забезпечення зіставності результатів для кожного класифікатора формувалась підмножина з десяти найбільш інформативних характеристик, яка використовувалася у подальшому моделюванні.

Для моделей Random Forest та Decision Tree було застосовано стандартний підхід на основі важливості ознак, що обчислюється за критерієм зменшення індексу Джині при розбиттях у вузлах дерева. Найвищі значення важливості отримали змінні X1, X3, X6, X7, X8, X10, X13, X14, X15 та X17 (рис. 3.5). Такий набір відображає, з одного боку, значущість показників ринкової вартості, нерозподіленого прибутку, амортизаційних відрахувань і

обігових активів, а з іншого – роль зобов’язань та дебіторської заборгованості в оцінці довгострокової стійкості підприємства.

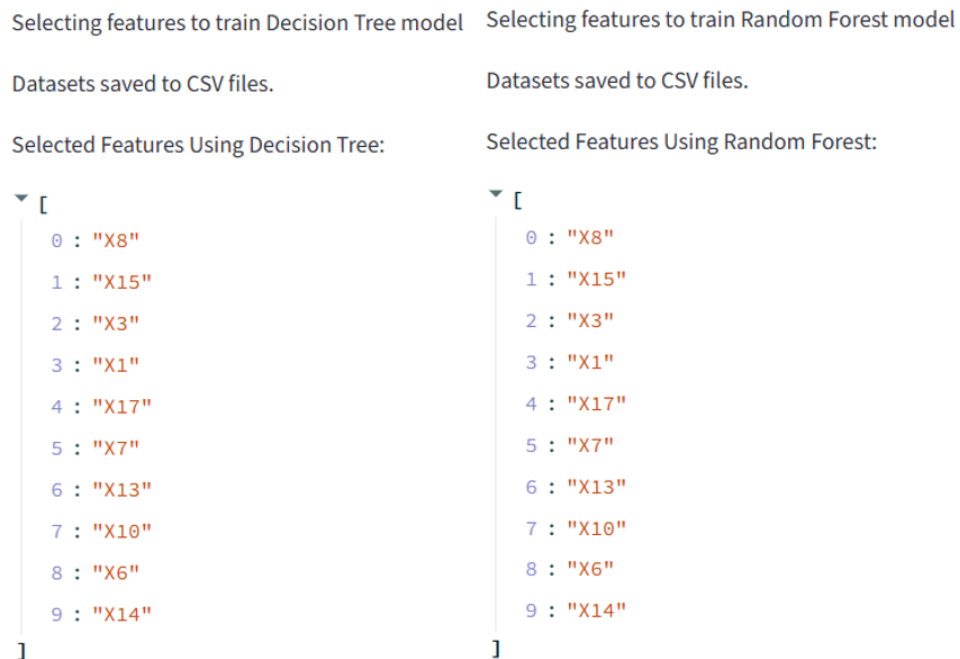


Рисунок 3.5 – Обрані ознаки Random Forest та Decision Tree

У випадку градієнтного бустингу на базі LightGBM (LGBMClassifier) використано вбудовані оцінки важливості ознак, що базуються на частоті та якості використання тієї чи іншої характеристики у розбиттях дерев ансамблю. За результатами аналізу до десятки найкращих потрапили X3, X5, X8, X15, змінна часу year, а також X1, X2, X7, X11 та X13 (рис. 3.6). Включення змінної year свідчить про те, що часовий контекст (період спостереження) також має помітний вплив на ймовірність банкрутства, відображаючи макроекономічні цикли або структурні зміни в економіці.

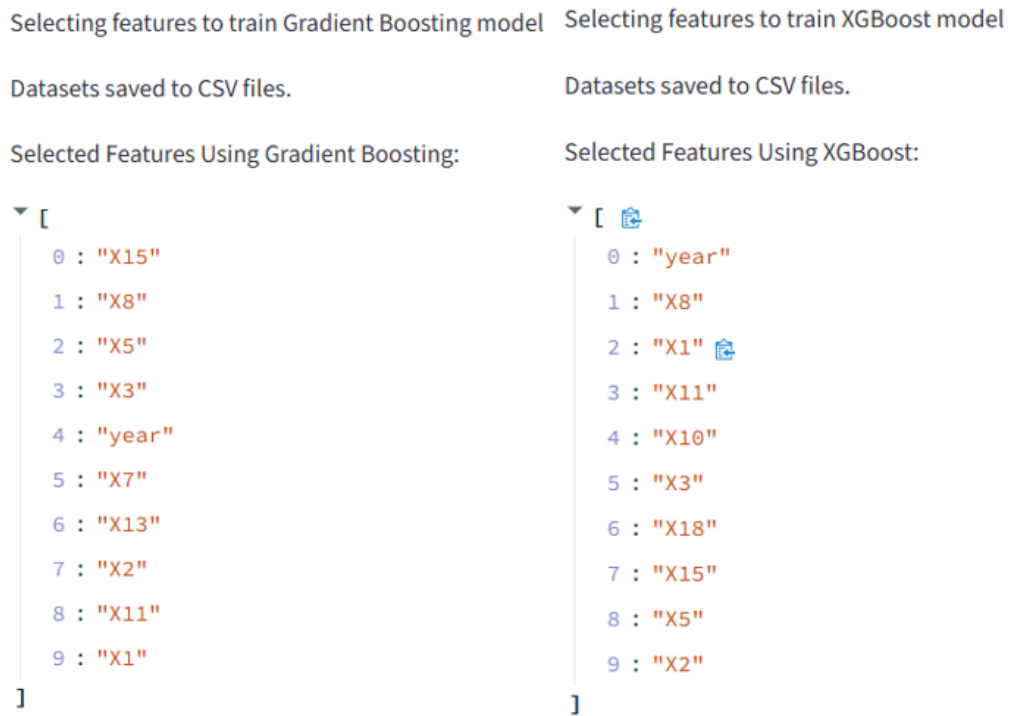


Рисунок 3.6 – Обрані ознаки Градієнтного бустингу та XGBoost

Для XGBoost відбір ознак здійснювався із використанням SHAP-аналізу, що дозволив ранжувати характеристики за їх середнім абсолютним внеском у прогноз. До фінального набору увійшли year, X1, X2, X3, X5, X8, X10, X11, X15 та X18 (рис. 3.6). Порівняння з результатами LightGBM показує, що XGBoost надає особливої ваги поєднанню ринкової вартості та обігових активів (X1, X8) з борговими зобов'язаннями (X11) та операційними витратами (X18), що добре узгоджується з економічною інтуїцією щодо ключових драйверів фінансової нестабільності.

У випадку методу опорних векторів (SVM), для якого прямі міри важливості ознак не є інтерпретованими, було застосовано рекурсивне виключення ознак (RFE). У результаті ітеративного вилучення найменш інформативних змінних сформовано підмножину з десяти ознак: year, X1, X4, X5, X6, X11, X12, X13, X15 та X16 (рис. 3.7). Це свідчить про чутливість SVM до узагальненої структури результатів діяльності (прибуток до відсотків і податків, чистий прибуток, загальний дохід) у поєднанні з часовим фактором та базовими показниками активів.

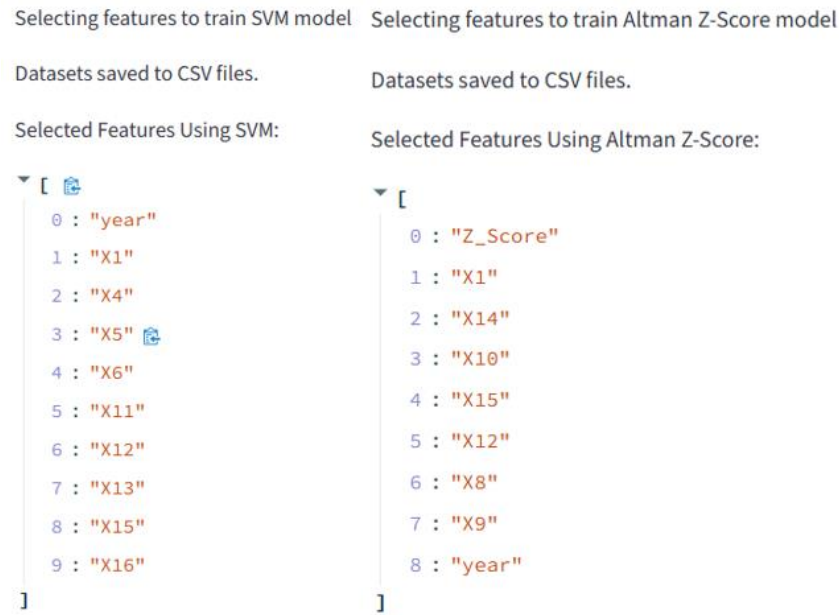


Рисунок 3.7 – Обрані ознаки SVM та Altman Z-Score

Для моделі Altman Z-Score набір характеристик задається конструкцією відповідних фінансових коефіцієнтів, однак у ході аналізу додатково враховано змінні X1, X8, X9, X10. X12, X14, X15 та year, що використовувалися як вихідні компоненти для розрахунку Z-показника й пов’язаних із ним метрик (рис. 3.7). Це дозволяє зіставити «класичний» простір ознак фінансової діагностики з простором, який формують сучасні алгоритми машинного навчання.

Узагальнені результати відбору наведено в таблиці 3.1.

Таблиця 3.1 – Відібрані ознаки

Модель	Відібрані ознаки
Random Forest, Decision Tree	X1, X3, X6, X7, X8, X10. X13, X14, X15, X17
Gradient Boosting (LGBM)	year, X1, X2, X3, X5, X7, X8, X11, X13, X15
XGBoost (SHAP-аналіз)	year, X1, X2, X3, X5, X8, X10. X11, X15, X18
SVM (RFE)	year, X1, X4, X5, X6, X11, X12, X13, X15, X16
Altman Z-Score	Z_Score, year, X1, X8, X9, X10. X12, X14, X15

Аналіз таблиці 3.1 дає змогу виокремити кілька важливих закономірностей. По-перше, ознаки X1, X3, X8 та X15 присутні у більшості моделей, що вказує на їхню стрижневу роль у поясненні ризику банкрутства. З економічної точки зору це означає, що ринкова вартість компанії, нерозподілений прибуток, амортизаційні відрахування та обігові активи є ключовими індикаторами як поточної фінансової стійкості, так і потенційних кризових станів.

По-друге, змінна *year* систематично з'являється в наборах ознак для моделей градієнтного бустингу, XGBoost, SVM та Altman Z-Score, що підкреслює значення часової компоненти. Це свідчить про те, що ймовірність банкрутства суттєво залежить від макроекономічного циклу та змін зовнішнього середовища, а не лише від внутрішніх характеристик підприємства.

По-третє, деякі моделі мають власні «сфери фокусу»: XGBoost виразно акцентує на поєднанні ринкових та боргових показників (X1, X8, X10, X11, X18), тоді як SVM, завдяки RFE, надає більшої ваги показникам результатів діяльності (X4, X5, X6, X12, X16). Така різниця в структурі ознак є додатковою перевагою ансамблевого підходу, оскільки дає змогу враховувати різні аспекти фінансового стану підприємства.

Загалом проведений відбір значущих характеристик дозволив суттєво зменшити розмірність простору ознак та зосередити увагу моделей на ключових фінансових індикаторах, що описують ліквідність, рентабельність, структуру капіталу, масштаби діяльності та ринкові оцінки. Це, у свою чергу, створює передумови для підвищення якості прогнозування, зменшення ризику перенавчання та покращення інтерпретованості результатів, що є критично важливим для інтеграції побудованих моделей у ІСППР.

3.3 Побудова моделей та порівняння результатів

На цьому етапі, спираючись на підготовлений набір даних, масштабовані ознаки та збалансований за допомогою SMOTE цільовий клас, було виконано навчання та оцінювання низки моделей прогнозування банкрутства підприємств: традиційної фінансової моделі Altman Z-Score, дерева рішень (DecisionTree), випадкового лісу (RandomForest), градієнтного бустингу (LGBMClassifier), екстремального градієнтного бустингу (XGBoost) та методу опорних векторів (SVM).

У межах програмної реалізації для кожної моделі формувалися типові візуалізації: матриця помилок, ROC-крива та крива Precision–Recall, а також зведена діаграма метрик на тестовій вибірці. Сукупність цих графіків дозволяє оцінити не лише інтегральні показники якості, а й структуру помилок (співвідношення FP/FN) та поведінку моделей при зміні порога прийняття рішення.

Для традиційної моделі Altman Z-Score матриця помилок демонструє задовільну здатність відокремлювати фінансово стійкі компанії, однак порівняно вищу частку пропусків реальних банкрутів (елементи FN), що є типово для лінійних скорингових підходів, оптимізованих на загальну влучність (рис. 3.8 – 3.9). На ROC-кривій модель формує плавну, але відносно пологоу траєкторію з площею під кривою близько 0.83, що вказує на прийнятну, але не найвищу розрізнявальну здатність. Precision–Recall-крива для Z-Score розташована нижче кривих сучасних моделей машинного навчання, особливо в області високих значень recall, що свідчить про обмежену здатність класичної моделі підтримувати задовільну якість прогнозу при спробі «впіймати» більшу кількість потенційних банкрутів.

Altman/Logistic Regression Accuracy: 87.62%

Confusion Matrix:

0	1
515	553
1395	13274

Classification Report:

	0	1	accuracy	macro avg	weighted avg
precision	0.2696	0.96	0.8762	0.6148	0.9132
recall	0.4822	0.9049	0.8762	0.6936	0.8762
f1-score	0.3459	0.9316	0.8762	0.6388	0.8919
support	1068	14669	0.8762	15737	15737

Рисунок 3.8 – Результати Altman Z-Score

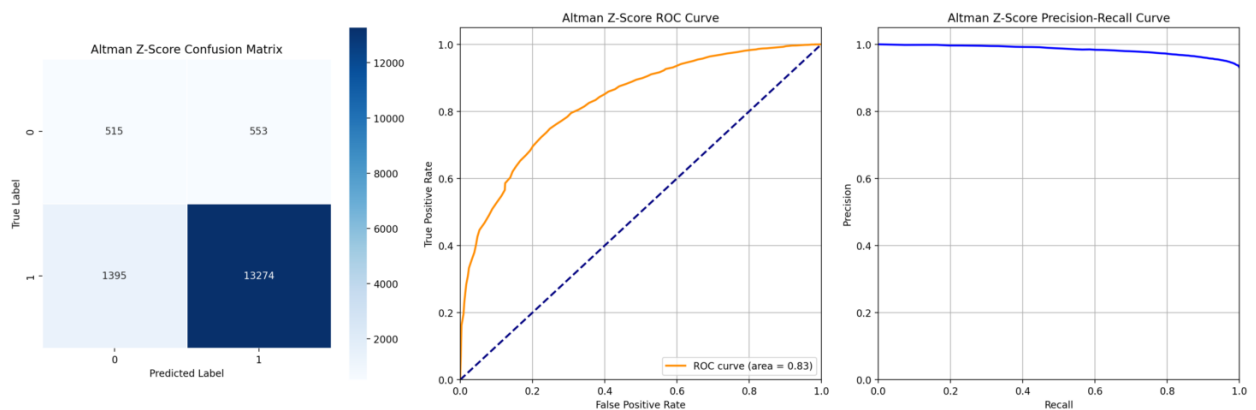


Рисунок 3.9 – Візуалізація результатів Altman Z-Score

Для дерева рішень (DecisionTree) матриця помилок виявляє характерну для поодиноких дерев схильність до перенавчання: модель добре відтворює структуру навчальних даних, але на тестовій вибірці демонструє суттєві перекося – частка як хибнопозитивних (FP), так і хибнонегативних (FN) класифікацій є вищою, ніж у більшості інших алгоритмів (рис. 3.10 – 3.11). На ROC-кривій DecisionTree формує траєкторію, що лише незначно відхиляється від діагоналі випадкового вгадування (AUC близько 0.67), а PR-крива

проходить у нижній частині площини, особливо в зоні високих значень recall. Це означає, що при спробі підвищити виявлення банкрутів модель різко втрачає якість, що знижує її придатність як базового інструменту ризик-менеджменту.

Decision Tree Accuracy: 89.15%

Confusion Matrix:

0	1
242	826
882	13787

Classification Report:

	0	1	accuracy	macro avg	weighted avg
precision	0.2153	0.9435	0.8915	0.5794	0.8941
recall	0.2266	0.9399	0.8915	0.5832	0.8915
f1-score	0.2208	0.9417	0.8915	0.5812	0.8927
support	1068	14669	0.8915	15737	15737

Best Decision Tree Accuracy: 93.26%

Confusion Matrix:

0	1
8	1060
1	14668

Classification Report:

	0	1	accuracy	macro avg	weighted avg
precision	0.8889	0.9326	0.9326	0.9107	0.9296
recall	0.0075	0.9999	0.9326	0.5037	0.9326
f1-score	0.0149	0.9651	0.9326	0.49	0.9006
support	1068	14669	0.9326	15737	15737

Рисунок 3.10 – Результати DecisionTree

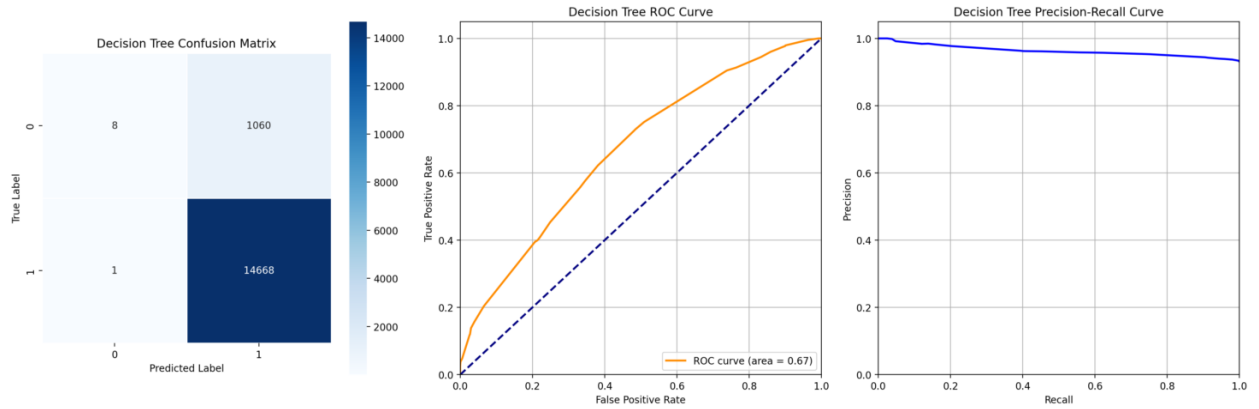


Рисунок 3.11 – Візуалізація результатів DecisionTree

Для випадкового лісу (RandomForest) картина суттєво покращується. Матриця помилок свідчить про значне збільшення кількості правильно виявлених банкрутів (TP) порівняно з Altman Z-Score та DecisionTree, при збереженні прийнятного рівня FP (рис. 3.12 – 3.13). Це є прямим наслідком використання SMOTE на етапі навчання, завдяки чому модель отримує достатню кількість прикладів міноритарного класу й не «ігнорує» його. На ROC-кривій RandomForest демонструє досить круту траєкторію з AUC близько 0.82, що підтверджує високу розрізнявальну здатність. Precision–Recall-крива розташована істотно вище за класичну модель та дерево рішень, зберігаючи високі значення precision навіть при зростанні recall, що є важливим для задачі виявлення компаній з підвищеним ризиком банкрутства.

