МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНОЇ ТА ПІСЛЯДИПЛОМНОЇ ОСВІТИ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **«Розробка інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання»**

Виконав: студент групи Іт-61

Спеціальності 126 «Інформаційні системи та технології»
(шифр і назва)
Віняр Володимир Васильович
(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.
(Прізвище та ініціали)

Рецензент: к.т.н., доцент Бабич М.І.
(Прізвище та ініціали)

ДУБЛЯНИ-2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНОЇ ТА
ПІСЛЯДИПЛОМНОЇ ОСВІТИ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри_____

д.т.н., проф. А.М. Тригуба
«____» _____2022 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Віняру Володимиру Васильовичу

1. Тема роботи: «Розробка інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання»

Керівник роботи Тригуба Анатолій Миколайович, професор
затверджені наказом по університету від 30.06.2022 року № 137/к-с.
2. Строк подання студентом роботи 10.12.2022 р.
3. Вихідні дані до роботи: *база даних щодо платоспроможності клієнтів банків; алгоритми машинного навчання; методика дослідження моделей машинного навчання.*
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Вступ.
1. Аналіз стану визначення платоспроможності клієнтів та завдання кваліфікаційної роботи.
2. Особливості вирішення задач класифікації та вибір методів машинного навчання для оцінювання платоспроможності сільськогосподарських підприємств.
3. Результати розробки інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств.
4. Охорона праці та безпека у надзвичайних ситуаціях.
5. Визначення ефективності запропонованої інтелектуальної інформаційної системи.
Висновки та пропозиції.
Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): *аналіз підходів до оцінювання платоспроможності позичальників; аналіз існуючих методів класифікації машинного навчання із бібліотекою Scikit-learn; особливості вирішення задач класифікації та оцінення якості класифікації; результати підготовки та аналізу даних для оцінювання платоспроможності сільськогосподарських підприємств; результати підбору параметрів та аналізу якості моделі градієнтного бустингу XGBoost; результати обґрунтування архітектури інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств; економічна ефективність.*

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	*Тригуба А.М., зав. кафедри інформаційних технологій*		
4	*Городецький І.М., доцент кафедри управління проектами та безпеки виробництва*		

7. Дата видачі завдання 30 червня 2022 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	*Написання першого розділу*	*30.06-04.07.22*	
2	*Виконання другого розділу та аркушів ілюстраційного матеріалу до нього*	*05.07-14.08.22*	
3.	*Виконання третього розділу та аркушів ілюстраційного матеріалу до нього*	*15.08-24.09.22*	
4.	*Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»*	*25.09-10.10.22*	
5.	*Оцінення ефективності запропонованої системи*	*20.10-31.10.22*	
6.	*Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу*	*01-30.11.22*	
7.	*Завершення роботи вцілому*	*01-10.12.22*	

Студент _____ Віняр В.В.
(підпис)

Керівник роботи _____ Тригуба А.М.
(підпис)

УДК 621.311.1

Розробка інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання.

Віняр В.В. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2022.

Кваліфікаційна робота: 66 с. текст. част., 11 рис., 4 табл., 10 арк. ілюстраційного матеріалу, 46 джерел.

Виконано аналіз підходів до оцінювання платоспроможності позичальників. Проаналізовано інформаційні системи та технології для фінансових установ. Наведено методи класифікації машинного навчання із бібліотекою Scikit-learn. Сформульовано завдання кваліфікаційної роботи.

Наведено особливості вирішення задач класифікації. Оцінення якості класифікації. Проаналізовано класифікатор випадкового лісу (Random Forest Classifier), підсилення градієнта для класифікації (Gradient Boosting Classifier) та градієнтний бустинг для класифікації (XGB Classifier).

Виконано підготовку та аналіз даних для оцінювання платоспроможності сільськогосподарських підприємств. Створено конвейєр та навчання моделей. Виконано підбір параметрів та аналіз якості моделі градієнтного бустингу XGBoost. Запропонована архітектура інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств.

Розроблено заходи стосовно охорони праці та безпека у надзвичайних ситуаціях. Визначено ефективність від використання інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств.

ЗМІСТ

ВСТУП

Оскільки все більше продуктів стають розумними та підключеними до різноманітного програмного забезпечення, проектування інтелектуальних інформаційних систем стає основною складовою для створення цінності окремих компаній різних прикладних сфер. Інтеграція фізичного та цифрового просторів починається з датчиків і сенсорних даних, які автоматизують і кількісно визначають відстеження як для розповсюдження продукту, так і для поведінки клієнтів у фізичному світі.

Водночас усе більше і більше компаній зараз формують та зберігають набори великих даних, які можна використовувати для аналітики та вирішення низки прикладних задач. При цьому машинне навчання має на меті зробити дані назагал доступними, щоб вирішувати окремі задачі, в тому числі це стосується і платоспроможності сільськогосподарських підприємств. Щоб спростити аналіз даних і, головне, знайти зв’язки між наборами даних слід автоматизувати машинне навчання. Це забезпечить всім співробітникам доступ до передових професійних аналітичних можливостей.

Розв’язання науково-прикладної задачі розробки інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств потребує проведення досліджень стосовно вибору ефективних алгоритмів машинного навчання. Для цього слід виконувати аналіз відомих алгоритмів, а також методів уже виконаних досліджень із використанням технологій машинного навчання, які скеровані на вирішення задач класифікації об’єктів за їх характеристиками.

Виконання досліджень потребує обґрунтування набору інструментів, які можуть вчитися з наявними даними. Розуміти закономірності та зв’язки між атрибутами даних та встановлення закономірностей лежить в основі якісного навчання та вишення задач класифікації. Отримані моделі Data Science розширюють можливості роботи з нелінійними зв’язками в системах.

Стосовно прогнозування платоспроможності позичальників, то існуючі підходи та методи зосереджені на надійності. Тоді як алгоритми машинного навчання набирають популярності завдяки своїм точним результатам. Головним недоліком передових інструментів науки про дані є їхня обмежена здатність інтерпретувати отримані результати. Використовувати ці методи не завжди можливо. Це тому, що вони часто є у центральних банках, які спеціалізуються в інших сферах.

На підставі цього вважаємо, що виконана кваліфікаційна робота «Розробка інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання» є достатньо актуальна, а також наявна її практична цінність.

Об’єктом дослідження є алгоритми машинного навчання, які призначені для вирішення задач класифікації та оцінювання платоспроможності сільськогосподарських підприємств.

Предмет дослідження є нелінійні зв’язки між платоспроможністю представників селянських господарств та їх характеристиками.

Метою роботи є підвищення ефективності оцінювання платоспроможності сільськогосподарських підприємств завдяки розробці інтелектуальної інформаційної системи, що базується на раціональній моделі класифікації даних.

РОЗДІЛ 1.
АНАЛІЗ СТАНУ ВИЗНАЧЕННЯ ПЛАТОСПРОМОЖНОСТІ КЛІЄНТІВ ТА ЗАВДАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1.1. Аналіз підходів до оцінювання платоспроможності позичальників

На у багатьох організаціях, в тому числі і сільськогосподарських підприємствах, виникає низка задач які стосуються кредитних відносин. При цьому використовується велика кількість підходів і методів оцінки кредитоспроможності позичальників. Заслуговує на увагу запропонована професором І.В. Вишняковим класифікація підходів до оцінки платоспроможності позичальників (рис. 1.1) [11]