Random Forest Accuracy: 93.51%

Confusion Matrix:

0	1
54	1014
7	14662

Classification Report:



	0	1	accuracy	macro avg	weighted avg
precision	0.8852	0.9353	0.9351	0.9103	0.9319
recall	0.0506	0.9995	0.9351	0.525	0.9351
f1-score	0.0957	0.9664	0.9351	0.531	0.9073
support	1068	14669	0.9351	15737	15737

Cross-Validation Scores:

Fold 1: 93.29%

Fold 2: 93.21%

Fold 3: 93.17%

Fold 4: 93.15%

Fold 5: 93.39%

Average Cross-Validation Score: 93.24%

Рисунок 3.12 – Результаты RandomForest

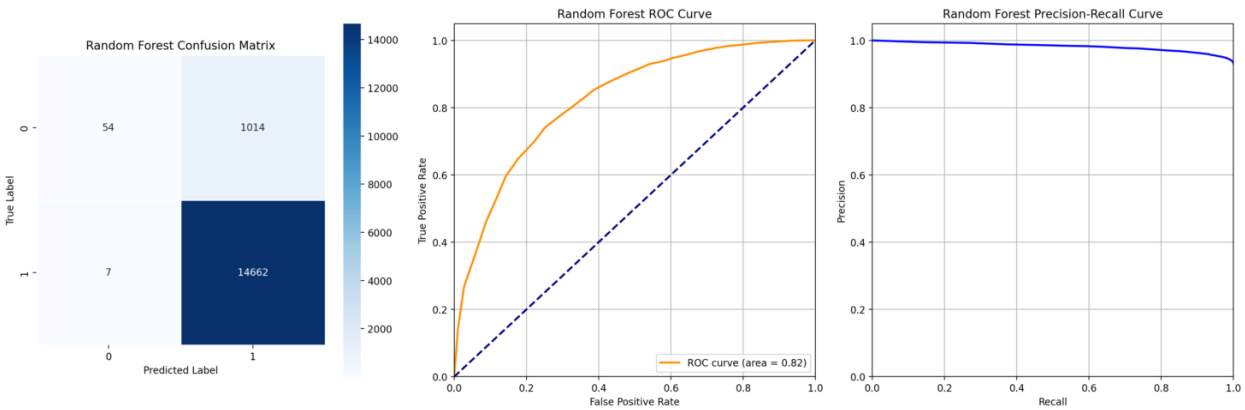


Рисунок 3.13 – Візуалізація результатів RandomForest

У випадку градієнтного бустингу (LGBMClassifier) матриця помилок показує подібну до RandomForest структуру помилок, із дещо більшою чутливістю до класу «failed» за рахунок невеликого збільшення FP (рис. 3.14 – 3.15). На зведеній діаграмі метрик градієнтний бустинг демонструє якість на рівні ~0.93–0.94 та F1-міру близько 0.95, тобто не поступається найкращим моделям. ROC-крива для GradientBoosting проходить вище за криву DecisionTree і SVM, але дещо нижче за XGBoost, а PR-крива підтверджує здатність моделі підтримувати прийнятний компроміс між precision та recall у широкому діапазоні порогів. Сукупність графіків вказує на те, що градієнтний бустинг є стабільним та продуктивним інструментом, хоча й не абсолютним лідером за жодною з ключових метрик.

Gradient Boosting Accuracy: 93.45%

Confusion Matrix:

0	1
23	1021
10	14683

Classification Report:

	0	1	accuracy	macro avg	weighted avg
precision	0.697	0.935	0.9345	0.816	0.9192
recall	0.022	0.9993	0.9345	0.5107	0.9345
f1-score	0.0427	0.9661	0.9345	0.5044	0.9048
support	1044	14693	0.9345	15737	15737

Рисунок 3.14 – Результати Градієнтного бустингу

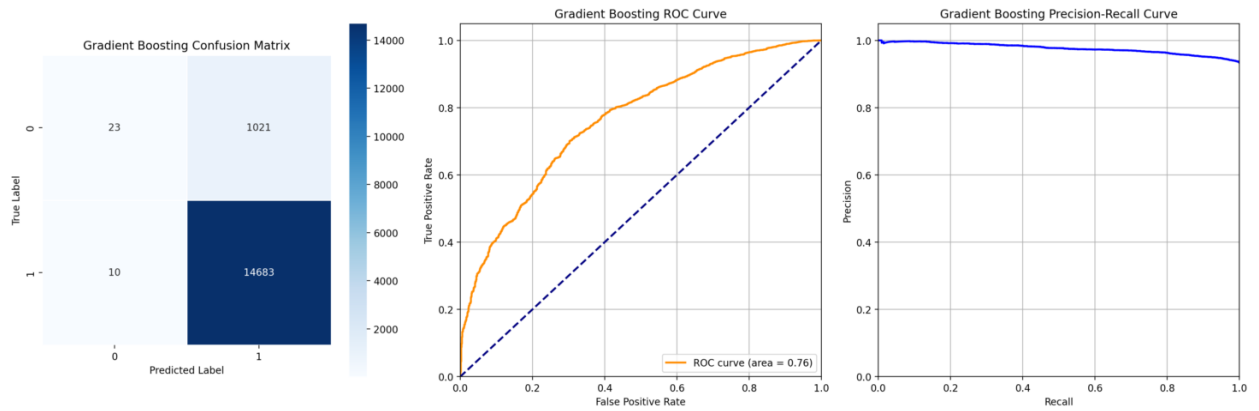


Рисунок 3.15 – Візуалізація результатів Градієнтного бустингу

Для XGBoost результати є найбільш переконливими. Матриця помилок демонструє одну з найнижчих часток пропущених банкрутів (FN) при помірній кількості FP, що особливо важливо з позицій мінімізації кредитних збитків (рис. 3.16 – 3.17). На зведеній діаграмі метрик XGBoost має accuracy на рівні $\sim 0.93\text{--}0.94$ і F1-міру близько 0.95, що відповідає групі найкращих моделей. Водночас ROC-крива XGBoost розташована вище за криві всіх інших алгоритмів, забезпечуючи площу під кривою близько 0.85, тобто найвищу розрізнявальну здатність серед досліджуваних моделей. Precision–Recall-крива XGBoost також є однією з найкращих: при зростанні recall до високих значень модель тривалий час зберігає високий рівень precision, що свідчить про здатність виявляти значну частку проблемних компаній без різкого зростання кількості хибних тривог.

XGBoost Accuracy: 93.63%

Confusion Matrix:

0	1
116	952
51	14618

Classification Report:

	0	1	accuracy	macro avg	weighted avg
precision	0.6946	0.9389	0.9363	0.8167	0.9223
recall	0.1086	0.9965	0.9363	0.5526	0.9363
f1-score	0.1879	0.9668	0.9363	0.5773	0.914
support	1068	14669	0.9363	15737	15737

Cross-Validation Scores:

Fold 1: 93.26%

Fold 2: 93.00%

Fold 3: 92.67%

Fold 4: 92.60%

Fold 5: 93.30%

Average Cross-Validation Score: 92.97%

Рисунок 3.16 – Результати XGBoost

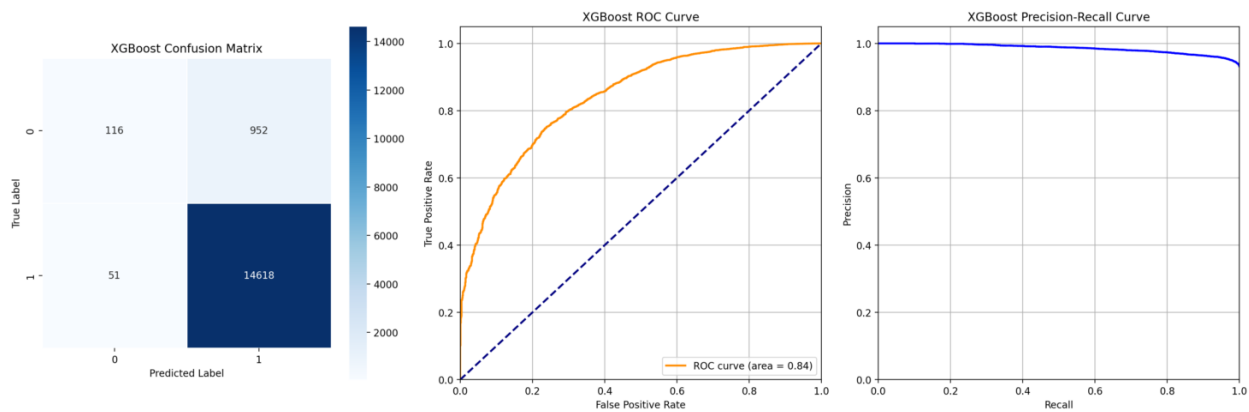


Рисунок 3.17 – Візуалізація результатів XGBoost

Для методу опорних векторів (SVM) матриця помилок показує дуже високу загальну точність класифікації та добре збалансований розподіл TP та TN, однак більш чутливу реакцію на зміну порога: при налаштуванні на максимізацію F1-міри модель досягає одних із найвищих значень цієї метрики (~ 0.96), але за рахунок дещо меншої стійкості при варіюванні порога (рис. 3.18 – 3.19). Це добре відображено на ROC-кривій, яка має площу близько 0.69 і пролягає нижче кривих XGBoost, RandomForest та навіть Altman Z-Score, що вказує на обмежену здатність SVM до стабільного ранжування ризикових та надійних компаній у широкому діапазоні порогів. Натомість Precision–Recall-крива для SVM у точці оптимального порогу демонструє дуже високий баланс precision та recall, що і приводить до максимального значення F1-міри серед усіх моделей.

SVM Accuracy: 94.50%

Confusion Matrix:

0	1
1	11
0	188

Classification Report:

	0	1	accuracy	macro avg	weighted avg
precision	1	0.9447	0.945	0.9724	0.948
recall	0.0833	1	0.945	0.5417	0.945
f1-score	0.1538	0.9716	0.945	0.5627	0.9225
support	12	188	0.945	200	200

Рисунок 3.18 – Результати SVM

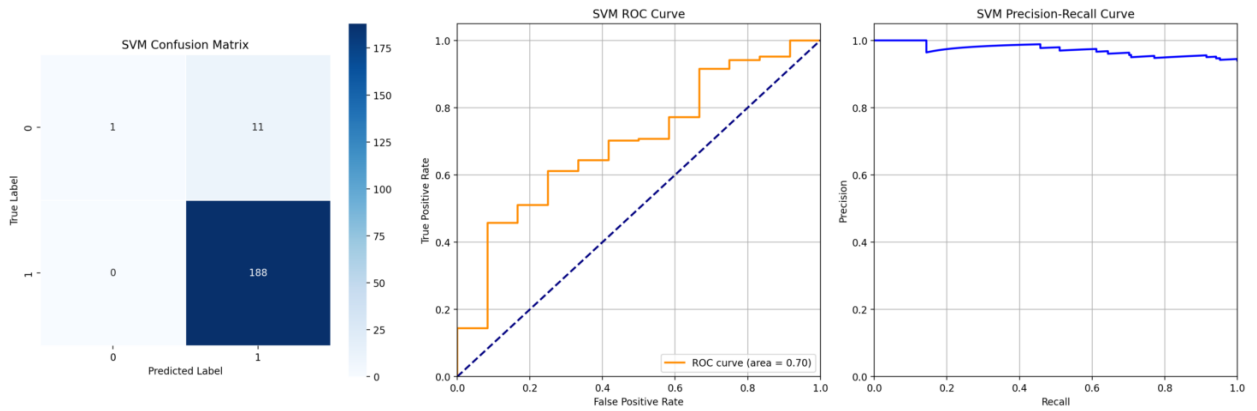


Рисунок 3.19 – Візуалізація результатів SVM

Зведена діаграма метрик (рис. 3.20) узагальнює наведені спостереження. За показником якості усі моделі машинного навчання (RandomForest, SVM, DecisionTree, XGBoost, GradientBoosting) демонструють дуже близькі значення (~ 0.93 – 0.94), що суттєво перевищує базовий рівень Altman Z-Score (~ 0.88). Це підтверджує, що використання сучасних алгоритмів машинного навчання дозволяє істотно підвищити загальну якість прогнозування банкрутства порівняно з класичною фінансовою моделлю.

При аналізі F1-міри (рис. 3.20), яка є більш показовою для задач із дисбалансом класів, SVM виступає явним лідером ($F1 \approx 0.96$), тоді як RandomForest, XGBoost та GradientBoosting мають дуже близькі, дещо нижчі значення (~ 0.95). Це свідчить про те, що SVM найкраще налаштований на баланс між точністю та повнотою виявлення компаній-банкрутів за обраного порога, тоді як ансамблеві моделі забезпечують трохи нижчий, але все ще дуже високий рівень F1 за рахунок більшої робастності до зміни порогів.

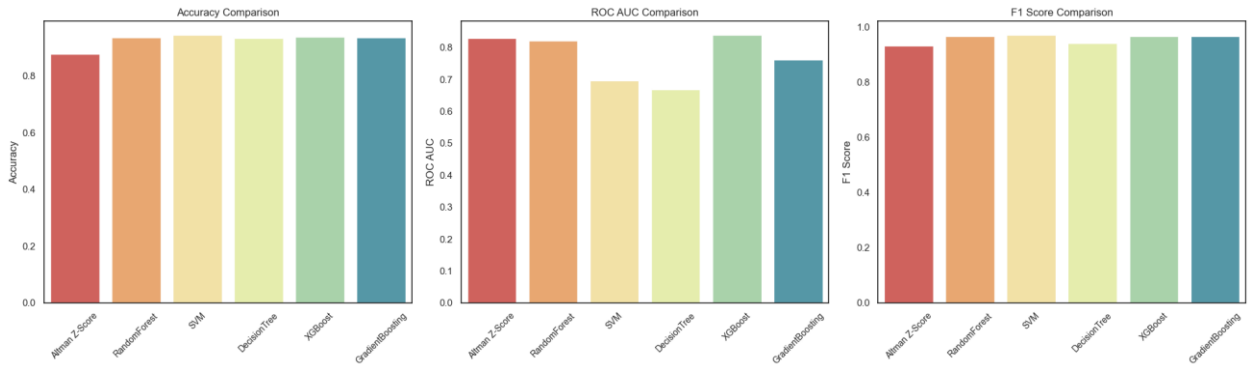


Рисунок 3.20 – Метрики адекватності моделей

Аналіз ROC AUC (рис. 3.20 – 3.21) виявляє іншу ієрархію: беззаперечним лідером є XGBoost (AUC ~ 0.85), за ним ідуть Altman Z-Score (~ 0.83) та RandomForest (~ 0.82), тоді як SVM (~ 0.69) та DecisionTree (~ 0.67) демонструють суттєво гірші результати, лише трохи вищі за випадкове вгадування. Така картина узгоджується з графіками Precision–Recall: саме XGBoost і RandomForest зберігають високі значення precision у широкому діапазоні значень recall, що є критичним для сценаріїв, коли ризик-менеджер може змінювати поріг залежно від рівня допустимого кредитного ризику.

Важливу роль у покращенні результатів відіграло попереднє балансування даних за допомогою SMOTE. Порівняння матриць помилок до та після застосування SMOTE показало суттєве збільшення кількості правильно виявлених банкрутів (TP) та зменшення пропусків (FN) для класу «failed» майже у всіх моделей, при помірному зростанні FP. Це означає, що моделі стали чутливішими до рідкісного, але критично важливого класу, що цілком відповідає цілям системи підтримки прийняття рішень у сфері аналізу економічних ризиків.

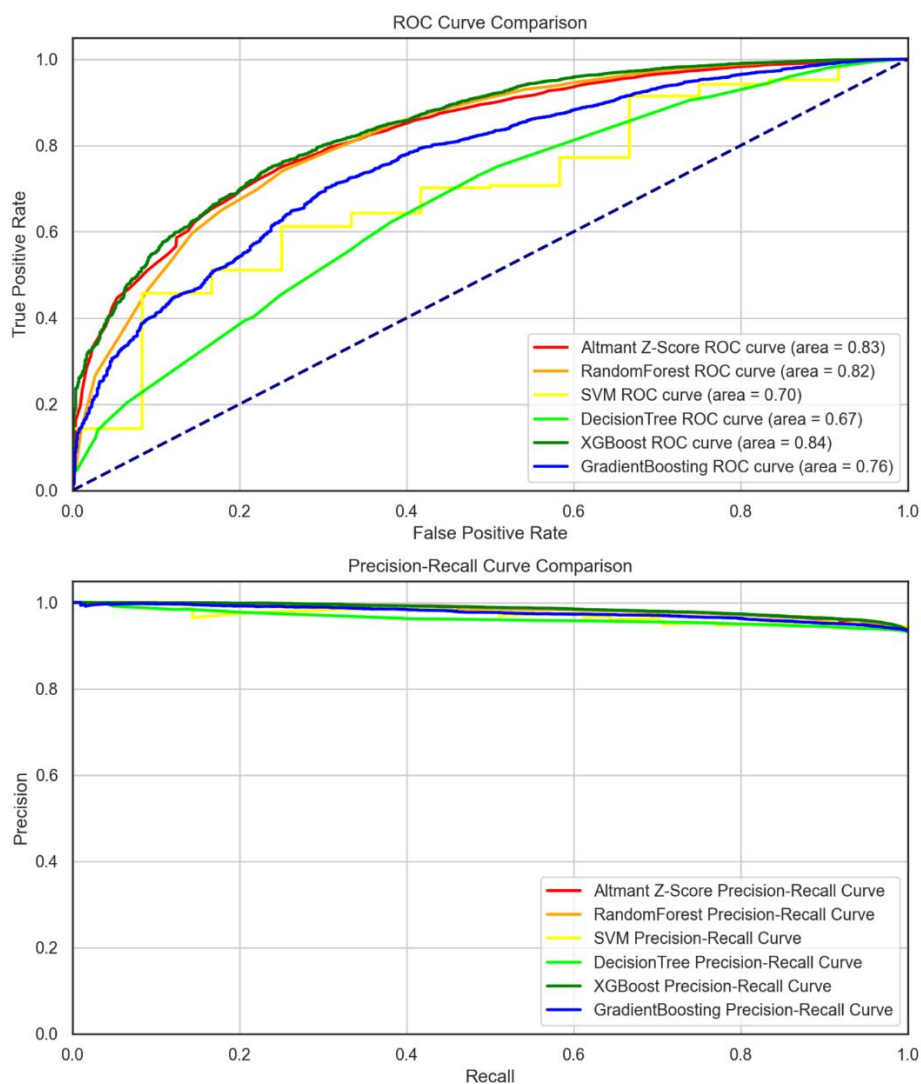


Рисунок 3.21 – ROC та Precision-Recall криві побудованих моделей

3.4 Розробка ІСППР

Розробка інтелектуальної системи підтримки прийняття рішень для аналізу економічних ризиків передбачає перехід від окремих моделей машинного навчання до цілісного програмного комплексу, що забезпечує повний цикл роботи з даними: від завантаження та попередньої обробки до побудови прогнозу та його інтерпретації. Необхідність саме інтегрованого рішення зумовлена як проблемами якості та незбалансованості фінансових

даних, так і вимогами до прозорості та пояснюваності рішень, що ухвалюються на основі складних ансамблевих моделей.

Функціональна схема ІСППР побудована у вигляді веб-орієнтованого застосунку з модульною структурою, де кожен модуль відповідає окремому етапу аналітичного процесу (рис. 3.22). На цій схемі виділено модулі завантаження та візуалізації даних, підготовки вибірок і відбору ознак, навчання та порівняння моделей, а також модуль прогнозування результатів. Такий поділ відображає послідовність кроків, які фактично виконувалися у процесі побудови моделей: від первинного аналізу вибірки до інтерпретації прогнозу й формування рекомендацій для користувача.

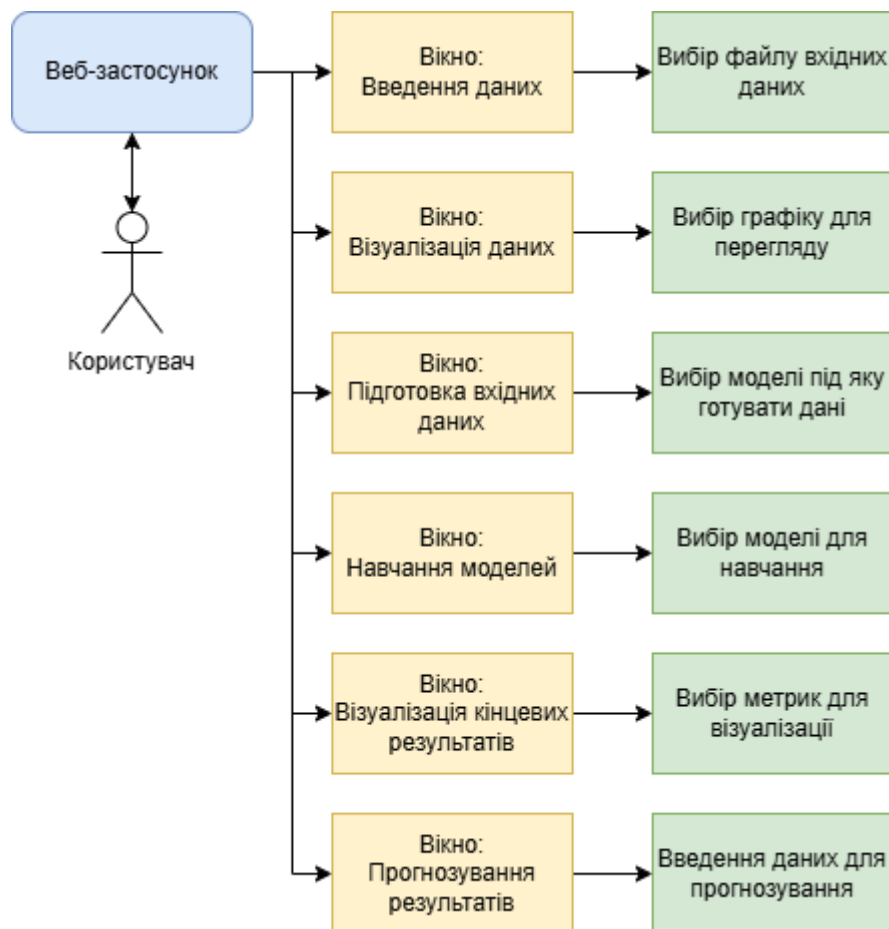


Рисунок 3.22 – Функціональна схема ІСППР

На верхньому рівні система реалізована як веб-застосунок, що слугує загальною оболонкою для всіх функціональних модулів. Через нього

організовано взаємодію користувача із системою: навігацію між вікнами, ініціацію обчислювальних процедур, відображення графіків та табличних результатів (рис. 3.23). Веб-інтерфейс забезпечує уніфікований доступ до всіх етапів аналізу, зменшуючи поріг входу для користувачів, які не є фахівцями в галузі науки про дані, але мають експертні знання у сфері фінансів та ризик-менеджменту.

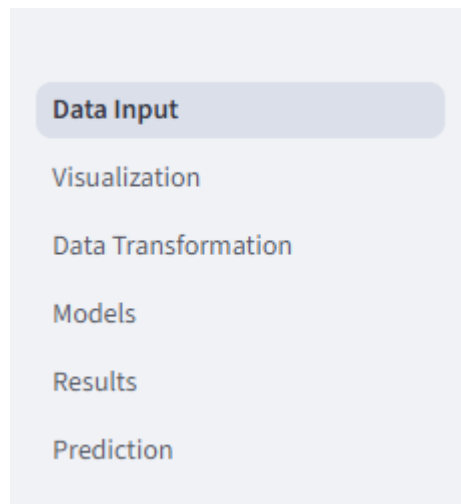


Рисунок 3.23 – Меню веб-інтерфейсу

Модуль завантаження та валідації даних реалізовано у вигляді вікна введення даних, що надає користувачеві можливість обрати файл із вхідним датасетом у стандартизованому форматі (наприклад, CSV чи Excel) (рис. 3.24 – 3.25). Після вибору файлу система виконує базову структурну перевірку: відповідність кількості та назв полів очікуваному переліку фінансових показників X1–X18, наявність цільової змінної статусу компанії, коректність типів даних. На цьому етапі фіксуються типові помилки, пов’язані з пропущеними значеннями, некоректними числовими форматами або дублюванням записів.

↓ 🔍 ↺

	company_name	status_label	year	X1	X2	X3	X4	X5	X6	X7
0	C_1	alive	1999	511.267	833.107	18.373	89.031	336.018	35.163	12
1	C_1	alive	2000	485.856	713.811	18.577	64.367	320.59	18.531	11
2	C_1	alive	2001	436.656	526.477	22.496	27.207	286.588	-58.939	7
3	C_1	alive	2002	396.412	496.747	27.172	30.745	259.954	-12.41	6
4	C_1	alive	2003	432.204	523.302	26.68	47.491	247.245	3.504	10
5	C_1	alive	2004	474.542	598.172	27.95	61.774	255.477	15.453	12
6	C_1	alive	2005	624.454	704.081	29.222	91.877	323.592	35.163	13
7	C_1	alive	2006	645.721	837.171	32.199	118.907	342.593	58.66	18
8	C_1	alive	2007	783.431	1080.895	39.952	168.522	435.608	75.144	20
9	C_1	alive	2008	851.312	1110.677	40.551	166.08	477.424	78.651	

Рисунок 3.24 – Відображення вхідних даних

Data shape: (78682, 21)

	Data Type	Minimum	Maximum	First value	Second value	Third value	Number of Null	Nu
company_na	object	None	None	C_1	C_1	C_1	0	
status_label	object	None	None	alive	alive	alive	0	
year	int64	1999	2018	1999	2000	2001	0	
X1	float64	-7.76	169662	511.267	485.856	436.656	0	
X2	float64	-366.645	374623	833.107	713.811	526.477	0	
X3	float64	0	28430	18.373	18.577	22.496	0	
X4	float64	-21913	81730	89.031	64.367	27.207	0	
X5	float64	0	62567	336.018	320.59	286.588	0	
X6	float64	-98696	104821	35.163	18.531	-58.939	0	
X7	float64	-0.006	65812	128.348	115.187	77.528	0	

Рисунок 3.25 – Аналіз вхідних даних

Наступним елементом функціональної схеми є модуль розвідувального аналізу та візуалізації даних, що реалізований як вікно візуалізації. У ньому користувач може обрати для перегляду кілька типів графіків: співвідношення збанкрутілих компаній до «живих» (рис. 3.26), кореляційну матрицю (рис. 3.27) та розподіл викидів (рис. 3.28). Графік співвідношення класів дозволяє оперативно оцінити ступінь дисбалансу між компаніями зі статусом «failed» і «alive», що є критично важливим для подальшого вибору методів балансування вибірки. Кореляційна матриця відображає структуру взаємозв'язків між фінансовими показниками X_1 – X_{18} , даючи змогу виявляти групи сильно корельованих змінних та потенційну мультиколінеарність. Діаграми з викидами (boxplot) виявляють аномальні значення, які можуть істотно впливати на параметри моделей або бути наслідком помилок обліку.

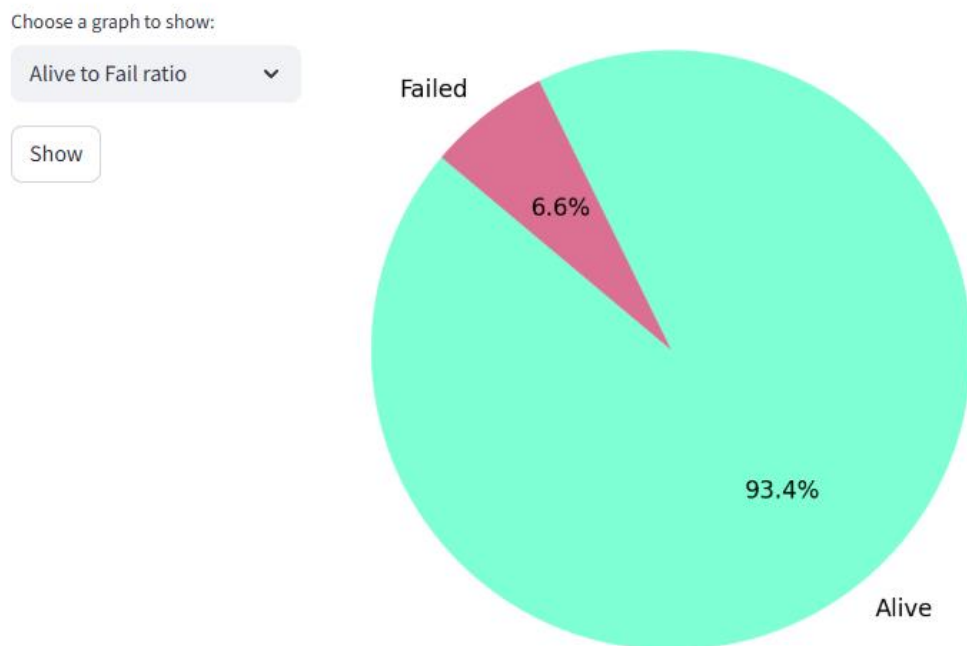


Рисунок 3.26 – Візуалізація відношення збанкрутілих компаній

Choose a graph to show:

Correlation Matrix

Show

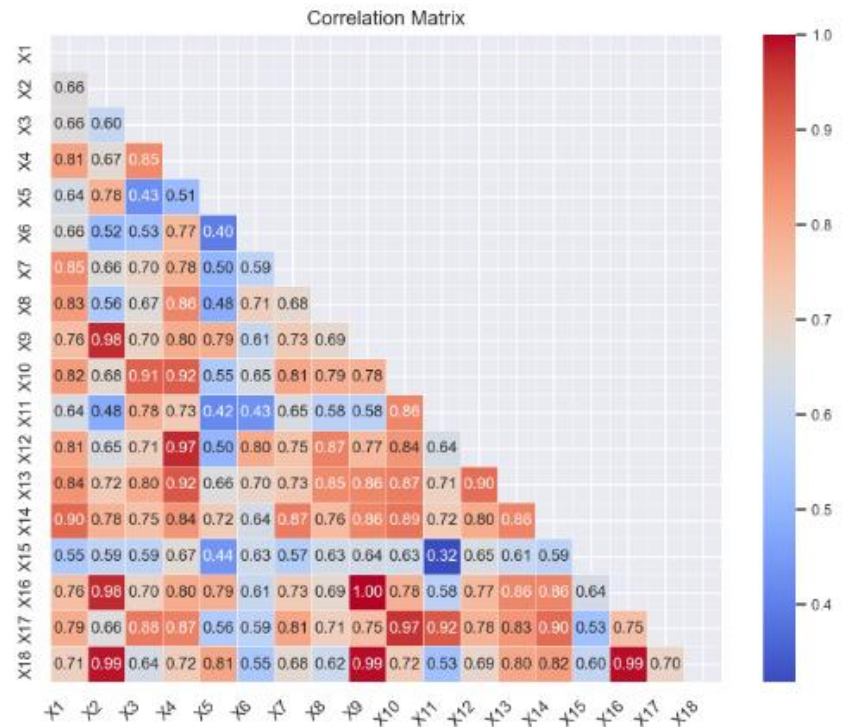


Рисунок 3.27 – Візуалізація кореляційної матриці

Choose a graph to show:

Outliers

Show

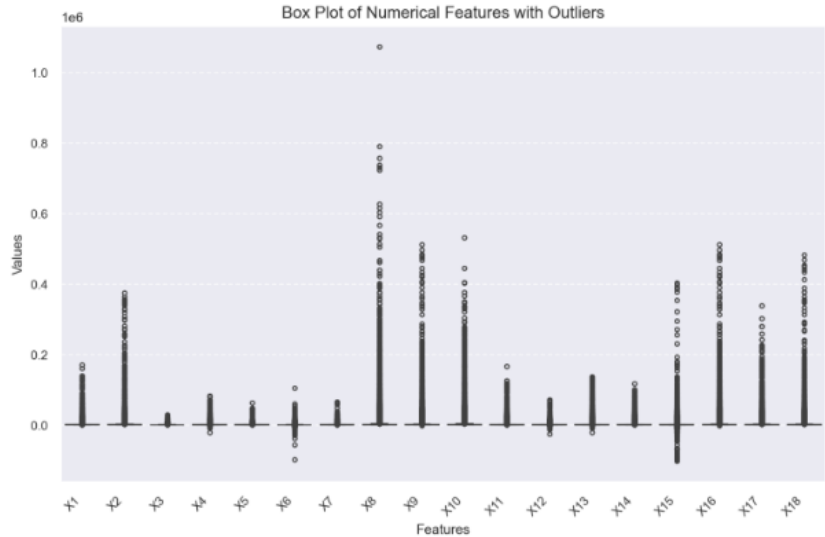


Рисунок 3.28 – Візуалізація викидів

Модуль підготовки вхідних даних відповідає за формування навчальних вибірок з урахуванням особливостей обраних моделей та результатів відбору ознак. У відповідному вікні користувач обирає модель, для якої готуються дані, визначає підмножину найбільш інформативних характеристик

(спираючись на результати аналізу важливості ознак та RFE) та ініціює застосування процедур масштабування, балансування класів за допомогою SMOTE і розбиття вибірки на навчальну та тестову частини (рис. 3.29). На виході модуль формує стандартизований набір даних, приведений до єдиного формату ознак та збережений в окремому файлі. Це забезпечує відтворюваність експериментів і можливість порівняння різних моделей на однаковій інформаційній основі.

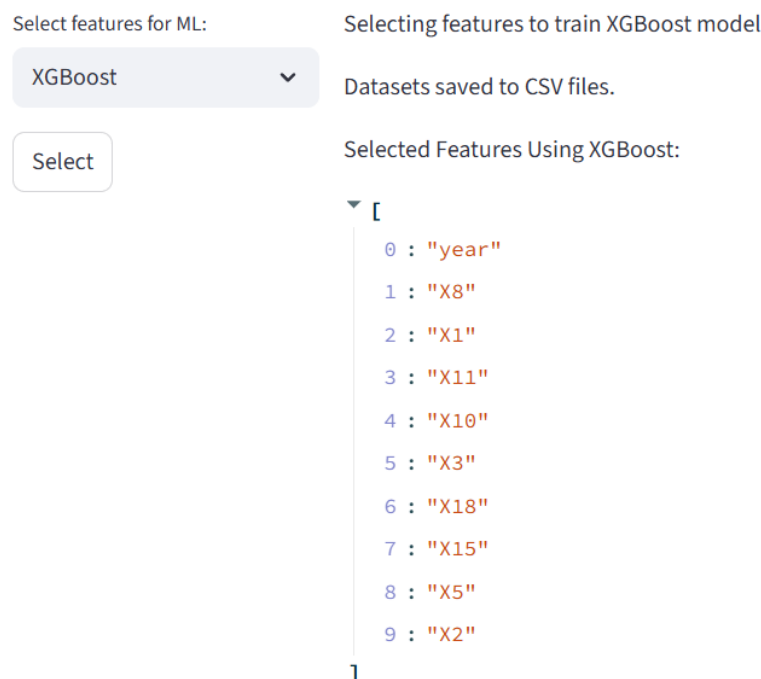


Рисунок 3.29 – Підготовка вхідних даних на прикладі XGBoost

Модуль навчання моделей реалізований у вигляді основного вікна системи, яке активується після завершення підготовки даних. Користувач обирає один з алгоритмів (DecisionTree, RandomForest, SVM, GradientBoosting, XGBoost, а також базову фінансову модель Altman Z-Score) та запускає процедуру навчання (рис. 3.30). Для кожної моделі система послідовно виконує навчання на збалансованій вибірці, розрахунок ключових метрик якості (accuracy, F1-міра, ROC AUC, показники матриці помилок) та побудову візуалізацій — матриць помилок, ROC-кривих та Precision–Recall-кривих. Отримані результати автоматично зберігаються у внутрішньому сховищі

разом із параметрами навчання та версіями сформованих моделей, що створює основу для подальшого відтворення й аудиту рішень.

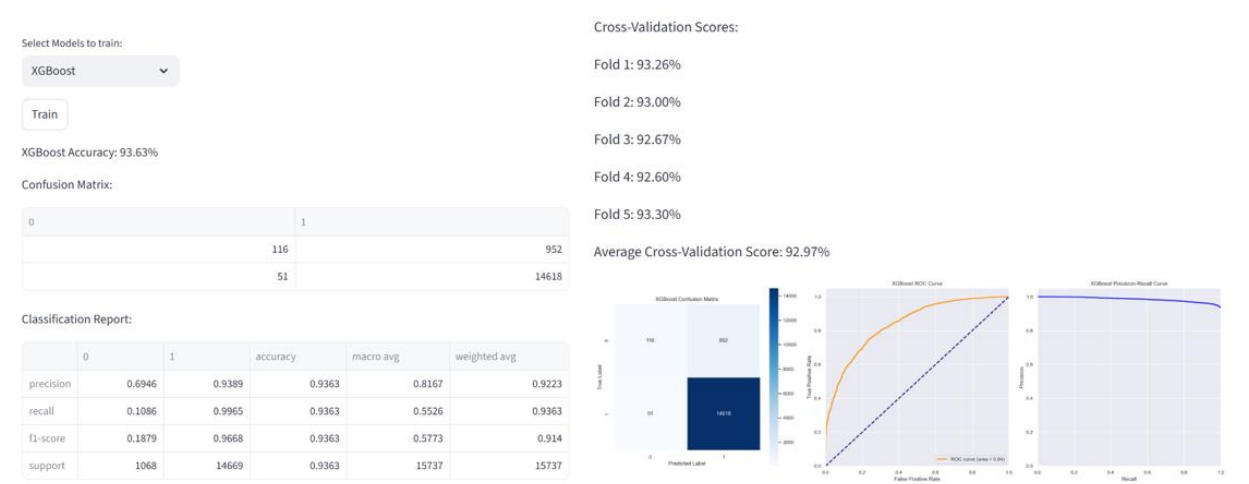


Рисунок 3.30 – Навчання моделей на прикладі XGBoost

Окремий модуль присвячено порівнянню моделей та вибору базового алгоритму для виробничого контуру ІСППР. Відповідне вікно візуалізації кінцевих результатів агрегує метрики всіх навчених моделей і відображає їх у вигляді таблиці (рис. 3.31) або порівняльних графіків: стовпчикових діаграм F1-міри та ROC AUC (рис. 3.32), а також накладених ROC і Precision–Recall-кривих (рис. 3.33). На основі цих візуалізацій користувач може оцінити компроміс між чутливістю до класу банкрутів, кількістю хибних тривог і загальною розрізнявальною здатністю.

Result

Table

	Model	Accuracy	ROC AUC	F1 Score	Cross-Validation Score
0	Altman Z-Score	0.8762	0.8283	0.9316	0.9324
1	RandomForest	0.9351	0.822	0.9664	0.9324
2	SVM	0.945	0.6964	0.9716	0.9388
3	DecisionTree	0.9326	0.6683	0.9417	0.9338
4	XGBoost	0.9363	0.8402	0.9668	0.9297
5	GradientBoosting	0.9345	0.761	0.9661	0.9334

Рисунок 3.31 – Таблица порівняння моделей

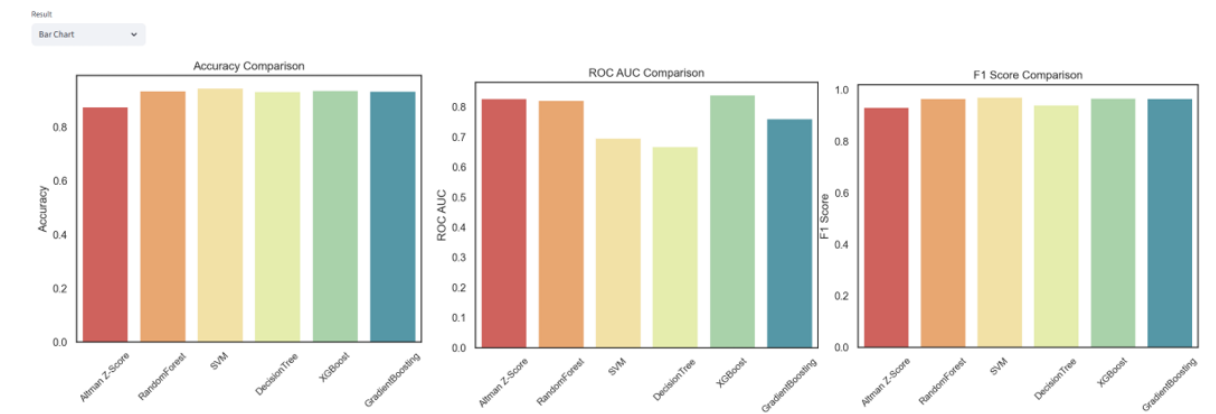


Рисунок 3.32 – Столпчикові діаграми порівняння моделей

Result

Graphs|

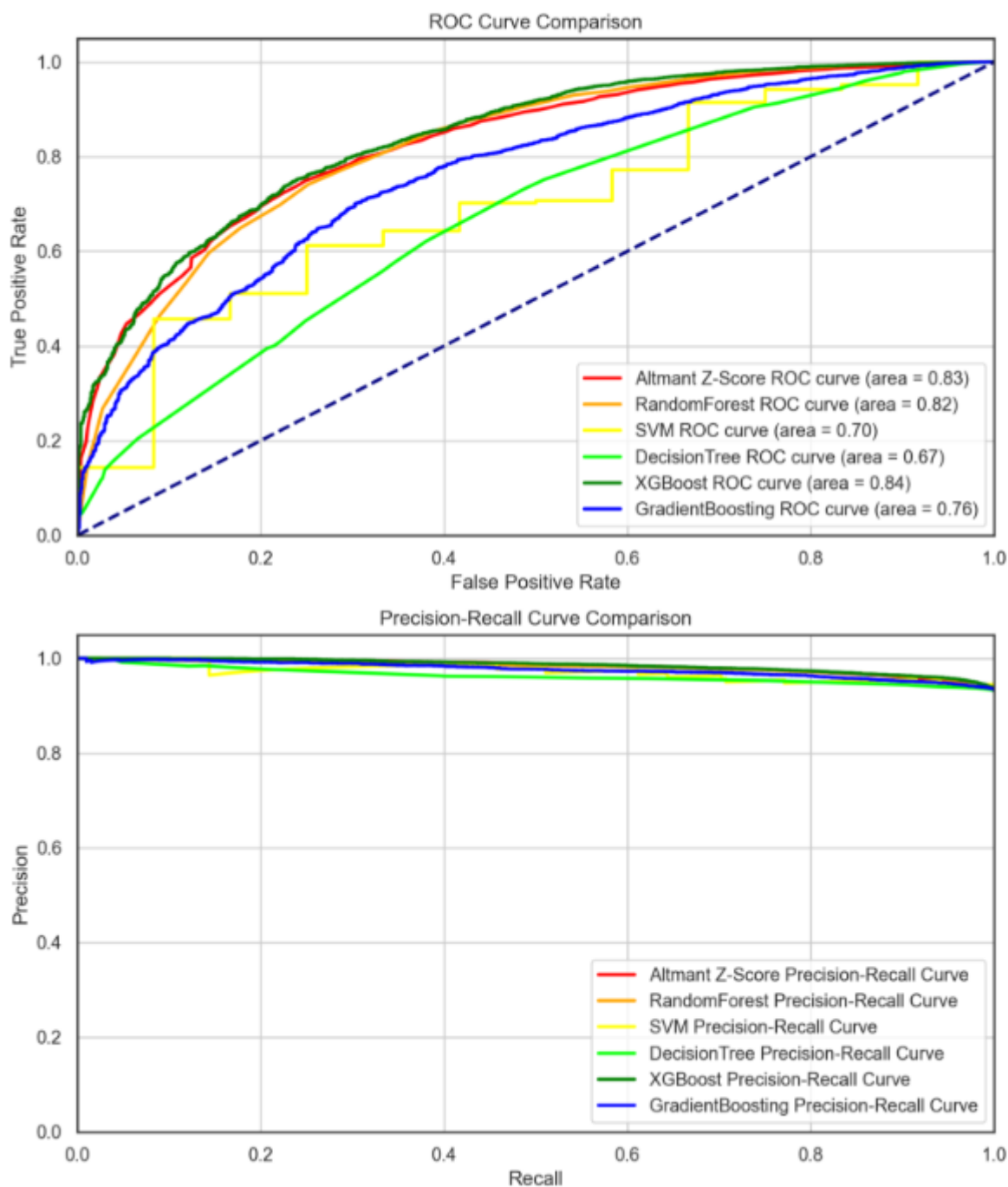


Рисунок 3.33 – Накладені ROC і Precision–Recall-криві

Модуль прогнозування та інтерпретації результатів відповідає за застосування навчених моделей до нових об'єктів і формування зрозумілих для користувача пояснень. У вікні прогнозування користувач обирає одну з попередньо збережених моделей (за замовчуванням — рекомендовану

XGBoost), завантажує або вводить фінансові показники окремої компанії, після чого система автоматично виконує ті самі процедури масштабування та перетворення ознак, які використовувалися на етапі навчання (рис. 3.34). Паралельно запускається підмодуль інтерпретації, що обчислює значення SHAP для кожної ознаки, розкладаючи індивідуальний прогноз на внески окремих факторів (рис. 3.35). Це дає змогу ідентифікувати ключові драйвери ризику для конкретного підприємства та подати їх у зручному для аналізу вигляді.

	year	X1	X2	X3	X4	X5	X6	X7	X8	X9
0	2016	1006.4939	845.0214	146.6846	173.0316	363.7101	144.2582	502.3378	19.7519	2948.71

Select Model
 XGBoost ▼

Predict

value
 1

Рисунок 3.34 – Приклад прогнозування методом XGBoost

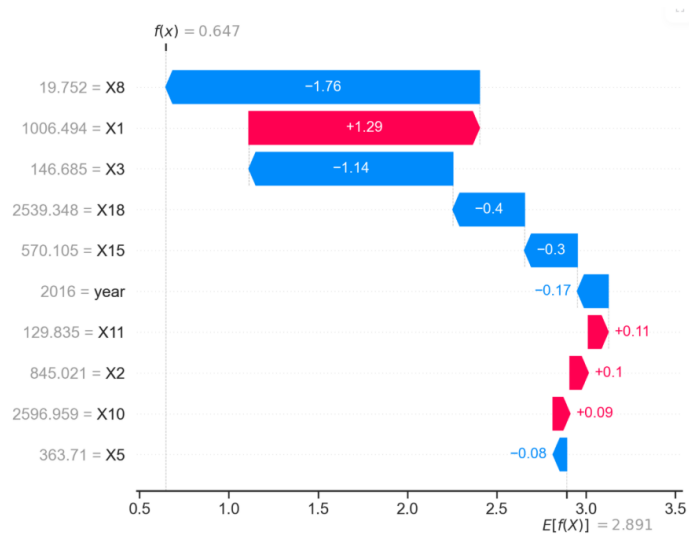


Рисунок 3.35 – SHAP інтерпретація

Логіка роботи користувача з ІСППР організована як послідовність інтуїтивно зрозумілих кроків. Спочатку завантажуються набір фінансових даних компаній, виконується їх базова перевірка й розвідувальний аналіз із використанням графіків співвідношення класів, кореляційної структури та викидів. Далі користувач формує вибірки для конкретних моделей, обираючи істотні ознаки та застосовуючи процедури балансування. Після цього проводиться навчання набору моделей і аналізуються отримані метрики та графіки, що дозволяє відібрати оптимальні алгоритми. На завершальному етапі обрана модель застосовується до нових компаній, а система генерує як прогноз імовірності банкрутства, так і його інтерпретацію, яку користувач може використовувати для ухвалення рішення щодо кредитування, реструктуризації боргу чи перегляду умов співпраці.

Програмна реалізація ІСППР базується на стеку технологій, що був обґрунтований на етапі аналізу засобів розробки: як мову реалізації обрано Python, для маніпуляції табличними даними використано бібліотеки pandas та NumPy, для побудови моделей — платформи машинного навчання з підтримкою ансамблевих алгоритмів (зокрема, реалізації XGBoost та бібліотеки scikit-learn), для балансування класів — пакет imbalanced-learn, а для інтерпретації — бібліотеку SHAP. Візуалізація даних та результатів моделювання здійснюється за допомогою спеціалізованих бібліотек побудови графіків. Такий вибір інструментів забезпечує як високу гнучкість і продуктивність, так і можливість подальшого масштабування системи та інтеграції її в існуючу IT-інфраструктуру фінансових установ.

3.5 Висновки до розділу 3

Узагальнено результати аналізу та підготовки вхідних фінансових даних. Встановлено, що вихідний датасет підприємств характеризується суттєвою

незбалансованістю класів, різномірністю масштабів показників, наявністю мультиколінеарності та викидів. Обґрунтовано застосування комплексу процедур попередньої обробки, до якого віднесено очищення даних, масштабування ознак, коректне розбиття на навчальну та тестову вибірки, а також використання алгоритму SMOTE для балансування класів. Показано, що така послідовність дій підвищує чутливість моделей до класу банкрутів і створює коректну основу для подальшого моделювання ризиків.

Розв'язано задачу відбору значущих характеристик для побудови моделей прогнозування економічних ризиків. На основі поєднання рекурсивного виключення ознак, оцінок важливості у деревоподібних ансамблях та аналізу внеску ознак за допомогою SHAP сформовано компактні підмножини фінансових показників для різних алгоритмів. Виявлено стабільне «ядро» інформативних змінних (зокрема X8, X15, X3, X1 та показник року спостереження), які систематично входять до складу найважливіших ознак. Зменшення розмірності простору ознак дозволило знизити ризик перенавчання, підвищити стійкість моделей і покращити інтерпретованість одержуваних результатів.

Виконано побудову та порівняння низки моделей прогнозування банкрутства підприємств, серед яких традиційна фінансова модель Altman Z-Score, дерево рішень, випадковий ліс, градієнтний бустинг, XGBoost та метод опорних векторів. Показано, що всі моделі машинного навчання істотно перевищують за якістю традиційну скорингову модель, зокрема за рахунок кращого врахування нелінійних взаємодій між фінансовими показниками та ефективнішого використання збалансованої вибірки. Встановлено, що SVM досягає найвищих значень F1-міри, тоді як XGBoost та Random Forest забезпечують найкращий компроміс між точністю, повнотою й розрізнявальною здатністю за ROC- та Precision–Recall-кривими. Особливу роль у покращенні результатів відіграло застосування SMOTE, яке дозволило суттєво зменшити кількість пропусків банкрутств при збереженні прийняттого рівня хибних спрацьовувань.

Розроблено функціональну схему та програмну реалізацію інтелектуальної системи підтримки прийняття рішень для аналізу економічних ризиків. Сформовано модульну структуру, що охоплює завантаження й валідацію даних, розвідувальний аналіз та візуалізацію, підготовку вибірок (масштабування, балансування, відбір ознак), навчання й порівняння моделей, а також модуль прогнозування та інтерпретації результатів. Обґрунтовано вибір XGBoost як базової моделі прогнозування в системі з огляду на високу розрізнявальну здатність, робастність до особливостей фінансових даних і сумісність із SHAP-аналізом. Реалізовано механізми формування ймовірнісних оцінок ризику та пояснень на рівні окремих підприємств, що підвищує прозорість прийняття управлінських рішень.

Отримані результати на етапі побудови моделей та розробки ІСППР формують цілісний прикладний інструментарій для діагностики економічних ризиків, який поєднує сучасні методи машинного навчання, засоби роботи з незбалансованими даними та технології пояснюваного штучного інтелекту.

РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП ПРОЕКТУ

В умовах постійної економічної нестабільності, спричиненої військовою агресією, розривом глобальних логістичних ланцюгів та волатильністю валютних ринків, питання забезпечення фінансової стійкості бізнесу набуває екзистенційного значення для економіки України. Традиційні підходи до оцінки ризиків банкрутства, такі як класичний Z-показник Е. Альтмана або коефіцієнтний аналіз Бівера, що були розроблені на даних середини минулого століття, демонструють зниження прогностичної ефективності в сучасних умовах цифрової економіки. Сучасний фінансовий ландшафт вимагає інструментів, здатних не лише констатувати поточний стан підприємства, але й виявляти приховані нелінійні закономірності, що передують дефолту, з високою якістю та швидкістю.

Розроблена в рамках магістерської дисертації «Інтелектуальна система підтримки прийняття рішень для аналізу економічних ризиків» (далі — ІСППР) пропонує інноваційне вирішення цієї проблеми. Використовуючи ансамблеві методи машинного навчання, зокрема XGBoost, який продемонстрував найвищу розрізнявальну здатність (ROC AUC ~ 0.85), та технології інтерпретації «чорної скриньки» (Explainable AI) на базі значень Шеплі (SHAP), система дозволяє мінімізувати ризики прийняття помилкових управлінських рішень. Цей розділ присвячено комплексному маркетинговому та економічному обґрунтуванню запуску ІСППР як повноцінного стартап-продукту на ринку B2B-рішень.

4.1 Опис ідеї стартап-проекту

В основі стартап-проекту лежить створення інтелектуальної екосистеми для прогнозування фінансової стійкості підприємств, яка базується на синергії передових методів машинного навчання та прозорості прийняття рішень. На відміну від класичних довідкових систем, що надають лише історичні дані («що сталося»), пропонований продукт відповідає на питання «що станеться?» та «чому це станеться?».

Сутність продукту полягає у хмарній платформі (SaaS), яка дозволяє користувачеві завантажити фінансову звітність (баланс, звіт про фінансові результати) контрагента у стандартизованому форматі (XML/XBRL/Excel). Система автоматично проводить попередню обробку даних, генерує синтетичні ознаки, виконує прогноз ймовірності банкрутства на горизонті 6-12 місяців та, що найважливіше, надає детальний звіт про фактори впливу.

У таблиці 4.1 наведено інформаційну карту стартап-проекту.

Таблиця 4.1 – Інформаційна карта стартап-проекту

Характеристика	Опис змісту
Назва проекту	«RiskSense AI» (Інтелектуальна система оцінки економічних ризиків)
Автори проекту	Мельник Микола Сергійович
Коротка анотація	SaaS-рішення для автоматизованого прогнозування ймовірності банкрутства підприємств із застосуванням ансамблевих методів машинного навчання (XGBoost, Random Forest) та пояснювального штучного інтелекту (SHAP).
Термін реалізації	12 місяців (від прототипу до виходу на ринок)

Продовження таблиці 4.1

Необхідні ресурси	Технологічні: хмарна інфраструктура (AWS/Azure), Python-бібліотеки (Scikit-learn, XGBoost, SHAP, Imbalanced-learn). Людські: Data Scientist, Backend/Frontend розробники, спеціаліст з B2B продажів. Фінансові: бюджет на розробку MVP, маркетинг та юридичний супровід (~\$30.000 - \$50.000 на перший рік).
Опис проблеми	Низька якість застарілих скорингових моделей (Альтман, Таффлер) в умовах війни; «чорна скринька» нейромереж, що унеможлиблює їх використання в регульованому банкінгу; висока вартість помилки при кредитуванні.
Цілі та завдання	Створити доступний інструмент, що забезпечує якість прогнозування ($AUC > 0.85$) та повну інтерпретованість результатів, знижуючи операційні витрати на ризик-менеджмент на 30-40%.
Очікувані результати	Функціонуючий веб-сервіс з API; підтверджена валідація на реальних даних українських компаній; залучення 50+ корпоративних клієнтів протягом першого року.

4.2 Технологічний аудит ідеї проекту

Технологічна досконалість є фундаментом конкурентоспроможності «RiskSense AI». У таблиці 4.2 наведено опис ідеї стартапу крізь призму технологічних інновацій та вигод для користувача.

Таблиця 4.2 – Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення комплексної системи, яка приймає фінансові дані, обробляє їх методами ML (балансування SMOTE, відбір ознак RFE) та повертає прогноз з поясненням.	Кредитний скоринг: Автоматизація прийняття рішень про видачу кредитів юридичним особам.	Зниження рівня NPL (проблемних кредитів), прискорення процесу розгляду заявок з днів до хвилин.
	Перевірка контрагентів: Моніторинг фінансового здоров'я постачальників та клієнтів.	Уникнення дебіторської заборгованості, раннє попередження про проблеми партнерів («Early Warning System»).
	Інвестиційний аналіз: Оцінка ризиків при купівлі корпоративних облігацій або M&A угодах.	Об'єктивна, незалежна оцінка активу, виявлення прихованих фінансових проблем, що не видно у стандартних звітах.

Ринок України представлений сильними гравцями у сфері агрегації даних, проте ніша глибокої предиктивної аналітики залишається менш

освоєною. Проведемо аналіз основних конкурентів: YouControl, Opendatabot та GiniMachine (таблиця 4.3).

Таблиця 4.3 – Порівняльний аналіз конкурентів проекту

№ п/ п	Техніко- економічні характеристи ки	Власний проект («RiskSense AI»)	YouContr ol (FinScore)	Opendata bot	GiniMach ine	W	N	S
1	Методологія прогнозуванн я	Ансамблі (XGBoost, RF) + SMOTE для дисбаланс у	Власні алгоритм и FinScore (переваж но фінансові коефіцієн ти та статистик а)	Базова аналітика на основі відкритих реєстрів, монітори нг статусів	AI/ML платформ а для побудови кастомни х моделей користув ачем			+
2	Інтерпретова ність (XAI)	Висока (SHAP values для кожного прогнозу)	Середня (пояснен ня факторів, але часто без кількісної ваги впливу)	Низька (констата ція фактів: борги, суди)	Висока (інструме нти валідації та звітності, діаграми важливос ті ознак)			+

Продовження таблиці 4.3.

3	Вимоги до даних клієнта	Мінімальні (лише фін. звітність поточної компанії)	Не потрібні (використовує власну базу даних)	Не потрібні (використовує відкриті реєстри)	Високі (клієнт має завантажити історичний датасет для навчання моделі)		+	
4	Цінова політика	Доступна (SaaS, Freemium для тесту)	Середня/Висока (корпоративна підписка)	Низька/Середня (доступні тарифи, API за запит)	Висока (від €100/міс, орієнтація на середній/великий фінтех)			+
5	Якість на незбалансованих даних	Висока (спеціалізоване застосування SMOTE)	Невідома (закриті алгоритми)	Н/Д (не є предиктивним інструментом)	Висока (автоматичний підбір моделей)			+

Аналіз:

1. YouControl є безперечним лідером у наданні даних (95% банків України є клієнтами). Проте їхній FinScore часто сприймається як індикатор

«загального здоров'я», а не як якісний інструмент прогнозування дефолту для конкретної угоди.

2. GiniMachine пропонує потужний інструмент, але він працює як «конструктор»: клієнт мусить мати власні історичні дані («bad/good loans») для навчання. Це створює бар'єр для нових компаній або тих, хто не має якісних архівів. «RiskSense AI» пропонує попередньо навчені моделі, що вирішує проблему «холодного старту».

3. Opendatabot фокусується на юридичних ризиках (суди, санкції), що робить його ідеальним партнером для інтеграції, а не прямим конкурентом у ніші фінансового моделювання.

Реалізація проекту є цілком реальною з використанням сучасного стеку технологій Python, що підтверджується результатами дисертаційного дослідження (таблиця 4.4).

Таблиця 4.4 – Технологічна здійсненність продукту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність
1	Створення комплексної системи прогнозування на основі фінансових даних	Використання мови програмування Python.	Наявні	Доступні
2		Використання мови програмування Julia.	Наявні	Доступні
3		Використання мови програмування Rust.	Наявні, необхідні допрацювання	Обмежено доступні
Обрана технологія реалізації ідеї проекту: Python				

Обрана технологія (Python + Boosting + SHAP) є золотим стандартом у сучасному Data Science, що гарантує наявність документації, спільноти та можливостей для масштабування.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Ринок програмного забезпечення для управління фінансовими ризиками демонструє стійке зростання. Глобальний обсяг ринку у 2024 році оцінюється в 22,35 млрд доларів США, і очікується його зростання до 49,33 млрд доларів до 2035 року. В Україні попит на такі системи підігрівається кризовими явищами: лише за перше півріччя 2024 року було зареєстровано понад 18 000

нових компаній, водночас кількість банкрутств та закриттів бізнесу залишається високою, що вимагає від контрагентів ретельної перевірки партнерів. В таблиці 4.5 характеризувано потенційний ринок стартап-проекту.

Таблиця 4.5 – Характеристика потенційного ринку стартап-проекту

№ п/п	Показники ринку	Характеристика
1	Кількість головних гравців (Україна)	3-5 (YouControl, Opendatabot, Contr Agent, GiniMachine)
2	Загальний обсяг ринку (Україна)	Оціночно \$15-20 млн (сегмент бізнес-аналітики та перевірки ризиків)
3	Динаміка ринку	Позитивна, зростаюча (CAGR ~15% глобально 6). В Україні драйвером є ріст NPL та вимоги комплаєнсу.
4	Наявність обмежень для входу	Середні. Потреба у довірі до алгоритмів, доступ до якісних даних, конкуренція з брендами.
5	Специфічні вимоги	Відповідність Постанові НБУ №351 (оцінка кредитного ризику), захист персональних даних (GDPR).
6	Середня норма рентабельності	Висока (SaaS-модель дозволяє маржинальність 70-80% після покриття R&D витрат).

Цільова аудиторія проекту є гетерогенною, охоплюючи як фінансові інституції, так і реальний сектор економіки (таблиця 4.6).

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба ринку	Цільова аудиторія	Особливості поведінки	Вимоги до товару
1	Мінімізація кредитних ризиків та NPL	Банки, кредитні спілки, лізингові компанії	Консервативні, вимагають валідації моделей, готові платити за зменшення резервів під збитки.	Висока якість, прозорість (SHAP), інтеграція в API, безпека даних.
2	Перевірка надійності постачальників /клієнтів	Великий та середній бізнес (Retail, Agro, Industry)	Шукають швидкі та зрозумілі інструменти («червона/зелена зона»). Масові перевірки.	Простота інтерфейсу, швидкість, пакетна обробка даних, доступна ціна.
3	Аналіз інвестиційної привабливості	Венчурні фонди, інвестиційні компанії	Фокус на динаміці та потенціалі зростання, а не тільки на ризиках.	Глибина аналізу, виявлення прихованих трендів, сценарний аналіз.
4	Незалежний аудит	Аудиторські та консалтингові фірми	Використовують як допоміжний інструмент для підтвердження експертних висновків.	Можливість експорту детальних звітів, обґрунтування прогнозів.

Успіх проекту залежить від здатності адаптуватися до зовнішніх загроз (таблиця 4.7).

Таблиця 4.7 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція з боку гігантів	YouControl або Opendatabot мають величезні бюджети та бази даних. Вони можуть скопіювати ML-функціонал.	Фокусування на ніші глибокої фінансової аналітики (vertical SaaS) на відміну від широкої агрегації даних конкурентами.
2	Недовіра до AI	Банківські ризик-менеджери можуть скептично ставитися до прогнозів «чорної скриньки», віддаючи перевагу ручному аналізу.	Акцент на Explainable AI (SHAP). Проведення безкоштовних пілотів для демонстрації якості на ретроспективних даних клієнта.
3	Якість даних	Фінансова звітність українських компаній часто викривлена податковою оптимізацією, що знижує якість моделей.	Використання робастних алгоритмів (Random Forest), стійких до викидів. Інтеграція нефінансових показників.

Успіх проекту також залежить від здатності використовувати ринкові можливості (таблиця 4.8).

Таблиця 4.8 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Євроінтеграція	Впровадження стандартів Basel III/IV вимагає від банків більш досконалих систем оцінки ризиків.	Позиціонування продукту як інструменту для відповідності європейським регуляторним нормам.
2	Ріст популярності AI	Ажіотаж навколо III робить клієнтів більш відкритими до інновацій. 22% фінтех-компаній вже використовують AI.	Активне використання термінів «Artificial Intelligence», «Machine Learning» у маркетингу як маркерів інноваційності.
3	API Economy	Зростання попиту на вбудовані фінанси та автоматизацію через API.	Розробка гнучкого API для легкої інтеграції в CRM та ERP системи клієнтів.

Конкурентна ситуація на ринку є напруженою, але має ознаки олігополії у сегменті даних (YouControl, Opendatabot) та монополістичної конкуренції у сегменті інструментів аналізу (таблиця 4.9).

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	Характеристика	Вплив на діяльність стартапу
1. Тип конкуренції	Олігополія (лідери YouControl, Opendatabot контролюють значну частку ринку агрегації даних).	Необхідність чіткої диференціації (нішування) у сегменті прогностичної аналітики, де конкуренція менша.
2. Рівень боротьби	Національний, з потенціалом виходу на міжнародний рівень (через універсальність ML-алгоритмів).	Орієнтація на український ринок як тестовий майданчик («пісочницю») перед масштабуванням в ЄС.
3. Галузева ознака	Міжгалузева (FinTech, LegalTech, RegTech).	Можливість залучення клієнтів з різних секторів (банки, агро, рітейл).
4. Характер переваг	Нецінова (якість прогнозу, інтерпретованість) та технологічна.	Інвестиції в R&D для підтримки технологічної переваги (якість моделей).

Далі виконаємо аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера (таблиця 4.10).

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товарозамінники
	Інші існуючі системи та продукти	Якість, ціни, кількість користувачів, капіталовкладення	Фактори сили постачальників	Контроль якості, порівняння цін	Сила бренду, якість, ціна, масштаби
Висновки	Конкуренція висока, але є ніша для ХАІ (пояснюваного ІІІ)	Ринок привабливий для нових гравців, загроза реальна	Критична залежність від відкритості державних даних	Необхідно пропонувати чітку фінансову вигоду (ROI), щоб переконати клієнтів	Загроза знижується, оскільки старі методи втрачають ефективність у кризі

На основі проведеного аналізу виділено ключові фактори успіху (таблиця 4.11).

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор	Обґрунтування
1	Якість прогнозування	XGBoost забезпечує AUC ~ 0.85 , що перевершує традиційні методи. ⁴ Це головна цінність для ризик-менеджера (гроші).
2	Пояснюваність	Використання SHAP дозволяє пояснити кожне рішення. Це знімає бар'єр недовіри до «чорної скриньки» AI.
3	Доступність та простота	Відсутність потреби у власному штаті Data Scientists (на відміну від GiniMachine). SaaS модель («бери і користуйся»).
4	Ціна	Гнучке ціноутворення (Freemium, підписка) робить продукт доступним для МСБ, на відміну від дорогих Enterprise рішень.

Тепер проведемо порівняльний аналіз сильних та слабких сторін (таблиця 4.12).

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	1	2	3
1	Якість прогнозування					+			
2	Пояснюваність						+		
3	Доступність та простота						+		
4	Ціна								+

Також проведемо SWOT-аналіз стартап проекту (таблиця 4.13)

Таблиця 4.13 – SWOT-аналіз стартап-проекту

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)
- Висока якість (AUC 0.85) та F1 Score (0.96).4 - Використання передових технологій (SHAP, SMOTE). - Масштабована SaaS архітектура. - Низька собівартість транзакції.	- Відсутність бренду та історії на ринку. - Залежність від якості даних, що вводяться користувачем. - Обмежені маркетингові ресурси. - Вузька спеціалізація (лише фінансові ризики).
Можливості (Opportunities)	Загрози (Threats)
- Зростання ринку FinTech та попиту на Risk Management. - Партнерство з CRM-системами та банками. - Вихід на ринки ЄС. - Розширення функціоналу (ESG-скоринг).	- Агресивна реакція конкурентів (YouControl). - Зміна законодавства щодо доступу до даних. - Кібератаки. - Економічна стагнація, що зменшить платоспроможність клієнтів.

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки (таблиця 4.14).

Таблиця 4.14 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Enterprise-рішення для банків: Прямі продажі, кастомна інтеграція, високий чек.	Низька	12-18 місяців
2	SaaS для МСБ: Масовий продукт, самообслуговування, низький чек, маркетинг через інтернет.	Висока	4-6 місяців
3	Партнерство (White Label): Інтеграція нашого скорингу в існуючі ERP/CRM системи (як додатковий модуль).	Середня	6-9 місяців

4.4 Розроблення ринкової стратегії стартап-проекту

Для ефективного виходу на ринок обрано стратегію фокусування на нішах, де потреба в якісній оцінці ризиків є найбільш гострою, а існуючі рішення є або занадто дорогими, або недостатньо технологічними.

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати цільову аудиторію проекту (таблиця 4.15).

Таблиця 4.15 – Вибір цільових груп споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів в сприйняти продукт	Орієнтовний попит у межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Фінансові установи	Висока	35%	Висока	Низька
2	Великий бізнес	Середня	20%	Середня	Середня
3	Малий та середній бізнес	Середня	30%	Низька	Висока
4	Державні органи та аудитори	Низька	10%	Низька	Низька
Які цільові групи обрано: 1 та 3.					

Визначимо базову стратегію розвитку (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
1	SaaS для МСБ та фінкомпаній	Диференційований маркетинг	Якість прогнозу, Прозорість (SHAP), Доступність	Стратегія диференціації

Визначимо стратегію конкурентної поведінки (таблиця 4.17).

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих?	Чи буде компанія копіювати основні характеристики товару конкурента?	Стратегія конкурентної поведінки
Ні	Так	Ні	Виклику лідеру та Нішева стратегія

Визначимо стратегію позиціонування (таблиця 4.18).

Таблиця 4.18 – Визначення стратегії позиціонування

Вимоги аудиторії	Конкурентні переваги	Позиціонування
Якість, прозорість, швидкість.	XGBoost (якість), SHAP (пояснення), Web/API.	«RiskSense AI» — це ваш персональний AI-аналітик, який не просто оцінює ризик, а пояснює його причини. Якість інституційного рівня, доступна кожному бізнесу.

4.5 Розроблення маркетингової програми стартап-проекту

Після проведеного комплексного аналізу, можемо повноцінно описати ключові переваги концепції потенційного товару (таблиця 4.19).

Таблиця 4.19 – Ключові переваги концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Уникнення збитків	Попередження про банкрутство.	Вища якість (AUC 0.85) завдяки ML.
2	Розуміння причин	Візуалізація факторів впливу.	Інтерпретованість (SHAP), відсутня у «чорних скриньок».
3	Швидкість	Миттєвий результат.	Автоматизація аналізу.

Також побудуємо концепцію маркетингових комунікацій (таблиця 4.20).

Таблиця 4.20 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Пошук спеціалізованих, надійних рішень	B2B продажі (LinkedIn, прямі контакти) Профільні конференції (FinTech Forum) Спеціалізовані видання (DOU, Banker.ua)	Якість Інтерпретованість (ХАІ) Відповідність регуляціям	Довести, що система є надійною альтернативою застарілим моделям та забезпечує прозорість для аудиту	«RiskSense AI» — це прозорість та якість інституційного рівня. Зменшуйте резерви під ризики, розуміючи кожне рішення AI.
2	Пошук доступного та швидкого продукту	Пошукова реклама (Google Ads) Соціальні мережі (Facebook) Бізнес-спільноти та форуми	Простота Швидкість Безкоштовний старт (MVP)	Вселити впевненість, що професійний ризик-менеджмент доступний кожному без складних налаштувань	Перевірте надійність партнера за 1 хвилину. Безкоштовна перевірка фінансового здоров'я вашого контрагента.

4.6 Висновки до розділу 4

У даному розділі проведено детальне обґрунтування стартап-проекту «RiskSense AI».

Ринок Risk Management Software зростає, а в Україні існує гостра потреба в якісних інструментах оцінки в умовах кризи.

Використання XGBoost та SHAP забезпечує конкурентну перевагу за рахунок поєднання якості (AUC 0.85) та інтерпретованості, чого бракує багатьом конкурентам.

Обрано стратегію диференціації та «кидка виклику» технологічним лідерам шляхом пропонування доступного SaaS-рішення для МСБ та фінкомпаній, які раніше не мали доступу до таких технологій.

Проект має високі шанси на успіх за умови швидкого виходу на ринок з MVP та ефективної маркетингової кампанії, спрямованої на подолання недовіри до штучного інтелекту.

ВИСНОВКИ

У магістерській дисертації вирішено актуальне науково-прикладне завдання підвищення ефективності аналізу та прогнозування економічних ризиків суб'єктів господарювання шляхом розробки інтелектуальної системи підтримки прийняття рішень, що базується на сучасних методах машинного навчання та технологіях інтерпретації штучного інтелекту. Основні наукові та практичні результати, отримані в ході дослідження, полягають у наступному.

Проведено системний аналіз сучасного стану проблеми оцінки ризику банкрутства. Встановлено, що в умовах високої волатильності економічного середовища та структурних змін ринків класичні дискримінантні моделі (зокрема Z-рахунок Альтмана) втрачають свою ефективність через нездатність враховувати нелінійні взаємозв'язки між фінансовими показниками. Визначено, що перспективним напрямом є застосування методів машинного навчання, проте їх впровадження ускладнюється проблемами незбалансованості реальних фінансових даних та низькою інтерпретованістю результатів («проблема чорної скриньки»).

Обґрунтовано методологічний підхід до побудови та навчання прогностичних моделей. Доведено необхідність попередньої обробки даних, зокрема застосування техніки синтетичного семплінгу (SMOTE) для усунення критичного дисбалансу класів, що дозволило значно підвищити чутливість моделей до виявлення потенційних банкрутів. Використання методу рекурсивного виключення ознак (RFE) дозволило зменшити розмірність вхідних даних та виділити набір найбільш значущих фінансових індикаторів, що сприяло підвищенню швидкодії та узагальнюючої здатності системи.

Здійснено експериментальне дослідження та порівняльний аналіз п'яти моделей машинного навчання: Decision Tree, Random Forest, SVM, Gradient Boosting та XGBoost. Результати моделювання на тестовій вибірці засвідчили

перевагу ансамблевих методів над поодинокими алгоритмами та класичними підходами.

Встановлено, що модель SVM забезпечує найвищий показник F1-score (~ 0.96), демонструючи відмінний баланс точності та повноти.

Модель XGBoost показала найкращий результат за метрикою ROC AUC (~ 0.85), що свідчить про високу якість ранжування об'єктів за ступенем ризику. На основі аналізу отриманих метрик алгоритм XGBoost було обрано як базове ядро інтелектуальної системи, оскільки він забезпечує найкращу розрізнявальну здатність, що є критичним для задач кредитного скорингу та ранжування позичальників.

Розроблено архітектуру та програмну реалізацію інтелектуальної системи підтримки прийняття рішень (ІСППР), реалізовану у вигляді веб-застосунку. Система забезпечує повний цикл аналізу: від завантаження фінансової звітності до візуалізації результатів прогнозу. Важливою особливістю розробленої системи є інтеграція модуля пояснень на основі методу SHAP (SHapley Additive exPlanations). Це дозволяє користувачу отримувати не лише ймовірнісну оцінку банкрутства, а й деталізовану інформацію про вплив кожного фінансового показника на сформований прогноз, що забезпечує прозорість прийняття рішень та відповідає сучасним вимогам Explainable AI.

Розроблено стартап-проект «RiskSense AI» для комерціалізації результатів наукового дослідження. Проведено аналіз ринкового середовища, який підтвердив високий попит на доступні інструменти експрес-оцінки ризиків серед малого та середнього бізнесу (SaaS-рішення). Розроблено бізнес-модель проекту, визначено стратегію виходу на ринок та оцінено основні ризики впровадження. Технологічний аудит підтвердив здійсненність проекту на основі використаного стеку технологій (Python, XGBoost, Web-фреймворки).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Звіт про фінансову стабільність. Грудень 2023 року. Київ: Національний банк України, 2023. 50 с. URL: https://bank.gov.ua/admin_uploads/article/FSR_2023-H2.pdf (дата звернення: 20.11.2025).
2. Altman E. I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*. 1968. Vol. 23, No. 4. P. 589–609. DOI: <https://doi.org/10.1111/j.1540-6261.1968.tb00843.x>
3. Чібісова І. В. Методи оцінки і прогнозування банкрутства підприємств. Наукові праці Кіровоградського національного технічного університету. Економічні науки. 2012. Вип. 22(2). С. 389–394. URL: <https://dspace.kntu.kr.ua/items/611eeba8-86db-4963-b13f-ba47661e9ee3>
4. Lessmann S., Baesens B., Seow H.-V., Thomas L. C. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*. 2015. Vol. 247, Iss. 1. P. 124–136. DOI: <https://doi.org/10.1016/j.ejor.2015.05.030>
5. Barboza F., Kimura H., Altman E. I. Machine learning models and bankruptcy prediction. *Expert Systems with Applications*. 2017. Vol. 83. P. 405–417. DOI: <https://doi.org/10.1016/j.eswa.2017.04.006>
6. Мірошниченко І. В., Івлієва К. Г. Оцінювання кредитного ризику методами машинного навчання. *Ефективна економіка*. 2019. № 12. 8 с. URL: <http://www.economy.nayka.com.ua/?op=1&z=7513> (дата звернення: 20.11.2025).
7. Коломієць Ю. Ю., Кочорба В. Ю. Оцінювання кредитних ризиків у системі ризик-менеджменту банківських структур. *Бізнес Інформ*. 2024. № 1. С. 320–332. DOI: <https://doi.org/10.32983/2222-4459-2024-1-320-332>
8. Bitetto A., Cerchiello P., Filomeni S., Tanda A., Tarantino B. Machine learning and credit risk: Empirical evidence from small- and mid-sized businesses.

Socio-Economic Planning Sciences. 2023. Vol. 90. Art. 101746. DOI: <https://doi.org/10.1016/j.seps.2023.101746>

9. Nguyen H. H., Viviani J.-L., Jabeur S. B. Bankruptcy prediction using machine learning and Shapley additive explanations. *Review of Quantitative Finance and Accounting*. 2025. Vol. 65. P. 107–148. DOI: <https://doi.org/10.1007/s11156-023-01192-x>

10. Chen T., Guestrin C. XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, 2016. P. 785–794. DOI: <https://doi.org/10.1145/2939672.2939785>

11. Lundberg S. M., Lee S.-I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* 30 (NIPS 2017). 2017. P. 4765–4774. URL: <https://arxiv.org/abs/1705.07874>

12. Lemaître G., Nogueira F., Aridas C. K. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*. 2017. Vol. 18, No. 17. P. 1–5. URL: <http://jmlr.org/papers/v18/16-365.html>

13. Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*. 2002. Vol. 16. P. 321–357. DOI: <https://doi.org/10.1613/jair.953>

14. Guyon I., Weston J., Barnhill S., Vapnik V. Gene selection for cancer classification using support vector machines. *Machine Learning*. 2002. Vol. 46, Iss. 1–3. P. 389–422. DOI: <https://doi.org/10.1023/A:1012487302797>

15. Breiman L., Friedman J. H., Olshen R. A., Stone C. J. *Classification and Regression Trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984. 368 p. ISBN 978-0534980502.

16. Breiman L. Random forests. *Machine Learning*. 2001. Vol. 45, Iss. 1. P. 5–32. DOI: <https://doi.org/10.1023/A:1010933404324>

17. Cortes C., Vapnik V. Support-vector networks. *Machine Learning*. 1995. Vol. 20, No. 3. P. 273–297. DOI: <https://doi.org/10.1007/BF00994018>

18. Friedman J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*. 2001. Vol. 29, No. 5. P. 1189–1232. DOI: <https://doi.org/10.1214/aos/1013203451>
19. Fawcett T. An introduction to ROC analysis. *Pattern Recognition Letters*. 2006. Vol. 27, No. 8. P. 861–874. DOI: <https://doi.org/10.1016/j.patrec.2005.10.010>
20. Saito T., Rehmsmeier M. The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*. 2015. Vol. 10, No. 3. e0118432. DOI: <https://doi.org/10.1371/journal.pone.0118432>
21. US Company Bankruptcy Prediction Dataset: <https://www.kaggle.com/datasets/utkarshx27/american-companies-bankruptcy-prediction-dataset/data> (дата звернення: 20.11.2025).
22. Simon H. A. *The New Science of Management Decision*. New York: Harper & Row, 1960. 175 p.
23. Gorry G. A., Scott Morton M. S. A Framework for Management Information Systems. *Sloan Management Review*. 1971. Vol. 13, No. 1. P. 55–70. URL: <https://sloanreview.mit.edu/article/a-framework-for-management-information-systems/>
24. Turban E., Sharda R., Delen D. *Decision Support and Business Intelligence Systems*. 9th ed. Upper Saddle River, NJ: Pearson Education, 2011. 696 p. ISBN 978-0136107293.
25. Мельник М.С., Левенчук Л.Б. Інтелектуальна система підтримки прийняття рішень для аналізу економічних ризиків. Системні науки та інформатика: збірник доповідей IV Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 01-05 грудня 2025 року, Київ. К., НН ІПСА КПІ ім. Ігоря Сікорського, 2025, 6 с.
26. Бідюк П. І., Романенко В. Д., Тимощук О. Л. Аналіз часових рядів : навч. посіб. К.: НТУУ «КПІ», 2013. 600 с.
27. Levenchuk L., Tymoshchuk O., Huskova V., Bidyuk P. Uncertainties in data processing, forecasting and decision making. *System research and information*

technologies. 2023. № 3. Р. 66–80. DOI: <https://doi.org/10.20535/SRIT.2308-8893.2023.3.05>

28. Левенчук Л. Б. Система підтримки прийняття рішень для аналізу операційних ризиків : дис. ... д-ра філософії : 124 Системний аналіз / Левенчук Людмила Борисівна. Київ, 2023. 168 с. URL: <https://ela.kpi.ua/items/d414d54d-f532-4a49-a77b-807572b1c3e5> (дата звернення: 12.11.2025).

29. Левенчук Л. Б., Гуськова В. Г., Бідюк П. І. Ймовірнісне моделювання операційних ризиків. Наукові вісті КПП, 2021. № 3(133). С. 26–37. URL: <https://ela.kpi.ua/items/8c30c23d-d8e4-4ada-8c84-86c751e8501a> (дата звернення: 12.11.2025)

ДОДАТОК А. ЛІСТИНГУ КОДУ

Файл «imports.py»:

```
import streamlit as st
import numpy as np
import pandas as pd
import seaborn as sns
import joblib

from matplotlib import pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import LinearSVC, SVC
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_curve, auc,
precision_recall_curve, f1_score
from sklearn.exceptions import ConvergenceWarning
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectKBest, f_classif

import warnings
warnings.filterwarnings('ignore', category=ConvergenceWarning)
warnings.filterwarnings('ignore', category=UserWarning)

import shap
import xgboost as xgb
from lightgbm import LGBMClassifier
from sklearn.feature_selection import RFE
from imblearn.over_sampling import SMOTE
```

Файл «Data_Input.py»:

```
from imports import *

df = pd.read_csv('input/american_bankruptcy.csv')
st.dataframe(df)

def summary(df_summary):
    st.write(f'Data shape: {df_summary.shape}')
    summ = pd.DataFrame(df_summary.dtypes, columns=['Data Type'])
    desc = pd.DataFrame(df_summary.describe(include='all').transpose())
    summ['Minimum'] = desc['min'].values
    summ['Maximum'] = desc['max'].values
    summ['First value'] = df_summary.loc[0].values
    summ['Second value'] = df_summary.loc[1].values
```

```
summ['Third value'] = df_summary.loc[2].values
summ['Number of Null'] = df_summary.isnull().sum().values
summ['Number of unique'] = df_summary.nunique().values
```

```
return summ
```

```
st.dataframe(summary(df))
```

```
df['status_label'] = df['status_label'].map({'alive': 1, 'failed': 0})
st.dataframe(df.head())
```

Файл «1_Visualization.py»:

```
from imports import *
from Data_Input import df
```

```
def show_alive_fail_ratio(df_ratio):
```

```
    # Given counts and ratios
    alive_count = df_ratio['status_label'].value_counts()[1]
    failed_count = df_ratio['status_label'].value_counts()[0]
    total_count = alive_count + failed_count
    alive_ratio = alive_count / total_count
    failed_ratio = failed_count / total_count
```

```
    print(«Alive Ratio:", alive_ratio)
    print(«Failed Ratio:", failed_ratio)
```

```
    # Data for the pie chart
    labels = ['Alive', 'Failed']
    sizes = [alive_ratio, failed_ratio]
    colors = ['aquamarine', 'palevioletred']
    #explode = (0.1, 0) # Explode the first slice
```

```
    # Create a pie chart
    plt.figure(figsize=(5, 5))
    plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
    plt.axis('equal') # Equal aspect ratio ensures the pie chart is circular
```

```
    # Save the pie chart as a PNG image
    # plt.savefig('pie_chart.png', bbox_inches='tight', dpi=300)
```

```
    with col2:
        st.pyplot(plt)
```

```
def plot_correlation_matrix(df_matrix):
```

```
    # Define variables
    variables = ['X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11', 'X12', 'X13',
'X14', 'X15', 'X16', 'X17', 'X18']
    data = df_matrix[variables]

    # Compute correlation matrix
    correlation_matrix = data.corr()
```

```

# Mask upper triangle for better readability
mask = np.triu(np.ones_like(correlation_matrix, dtype=bool))

# Set plot size and font scale
plt.figure(figsize=(9, 7)) # Adjusted size for a more compact look
sns.set(font_scale=0.8) # Adjust font size relative to plot

# Create heatmap
ax = sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap='coolwarm',
    fmt='%.2f',
    linewidths=0.3, # Thinner lines for better aesthetics
    mask=mask,
    cbar=True # Keeps color bar for reference
)

# Adjust title and layout
plt.title('Correlation Matrix', fontsize=12)
plt.xticks(rotation=45, ha="right", fontsize=10) # Rotate labels for readability
plt.yticks(fontsize=10)

with col2:
    st.pyplot(plt)

def plot_outliers(df_outliers):
    # Define numerical variables
    variables = ['X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11', 'X12', 'X13',
                'X14', 'X15', 'X16', 'X17', 'X18']

    # Compute IQR for outliers detection
    Q1 = df_outliers[variables].quantile(0.25)
    Q3 = df_outliers[variables].quantile(0.75)
    IQR = Q3 - Q1

    outliers = (df_outliers[variables] < (Q1 - 1.5 * IQR)) | (df_outliers[variables] > (Q3 + 1.5 *
IQR))

# Set plot size and font scale
plt.figure(figsize=(10, 6)) # More compact layout
sns.set(font_scale=0.9) # Adjust font size to match the new figure size

# Create box plot
ax = sns.boxplot(data=df_outliers[variables], linewidth=0.8, fliersize=3)

# Adjust aesthetics
plt.xticks(rotation=45, ha="right", fontsize=9) # Rotate for better readability
plt.yticks(fontsize=9)
plt.title('Box Plot of Numerical Features with Outliers', fontsize=12)

```

```

plt.xlabel(«Features", fontsize=10)
plt.ylabel(«Values", fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7) # Add light grid lines for clarity

with col2:
    st.pyplot(plt)

col1, col2 = st.columns([3,7])

with col1:
    option = st.selectbox('Choose a graph to show:',
                          ('Alive to Fail ratio', 'Correlation Matrix', 'Outliers'), 0)

    option_button = st.button('Show')

if option_button:
    match option:
        case 'Alive to Fail ratio':
            show_alive_fail_ratio(df)
        case 'Correlation Matrix':
            plot_correlation_matrix(df)
        case 'Outliers':
            plot_outliers(df)

```

Файл «2_Data_Transformation.py»:

```

from imports import *
from Data_Input import df

def select_features_for_ml(X_train,y_train, df_analysis, model):

    with col2:
        st.write(«Selecting features to train « + model + « model»)

    if model == 'SVM':
        # Selected Features Using SVM = ['fyear', 'X1', 'X4', 'X5', 'X6', 'X11', 'X12', 'X13', 'X15',
'X16']

        selected_features = ['year', 'X1', 'X4', 'X5', 'X6', 'X11', 'X12', 'X13', 'X15', 'X16']

        _= «"""
        # Step 1: Initialize a Faster SVM Model
        model = SVC(kernel="linear", probability=True)

        # Step 2: Apply RFE with SVM (Select 10 Features)
        rfe = RFE(model, n_features_to_select=10, step=5) # step=5 speeds up the process
        rfe.fit(X_train, y_train) # Fit RFE on training data

        # Step 3: Get Selected Feature Names
        selected_features = X_train.columns[rfe.support_].tolist()
        «"""
        df_selected = df_analysis[['company_name', 'status_label'] + selected_features]

```

```

df_selected.to_csv('datasets/svm_selected_features_test_dataset.csv', index=False)

with col2:
    st.write(«Datasets saved to CSV files.»)

elif model == 'Random Forest':
    #Selected Features Using Random Forest: ['X8', 'X15', 'X3', 'X1', 'X17', 'X7', 'X13', 'X10',
'X6', 'X14']
    selected_features = ['X8', 'X15', 'X3', 'X1', 'X17', 'X7', 'X13', 'X10', 'X6', 'X14']

    _ = «"""
    # Random Forest or Decision Tree feature selection
    model_rf = RandomForestClassifier(n_estimators=100, random_state=42)
    model_rf.fit(X_train, y_train)

    feature_importance_rf = pd.DataFrame({
        «Feature": X_train.columns,
        «Importance": model_rf.feature_importances_
    }).sort_values(by="Importance", ascending=False)
    selected_features = feature_importance_rf['Feature'].head(10).tolist()
    «"""

    df_selected = df_analysis[['company_name', 'status_label'] + selected_features]
    df_selected.to_csv('datasets/rf_selected_features_test_dataset.csv', index=False)

    with col2:
        st.write(«Datasets saved to CSV files.»)

elif model == 'Decision Tree':
    # Selected Features Using Decision Tree: ['X8', 'X15', 'X3', 'X1', 'X17', 'X7', 'X13', 'X10',
'X6', 'X14']
    selected_features = ['X8', 'X15', 'X3', 'X1', 'X17', 'X7', 'X13', 'X10', 'X6', 'X14']

    _ = «"""
    # Random Forest or Decision Tree feature selection
    model_rf = RandomForestClassifier(n_estimators=100, random_state=42)
    model_rf.fit(X_train, y_train)

    feature_importance_rf = pd.DataFrame({
        «Feature": X_train.columns,
        «Importance": model_rf.feature_importances_
    }).sort_values(by="Importance", ascending=False)
    selected_features = feature_importance_rf['Feature'].head(10).tolist()
    «"""

    df_selected = df_analysis[['company_name', 'status_label'] + selected_features]
    df_selected.to_csv('datasets/dt_selected_features_test_dataset.csv', index=False)

    with col2:
        st.write(«Datasets saved to CSV files.»)

```

```

elif model == 'XGBoost':
    # Selected Features Using XGBoost: ['year', 'X8', 'X1', 'X11', 'X10', 'X3', 'X18', 'X15',
'X5', 'X2']

    selected_features = ['year', 'X8', 'X1', 'X11', 'X10', 'X3', 'X18', 'X15', 'X5', 'X2']

    _ = «"""
    # XGB feature selection
    model_xgb = xgb.XGBClassifier(n_estimators=100, random_state=42)
    model_xgb.fit(X_train, y_train)

    explainer = shap.Explainer(model_xgb, X_train)
    shap_values = explainer(X_train)

    # Get Mean Absolute SHAP Values for Each Feature
    shap_importance = pd.DataFrame({
        «Feature": X_train.columns,
        «SHAP Importance": abs(shap_values.values).mean(axis=0)
    }).sort_values(by="SHAP Importance", ascending=False)

    # Sort by Importance and Select Top 10 Features
    selected_features = shap_importance['Feature'].head(10).tolist()
    «"""
    df_selected = df_analysis[['company_name', 'status_label'] + selected_features]
    df_selected.to_csv('datasets/xgb_selected_features_test_dataset.csv', index=False)

    with col2:
        st.write(«Datasets saved to CSV files.»)

    #shap.summary_plot(shap_values, X_train)
    df_selected = df_analysis[['company_name', 'status_label'] + selected_features]

elif model == 'Gradient Boosting':
    # Selected Features Using Gradient Boosting: ['X15', 'X8', 'X5', 'X3', 'year', 'X7', 'X13',
'X2', 'X11', 'X1']

    selected_features = ['X15', 'X8', 'X5', 'X3', 'year', 'X7', 'X13', 'X2', 'X11', 'X1']

    _ = «"""
    # Gradient Boosting feature selection
    model_gb = LGBMClassifier(n_estimators=100, random_state=42, verbosity=-1)
    model_gb.fit(X_train, y_train)

    feature_importance_gb = pd.DataFrame({
        «Feature": X_train.columns,
        «Importance": model_gb.feature_importances_
    }).sort_values(by="Importance", ascending=False)

    selected_features = feature_importance_gb['Feature'].head(10).tolist()
    «"""
    df_selected = df_analysis[['company_name', 'status_label'] + selected_features]
    df_selected.to_csv('datasets/gb_selected_features_test_dataset.csv', index=False)

```

```

        with col2:
            st.write(«Datasets saved to CSV files.»)

    elif model == 'Altman Z-Score':
        # Selected Features Using Altman Z-Score:  ['Z_Score', 'X1', 'X14', 'X10', 'X15', 'X12',
        'X8', 'X9', 'year']

        df_analysis['Z_Score'] = df_analysis.apply(
            lambda row: 1.2 * ((row['X1'] - row['X14']) / row['X10']) + 1.4 * (row['X15'] / row['X10'])
                + 3.3 * (row['X12'] / row['X10']) + 0.6 * (row['X8'] / row['X14']) +
                1.0 * (row['X9'] / row['X10']), axis=1)
        selected_features = ['Z_Score', 'X1', 'X14', 'X10', 'X15', 'X12', 'X8', 'X9', 'year']
        df_selected = df_analysis[['company_name', 'status_label'] + selected_features]
        df_selected.to_csv('datasets/altman_selected_features_test_dataset.csv', index=False)

        with col2:
            st.write(«Datasets saved to CSV files.»)

    with col2:
        st.write(«Selected Features Using « + model + «: «, selected_features)

    return selected_features, df_selected

_ = '''
st.write(«Select features for ML:»)
col1, col2, col3 = st.columns(3)

with col1:
    button1 = st.button(«Random Forest»)
    button4 = st.button(«XGBoost»)

with col2:
    button2 = st.button(«Decision Tree»)
    button5 = st.button(«SVM»)
    button7 = st.button(«Select All»)

with col3:
    button3 = st.button(«Gradient Boosting»)
    button6 = st.button(«Altman Z-Score»)

if button1:
    selected_features_rf, df_rf = select_features_for_ml(X_pre, y_pre, df, 'Random Forest')

if button2:
    selected_features_dt, df_dt = select_features_for_ml(X_pre, y_pre, df, 'Decision Tree')

if button3:
    selected_features_gb, df_gb = select_features_for_ml(X_pre, y_pre, df, 'Gradient Boosting')

if button4:
    selected_features_xgb, df_xgb = select_features_for_ml(X_pre, y_pre, df, 'XGBoost')

```

```

if button5:
    selected_features_svm, df_svm = select_features_for_ml(X_pre, y_pre, df, 'SVM')

if button6:
    selected_features_alt, df_alt = select_features_for_ml(X_pre, y_pre, df, 'Altman Z-Score')

if button7:
    selected_features_rf, df_rf = select_features_for_ml(X_pre, y_pre, df, 'Random Forest')
    selected_features_dt, df_dt = select_features_for_ml(X_pre, y_pre, df, 'Decision Tree')
    selected_features_gb, df_gb = select_features_for_ml(X_pre, y_pre, df, 'Gradient Boosting')
    selected_features_xgb, df_xgb = select_features_for_ml(X_pre, y_pre, df, 'XGBoost')
    selected_features_svm, df_svm = select_features_for_ml(X_pre, y_pre, df, 'SVM')
    selected_features_alt, df_alt = select_features_for_ml(X_pre, y_pre, df, 'Altman Z-Score')

...

X_pre = df.drop(columns=['company_name', 'status_label'])
y_pre = df['status_label']

col1, col2 = st.columns([3, 7])

with col1:
    option = st.selectbox(
        'Select features for ML:',
        («Altman Z-Score», «RandomForest», «SVM»,
        «DecisionTree», «XGBoost», «GradientBoosting»,
        'Select All'), 4)
    option_button = st.button('Select')

if option_button:
    match option:
        case 'Random Forest':
            selected_features_rf, df_rf = select_features_for_ml(X_pre, y_pre, df, 'Random Forest')
        case 'XGBoost':
            selected_features_xgb, df_xgb = select_features_for_ml(X_pre, y_pre, df, 'XGBoost')
        case 'Decision Tree':
            selected_features_dt, df_dt = select_features_for_ml(X_pre, y_pre, df, 'Decision Tree')
        case 'Altman Z-Score':
            selected_features_alt, df_alt = select_features_for_ml(X_pre, y_pre, df, 'Altman Z-
Score')
        case 'SVM':
            selected_features_svm, df_svm = select_features_for_ml(X_pre, y_pre, df, 'SVM')
        case 'Gradient Boosting':
            selected_features_gb, df_gb = select_features_for_ml(X_pre, y_pre, df, 'Gradient
Boosting')
        case 'Select All':
            selected_features_rf, df_rf = select_features_for_ml(X_pre, y_pre, df, 'Random Forest')
            selected_features_dt, df_dt = select_features_for_ml(X_pre, y_pre, df, 'Decision Tree')
            selected_features_gb, df_gb = select_features_for_ml(X_pre, y_pre, df, 'Gradient
Boosting')

```

```

selected_features_xgb, df_xgb = select_features_for_ml(X_pre, y_pre, df, 'XGBoost')
selected_features_svm, df_svm = select_features_for_ml(X_pre, y_pre, df, 'SVM')
selected_features_alt, df_alt = select_features_for_ml(X_pre, y_pre, df, 'Altman Z-
Score')

```

Файл «3_Models.py»:

```

from imports import *
from Data_Input import df

def plot_confusion_matrix(ax, cm, method_name):
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", linewidths=0.5, square=True, ax=ax)
    ax.set_xlabel('Predicted Label')
    ax.set_ylabel('True Label')
    ax.set_title(method_name + ' Confusion Matrix')

def plot_roc_auc_curve(ax, fpr, tpr, roc_auc, method_name):
    ax.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = {:.2f})'.format(roc_auc))
    ax.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    ax.set_xlim([0.0, 1.0])
    ax.set_ylim([0.0, 1.05])
    ax.set_xlabel('False Positive Rate')
    ax.set_ylabel('True Positive Rate')
    ax.set_title(method_name + ' ROC Curve')
    ax.legend(loc="lower right")
    ax.grid(True)

def plot_precision_recall(ax, recall, precision, method_name):
    ax.plot(recall, precision, color='blue', lw=2)
    ax.set_xlim([0.0, 1.0])
    ax.set_ylim([0.0, 1.05])
    ax.set_xlabel('Recall')
    ax.set_ylabel('Precision')
    ax.set_title(method_name + ' Precision-Recall Curve')
    ax.grid(True)

def plot_all_metrics(cm, fpr, tpr, roc_auc, recall, precision, method_name):
    fig, axs = plt.subplots(1, 3, figsize=(18, 6)) # Adjust figsize as needed

    plot_confusion_matrix(axs[0], cm, method_name)
    plot_roc_auc_curve(axs[1], fpr, tpr, roc_auc, method_name)
    plot_precision_recall(axs[2], recall, precision, method_name)

    plt.tight_layout() # Adjust spacing
    st.pyplot(plt)

def altman_z_score_classify(df_classify):

    # Define features and target
    X = df_classify.drop(columns=['company_name', 'status_label'], axis=1)
    y = df_classify['status_label']

```

```

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Apply SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# Train a model
model = RandomForestClassifier(random_state=42)
model.fit(X_train_resampled, y_train_resampled)

joblib.dump(model, «models/alt.joblib», compress=3)

# Predictions
y_pred = model.predict(X_test)

# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
st.write(«Altman/Logistic Regression Accuracy: {:.2f}%».format(accuracy * 100))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
st.write(«Confusion Matrix:»)
st.write(cm)

# Classification Report
cr = classification_report(y_test, y_pred, digits=5, output_dict=True)
st.write(«Classification Report:»)
st.dataframe(cr)

# ROC Curve and AUC-ROC
y_scores = model.predict_proba(X_test)[: , 1]
fpr, tpr, thresholds = roc_curve(y_test, y_scores)
roc_auc = auc(fpr, tpr)

# Precision-Recall Curve
precision, recall, thresholds_pr = precision_recall_curve(y_test, y_scores)

# Calculate F1 Score for Altman
f1_score_altman = f1_score(y_test, y_pred)
cv_scores = cross_val_score(model, X, y, cv=5)
average_cv_score = cv_scores.mean()

# Plot and save confusion matrix, ROC-AUC curve, precision-recall
plot_all_metrics(cm, fpr, tpr, roc_auc, recall, precision, 'Altman Z-Score')

return roc_auc, accuracy, model, average_cv_score, f1_score_altman, fpr, tpr, precision, recall

```

```

def rf_classify(df_classify):
    # RandomForestClassifier
    X = df_classify.drop(columns=['company_name', 'status_label'], axis=1)
    y = df_classify['status_label']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    joblib.dump(model, «models/rf.joblib», compress=3)

    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    st.write(«Random Forest Accuracy: {:.2f}%».format(accuracy * 100))

    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    st.write(«Confusion Matrix:»)
    st.write(cm)

    # Classification Report
    cr = classification_report(y_test, y_pred, digits=5, output_dict=True)
    st.write(«Classification Report:»)
    st.dataframe(cr)

    # ROC Curve and AUC-ROC
    y_scores = model.predict_proba(X_test)[:, 1]
    fpr, tpr, thresholds = roc_curve(y_test, y_scores)
    roc_auc = auc(fpr, tpr)

    # Precision-Recall Curve
    precision, recall, thresholds_pr = precision_recall_curve(y_test, y_scores)

    #Cross-Validation Scores
    cv_scores = cross_val_score(model, X, y, cv=5)
    st.write(«\nCross-Validation Scores:»)
    for i, score in enumerate(cv_scores):
        st.write(«Fold {}: {:.2f}%».format(i + 1, score * 100))

    # Average Cross-Validation Score
    average_cv_score = cv_scores.mean()
    st.write(«Average Cross-Validation Score: {:.2f}%».format(average_cv_score * 100))

    # Calculate F1 Score for SVM
    f1_score_rf = f1_score(y_test, y_pred)

    # Plot and save confusion matrix, ROC-AUC curve, precision-recall
    plot_all_metrics(cm, fpr, tpr, roc_auc, recall, precision, 'Random Forest')

    return roc_auc, accuracy, model, average_cv_score, f1_score_rf, fpr, tpr, precision, recall

```

```

def decision_tree_classify(df_classify):
    # DecisionTreeClassifier
    X = df_classify.drop(columns=['company_name', 'status_label'], axis=1)
    y = df_classify['status_label']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model = DecisionTreeClassifier(random_state=42)
    model.fit(X_train, y_train)
    joblib.dump(model, «models/dt.joblib», compress=3)

    y_pred = model.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    st.write(«Decision Tree Accuracy: {:.2f}%».format(accuracy * 100))

    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    st.write(«Confusion Matrix:»)
    st.write(cm)

    # Classification Report
    cr = classification_report(y_test, y_pred, output_dict=True)
    st.write(«Classification Report:»)
    st.dataframe(cr)

    # Hyperparameters tuning using Grid Search
    param_grid = {
        'max_depth': [None, 5, 10, 15],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
        'criterion': ['gini', 'entropy']
    }

    dt_tuned = DecisionTreeClassifier(random_state=42)
    grid_search = GridSearchCV(dt_tuned, param_grid, cv=5)
    grid_search.fit(X_train, y_train)

    best_model = grid_search.best_estimator_
    best_pred = best_model.predict(X_test)
    best_accuracy = accuracy_score(y_test, best_pred)
    st.write(«Best Decision Tree Accuracy: {:.2f}%».format(best_accuracy * 100))

    # Confusion Matrix
    cm = confusion_matrix(y_test, best_pred)
    st.write(«Confusion Matrix:»)
    st.write(cm)

    # Classification Report

```

```

cr = classification_report(y_test, best_pred, digits=5, output_dict=True)
st.write(«Classification Report:»)
st.dataframe(cr)

# ROC Curve and AUC-ROC
y_scores = best_model.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_scores)
roc_auc = auc(fpr, tpr)

# Precision-Recall Curve
precision, recall, thresholds_pr = precision_recall_curve(y_test, y_scores)

# Calculate F1 Score for Decision Tree
f1_score_dt = f1_score(y_test, y_pred)
cv_scores = cross_val_score(best_model, X, y, cv=5)
average_cv_score = cv_scores.mean()

# Plot and save confusion matrix, ROC-AUC curve, precision-recall
plot_all_metrics(cm, fpr, tpr, roc_auc, recall, precision, 'Decision Tree')

return roc_auc, best_accuracy, best_model, average_cv_score, f1_score_dt, fpr, tpr, precision,
recall

def grad_boosting_classify(df_classify):
    # GradientBoostingClassifier
    X = df_classify.drop(columns=['company_name', 'status_label'], axis=1)
    y = df_classify['status_label']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,
stratify=y)

    model = GradientBoostingClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    joblib.dump(model, «models/gb.joblib», compress=3)

    y_pred = model.predict(X_test)

    # Calculate accuracy
    accuracy = accuracy_score(y_test, y_pred)
    st.write(«Gradient Boosting Accuracy: {:.2f}%».format(accuracy * 100))

    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    st.write(«Confusion Matrix:»)
    st.write(cm)

    # Classification Report
    cr = classification_report(y_test, y_pred, digits=5, output_dict=True)
    st.write(«Classification Report:»)
    st.dataframe(cr)

```

```

y_scores = model.predict_proba(X_test)[: , 1]
fpr, tpr, thresholds = roc_curve(y_test, y_scores)
roc_auc = auc(fpr, tpr)

# Precision-Recall Curve
precision, recall, thresholds_pr = precision_recall_curve(y_test, y_scores)

# Calculate Cross-Validation Scores for Gradient Boosting
cv_scores = cross_val_score(model, X, y, cv=5)
average_cv_score = cv_scores.mean()

# Calculate F1 Score for Gradient Boosting
f1_score_gb = f1_score(y_test, y_pred)

# Plot and save confusion matrix, ROC-AUC curve, precision-recall
plot_all_metrics(cm, fpr, tpr, roc_auc, recall, precision, 'Gradient Boosting')

return roc_auc, accuracy, model, average_cv_score, f1_score_gb, fpr, tpr, precision, recall

def xgboost_classify(df_classify):

    # Splitting the data
    X = df_classify.drop(columns=['company_name', 'status_label'], axis=1)
    y = df_classify['status_label']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # XGB feature selection
    model = xgb.XGBClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    joblib.dump(model, «models/xgb.joblib», compress=3)

    # Predictions
    y_pred = model.predict(X_test)

    # Accuracy
    accuracy = accuracy_score(y_test, y_pred)
    st.write(«XGBoost Accuracy: {:.2f}%».format(accuracy * 100))

    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    st.write(«Confusion Matrix:»)
    st.write(cm)

    # Classification Report
    cr = classification_report(y_test, y_pred, digits=5, output_dict=True)
    st.write(«Classification Report:»)
    st.dataframe(cr)

    # ROC Curve and AUC-ROC
    y_scores = model.predict_proba(X_test)[: , 1]

```

```

fpr_xgb, tpr_xgb, thresholds_xgb = roc_curve(y_test, y_scores)
roc_auc = auc(fpr_xgb, tpr_xgb)

# Precision-Recall Curve
precision_xgb, recall_xgb, thresholds_pr_xgb = precision_recall_curve(y_test, y_scores)

# Cross-Validation Scores
cv_scores_xgb = cross_val_score(model, X, y, cv=5)
st.write(«\nCross-Validation Scores:»)
for i, score in enumerate(cv_scores_xgb):
    st.write(«Fold {}: {:.2f}%».format(i + 1, score * 100))

# Average Cross-Validation Score
average_cv_score = cv_scores_xgb.mean()
st.write(«Average Cross-Validation Score: {:.2f}%».format(average_cv_score * 100))

y_scores = model.predict_proba(X_test)[: , 1]
fpr_xgb, tpr_xgb, thresholds_xgb = roc_curve(y_test, y_scores)
roc_auc = auc(fpr_xgb, tpr_xgb)

# Precision-Recall Curve
precision_xgb, recall_xgb, thresholds_pr_xgb = precision_recall_curve(y_test, y_scores)

# Calculate F1 Score for Altman
f1_score_xgb = f1_score(y_test, y_pred)

# Plot and save confusion matrix, ROC-AUC curve, precision-recall
plot_all_metrics(cm, fpr_xgb, tpr_xgb, roc_auc, recall_xgb, precision_xgb, 'XGBoost')

return roc_auc, accuracy, model, average_cv_score, f1_score_xgb, fpr_xgb, tpr_xgb, precision_xgb,
recall_xgb

def svm_classify(df_classify):
    df_classify = df_classify.sample(n=min(1000, len(df_classify)), random_state=42)

    X = df_classify.drop(columns=['company_name', 'status_label'], axis=1)
    y = df_classify['status_label']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y,
random_state=42)

    # Feature selection using SelectKBest
    selector = SelectKBest(score_func=f_classif, k=20) # Select top 20 features
    X_train_selected = selector.fit_transform(X_train, y_train)
    X_test_selected = selector.transform(X_test)

    # Dimensionality Reduction
    pca = PCA(n_components=10) # Reduce to 10 principal components
    X_train_pca = pca.fit_transform(X_train_selected)
    X_test_pca = pca.transform(X_test_selected)

```

```

svm_model = LinearSVC()
svm_model.fit(X_train_pca, y_train)
joblib.dump(svm_model, «models/svm.joblib", compress=3)

y_pred = svm_model.predict(X_test_pca)

accuracy = accuracy_score(y_test, y_pred)
st.write(«SVM Accuracy: {:.2f}%».format(accuracy * 100))

cm = confusion_matrix(y_test, y_pred)
st.write(«Confusion Matrix:»)
st.write(cm)

cr = classification_report(y_test, y_pred, digits=5, output_dict=True)
st.write(«Classification Report:»)
st.dataframe(cr)

y_scores = svm_model.decision_function(X_test_pca)
fpr, tpr, thresholds = roc_curve(y_test, y_scores)
roc_auc = auc(fpr, tpr)

precision, recall, thresholds_pr = precision_recall_curve(y_test, y_scores)
f1_score_svm = f1_score(y_test, y_pred)

cv_scores = cross_val_score(svm_model, X_train_pca, y_train, cv=3, n_jobs=-1) # Parallel CV
average_cv_score = cv_scores.mean()

# Assuming plot_all_metrics is defined elsewhere
plot_all_metrics(cm, fpr, tpr, roc_auc, recall, precision, 'SVM')

return roc_auc, accuracy, svm_model, average_cv_score, f1_score_svm, fpr, tpr, precision, recall

def save_metrics(name, roc_auc, accuracy, model, avg_cv_score, f1_score, fpr, tpr, precision, recall):
    summ = pd.read_csv('model_metrics.csv')

    col1, col2 = st.columns([3, 7])

    with col1:
        option = st.selectbox(
            'Select Models to train:',
            ('Altman Z-Score', «Random Forest", «SVM",
            «Decision Tree", «XGBoost", «Gradient Boosting»), 4)
        option_button = st.button('Train')

    _=''rf_roc_auc, rf_accuracy, rf_model, rf_avg_cv_score, rf_f1_score, rf_fpr, rf_tpr, rf_precision,
rf_recall = 0, 0, 0, 0, 0, 0, 0, 0, 0
    xgb_roc_auc, xgb_accuracy, xgb_model, xgb_avg_cv_score, xgb_f1_score, xgb_fpr, xgb_tpr,
xgb_precision, xgb_recall = 0, 0, 0, 0, 0, 0, 0, 0, 0
    dt_roc_auc, dt_accuracy, dt_model, dt_avg_cv_score, dt_f1_score, dt_fpr, dt_tpr, dt_precision,
dt_recall = 0, 0, 0, 0, 0, 0, 0, 0, 0

```

```

    alt_roc_auc, alt_accuracy, alt_model, alt_avg_cv_score, alt_f1_score, alt_fpr, alt_tpr,
    alt_precision, alt_recall = 0, 0, 0, 0, 0, 0, 0, 0, 0
    svm_roc_auc, svm_accuracy, svm_model, svm_avg_cv_score, svm_f1_score, svm_fpr, svm_tpr,
    svm_precision, svm_recall = 0, 0, 0, 0, 0, 0, 0, 0, 0
    gb_roc_auc, gb_accuracy, gb_model, gb_avg_cv_score, gb_f1_score, gb_fpr, gb_tpr, gb_precision,
    gb_recall = 0, 0, 0, 0, 0, 0, 0, 0, 0

    with col1:
        export_button = st.button('Export')

    if export_button:
        model_metrics = {
            'Model': ['Altman Z-Score', 'RandomForest', 'SVM', 'DecisionTree', 'XGBoost',
'GradientBoosting'],
            'Accuracy': [alt_accuracy, rf_accuracy, svm_accuracy, dt_accuracy, xgb_accuracy,
gb_accuracy],
            'ROC AUC': [alt_roc_auc, rf_roc_auc, svm_roc_auc, dt_roc_auc, xgb_roc_auc, gb_roc_auc],
            'F1 Score': [alt_f1_score, rf_f1_score, svm_f1_score, dt_f1_score, xgb_f1_score,
gb_f1_score],
            'Cross-Validation Score': [alt_avg_cv_score, rf_avg_cv_score, svm_avg_cv_score,
dt_avg_cv_score, xgb_avg_cv_score, gb_avg_cv_score]
        }
        summary_df = pd.DataFrame(model_metrics)
        summary_df.to_csv('model_metrics.csv', index=False)'''

    if option_button:
        match option:
            case 'Random Forest':
                rf_roc_auc, rf_accuracy, rf_model, rf_avg_cv_score, rf_f1_score, rf_fpr, rf_tpr,
rf_precision, rf_recall = rf_classify(pd.read_csv(«datasets/rf_selected_features_test_dataset.csv»))
                summ = pd.read_csv('model_metrics.csv')
                summ.loc[1] = [summ.loc[1, 'Model'], rf_accuracy, rf_roc_auc, rf_f1_score,
rf_avg_cv_score]
                summ.to_csv('model_metrics.csv', index=False)
                np.savetxt('metrics/fpr/rf.txt', rf_fpr)
                np.savetxt('metrics/tpr/rf.txt', rf_tpr)
                np.savetxt('metrics/precision/rf.txt', rf_precision)
                np.savetxt('metrics/recall/rf.txt', rf_recall)
            case 'XGBoost':
                xgb_roc_auc, xgb_accuracy, xgb_model, xgb_avg_cv_score, xgb_f1_score, xgb_fpr, xgb_tpr,
xgb_precision, xgb_recall =
xgboost_classify(pd.read_csv(«datasets/xgb_selected_features_test_dataset.csv»))
                summ = pd.read_csv('model_metrics.csv')
                summ.loc[4] = [summ.loc[4, 'Model'], xgb_accuracy, xgb_roc_auc, xgb_f1_score,
xgb_avg_cv_score]
                summ.to_csv('model_metrics.csv', index=False)
                np.savetxt('metrics/fpr/xgb.txt', xgb_fpr)
                np.savetxt('metrics/tpr/xgb.txt', xgb_tpr)
                np.savetxt('metrics/precision/xgb.txt', xgb_precision)
                np.savetxt('metrics/recall/xgb.txt', xgb_recall)
            case 'Decision Tree':

```

```

        dt_roc_auc, dt_accuracy, dt_model, dt_avg_cv_score, dt_f1_score, dt_fpr, dt_tpr,
dt_precision, dt_recall
        =
decision_tree_classify(pd.read_csv(«datasets/dt_selected_features_test_dataset.csv»))
        summ = pd.read_csv('model_metrics.csv')
        summ.loc[3] = [summ.loc[3, 'Model'], dt_accuracy, dt_roc_auc, dt_f1_score,
dt_avg_cv_score]
        summ.to_csv('model_metrics.csv', index=False)
        np.savetxt('metrics/fpr/dt.txt', dt_fpr)
        np.savetxt('metrics/tpr/dt.txt', dt_tpr)
        np.savetxt('metrics/precision/dt.txt', dt_precision)
        np.savetxt('metrics/recall/dt.txt', dt_recall)
    case 'Altman Z-Score':
        alt_roc_auc, alt_accuracy, alt_model, alt_avg_cv_score, alt_f1_score, alt_fpr, alt_tpr,
alt_precision, alt_recall
        =
altman_z_score_classify(pd.read_csv(«datasets/altman_selected_features_test_dataset.csv»))
        summ = pd.read_csv('model_metrics.csv')
        summ.loc[0] = [summ.loc[0, 'Model'], alt_accuracy, alt_roc_auc, alt_f1_score,
alt_avg_cv_score]
        summ.to_csv('model_metrics.csv', index=False)
        np.savetxt('metrics/fpr/alt.txt', alt_fpr)
        np.savetxt('metrics/tpr/alt.txt', alt_tpr)
        np.savetxt('metrics/precision/alt.txt', alt_precision)
        np.savetxt('metrics/recall/alt.txt', alt_recall)
    case 'SVM':
        svm_roc_auc, svm_accuracy, svm_model, svm_avg_cv_score, svm_f1_score, svm_fpr, svm_tpr,
svm_precision, svm_recall = svm_classify(pd.read_csv(«datasets/svm_selected_features_test_dataset.csv»))
        summ = pd.read_csv('model_metrics.csv')
        summ.loc[2] = [summ.loc[2, 'Model'], svm_accuracy, svm_roc_auc, svm_f1_score,
svm_avg_cv_score]
        summ.to_csv('model_metrics.csv', index=False)
        np.savetxt('metrics/fpr/svm.txt', svm_fpr)
        np.savetxt('metrics/tpr/svm.txt', svm_tpr)
        np.savetxt('metrics/precision/svm.txt', svm_precision)
        np.savetxt('metrics/recall/svm.txt', svm_recall)
    case 'Gradient Boosting':
        gb_roc_auc, gb_accuracy, gb_model, gb_avg_cv_score, gb_f1_score, gb_fpr, gb_tpr,
gb_precision, gb_recall
        =
grad_boosting_classify(pd.read_csv(«datasets/gb_selected_features_test_dataset.csv»))
        summ = pd.read_csv('model_metrics.csv')
        summ.loc[5] = [summ.loc[5, 'Model'], gb_accuracy, gb_roc_auc, gb_f1_score,
gb_avg_cv_score]
        summ.to_csv('model_metrics.csv', index=False)
        np.savetxt('metrics/fpr/gb.txt', gb_fpr)
        np.savetxt('metrics/tpr/gb.txt', gb_tpr)
        np.savetxt('metrics/precision/gb.txt', gb_precision)
        np.savetxt('metrics/recall/gb.txt', gb_recall)

```

Файл «4_Results.py»:

```
from imports import *
```

```
def compare_ml_models(summary_df):
```

```

sns.set_theme(style="white", rc={"axes.grid": False})

fig, axes = plt.subplots(3, 1, figsize=(6, 16))

pallette = «Spectral"

# Create a bar plot for Accuracy
sns.barplot(x='Model', y='Accuracy', data=summary_df, ax=axes[0], hue='Model', palette=pallette,
legend=False)
axes[0].set_title('Accuracy Comparison')
axes[0].set_xlabel('')
axes[0].set_ylabel('Accuracy')
axes[0].tick_params(axis='x', rotation=45, labelsz=10)

# Create a bar plot for ROC AUC
sns.barplot(x='Model', y='ROC AUC', data=summary_df, ax=axes[1], hue='Model', palette=pallette,
legend=False)
axes[1].set_title('ROC AUC Comparison')
axes[1].set_xlabel('')
axes[1].set_ylabel('ROC AUC')
axes[1].tick_params(axis='x', rotation=45, labelsz=10)

# Create a bar plot for F1 Score
sns.barplot(x='Model', y='F1 Score', data=summary_df, ax=axes[2], hue='Model', palette=pallette,
legend=False)
axes[2].set_title('F1 Score Comparison')
axes[2].set_xlabel('')
axes[2].set_ylabel('F1 Score')
axes[2].tick_params(axis='x', rotation=45, labelsz=10)

plt.tight_layout()
#plt.savefig('model_comparison.png', bbox_inches='tight', dpi=300)
st.pyplot(plt)

def plot_roc_auc_curves(ax, fpr, tpr, roc_auc, methods, colors):
    for x in range(len(roc_auc)):
        ax.plot(fpr[x], tpr[x], color=colors[x], lw=2, label='{0} ROC curve (area =
{1:.2f})'.format(methods[x], roc_auc[x]))
    ax.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    ax.set_xlim([0.0, 1.0])
    ax.set_ylim([0.0, 1.05])
    ax.set_xlabel('False Positive Rate')
    ax.set_ylabel('True Positive Rate')
    ax.set_title('ROC Curve Comparison')
    ax.legend(loc="lower right»)
    ax.grid(True)

def plot_precision_recalls(ax, recall, precision, methods, colors):
    for x in range(len(recall)):
        ax.plot(recall[x], precision[x], color=colors[x], lw=2, label='{0} Precision-Recall
Curve'.format(methods[x]))

```

```

ax.set_xlim([0.0, 1.0])
ax.set_ylim([0.0, 1.05])
ax.set_xlabel('Recall')
ax.set_ylabel('Precision')
ax.set_title('Precision-Recall Curve Comparison')
ax.legend(loc="lower right»)
ax.grid(True)

colors = ["red", «orange", «yellow", «lime", «green", «blue"]

col1, col2 = st.columns([3, 7])
with col1:
    option = st.selectbox('Result',
                          ('Table', 'Bar Chart', 'Graphs'), 0)

match option:
    case 'Table':
        st.dataframe(pd.read_csv('model_metrics.csv'))
    case 'Bar Chart':
        compare_ml_models(pd.read_csv('model_metrics.csv'))
    case 'Graphs':
        fig, axs = plt.subplots(2, 1, figsize=(10, 12))

        plot_roc_auc_curves(axs[0], [np.loadtxt('metrics/fpr/alt.txt', dtype=float),
np.loadtxt('metrics/fpr/rf.txt', dtype=float),
np.loadtxt('metrics/fpr/svm.txt', dtype=float),
np.loadtxt('metrics/fpr/dt.txt', dtype=float),
np.loadtxt('metrics/fpr/xgb.txt', dtype=float),
np.loadtxt('metrics/fpr/gb.txt', dtype=float)],
[ np.loadtxt('metrics/tpr/alt.txt', dtype=float),
np.loadtxt('metrics/tpr/rf.txt', dtype=float),
np.loadtxt('metrics/tpr/svm.txt', dtype=float),
np.loadtxt('metrics/tpr/dt.txt', dtype=float),
np.loadtxt('metrics/tpr/xgb.txt', dtype=float),
np.loadtxt('metrics/tpr/gb.txt', dtype=float)],
[pd.read_csv('model_metrics.csv').loc[0, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[1, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[2, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[3, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[4, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[5, 'ROC AUC']],
["Altman Z-Score", «RandomForest", «SVM", «DecisionTree", «XGBoost",
«GradientBoosting"], colors)

        plot_precision_recalls(axs[1], [np.loadtxt('metrics/recall/alt.txt', dtype=float),
np.loadtxt('metrics/recall/rf.txt', dtype=float),
np.loadtxt('metrics/recall/svm.txt', dtype=float),
np.loadtxt('metrics/recall/dt.txt', dtype=float),
np.loadtxt('metrics/recall/xgb.txt', dtype=float),
np.loadtxt('metrics/recall/gb.txt', dtype=float)],
[ np.loadtxt('metrics/precision/alt.txt', dtype=float),
np.loadtxt('metrics/precision/rf.txt', dtype=float),

```

```

        np.loadtxt('metrics/precision/svm.txt', dtype=float),
np.loadtxt('metrics/precision/dt.txt', dtype=float),
        np.loadtxt('metrics/precision/xgb.txt', dtype=float),
np.loadtxt('metrics/precision/gb.txt', dtype=float)],
        ["Altman Z-Score", «RandomForest", «SVM", «DecisionTree",
«XGBoost", «GradientBoosting"], colors)

    st.pyplot(plt)

_ = '''table_button = st.button('Table')
graph_button = st.button('Bar Chart')
plot_button = st.button('Plots')

if table_button:
    st.dataframe(pd.read_csv('model_metrics.csv'))

if graph_button:
    compare_ml_models(pd.read_csv('model_metrics.csv'))

if plot_button:
    fig, axs = plt.subplots(2, 1, figsize=(10, 12))

    plot_roc_auc_curves(axs[0], [np.loadtxt('metrics/fpr/alt.txt', dtype=float),
np.loadtxt('metrics/fpr/rf.txt', dtype=float),
        np.loadtxt('metrics/fpr/svm.txt', dtype=float),
np.loadtxt('metrics/fpr/dt.txt', dtype=float),
        np.loadtxt('metrics/fpr/xgb.txt', dtype=float),
np.loadtxt('metrics/fpr/gb.txt', dtype=float)],
        [np.loadtxt('metrics/tpr/alt.txt', dtype=float),
np.loadtxt('metrics/tpr/rf.txt', dtype=float),
        np.loadtxt('metrics/tpr/svm.txt', dtype=float),
np.loadtxt('metrics/tpr/dt.txt', dtype=float),
        np.loadtxt('metrics/tpr/xgb.txt', dtype=float),
np.loadtxt('metrics/tpr/gb.txt', dtype=float)],
        [pd.read_csv('model_metrics.csv').loc[0, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[1, 'ROC AUC'],
        pd.read_csv('model_metrics.csv').loc[2, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[3, 'ROC AUC'],
        pd.read_csv('model_metrics.csv').loc[4, 'ROC AUC'],
pd.read_csv('model_metrics.csv').loc[5, 'ROC AUC']],
        ["Altman Z-Score", «RandomForest", «SVM", «DecisionTree", «XGBoost",
«GradientBoosting"], colors)

    plot_precision_recalls(axs[1], [np.loadtxt('metrics/recall/alt.txt', dtype=float),
np.loadtxt('metrics/recall/rf.txt', dtype=float),
        np.loadtxt('metrics/recall/svm.txt', dtype=float),
np.loadtxt('metrics/recall/dt.txt', dtype=float),
        np.loadtxt('metrics/recall/xgb.txt', dtype=float),
np.loadtxt('metrics/recall/gb.txt', dtype=float)],
        [np.loadtxt('metrics/precision/alt.txt', dtype=float),
np.loadtxt('metrics/precision/rf.txt', dtype=float),
        np.loadtxt('metrics/precision/svm.txt', dtype=float),
np.loadtxt('metrics/precision/dt.txt', dtype=float),
        np.loadtxt('metrics/precision/xgb.txt', dtype=float),
np.loadtxt('metrics/precision/gb.txt', dtype=float)],
        ["Altman Z-Score", «RandomForest", «SVM", «DecisionTree", «XGBoost",
«GradientBoosting"], colors)

```

```

np.loadtxt('metrics/precision/svm.txt', dtype=float),
np.loadtxt('metrics/precision/dt.txt', dtype=float),
np.loadtxt('metrics/precision/xgb.txt', dtype=float),
np.loadtxt('metrics/precision/gb.txt', dtype=float)],
["Altman Z-Score", «RandomForest», «SVM», «DecisionTree»,
«XGBoost», «GradientBoosting"], colors)

st.pyplot(plt)'''

```

Файл «5_Prediction.py»:

```

from imports import *

xgb = joblib.load(«models/xgb.joblib»)
input_data = {
    'year': [2016],
    'X1': [1006.4939],
    'X2': [845.0214],
    'X3': [146.6846],
    'X4': [173.0316],
    'X5': [363.7101],
    'X6': [144.2582],
    'X7': [502.3378],
    'X8': [19.7519],
    'X9': [2948.7194],
    'X10': [2596.9589],
    'X11': [129.8355],
    'X12': [322.5764],
    'X13': [1140.2834],
    'X14': [87.0071],
    'X15': [570.1054],
    'X16': [2046.2967],
    'X17': [2378.3330],
    'X18': [2539.3476]
}

X_new = pd.DataFrame(input_data)
st.dataframe(X_new)

col1, col2 = st.columns([3, 7])

with col1:
    model_option = st.selectbox('Select Model',
                                ('Random Forest', 'XGBoost', 'Decision Tree',
                                'SVM', 'Gradient Boosting', 'Altman Z-Score'), 1)
    predict_button = st.button('Predict')

if predict_button:
    match model_option:

```

```
case 'XGBoost':  
    X_new = X_new[list(pd.read_csv('datasets/xgb_selected_features_test_dataset.csv'))[2:]]  
    with coll:  
        st.write(xgb.predict(X_new))  
    xgb_model = joblib.load(«models/xgb.joblib»)  
    explainer = shap.TreeExplainer(xgb_model)  
    shap_values = explainer(X_new)  
    st.pyplot(shap.plots.waterfall(shap_values[0]))
```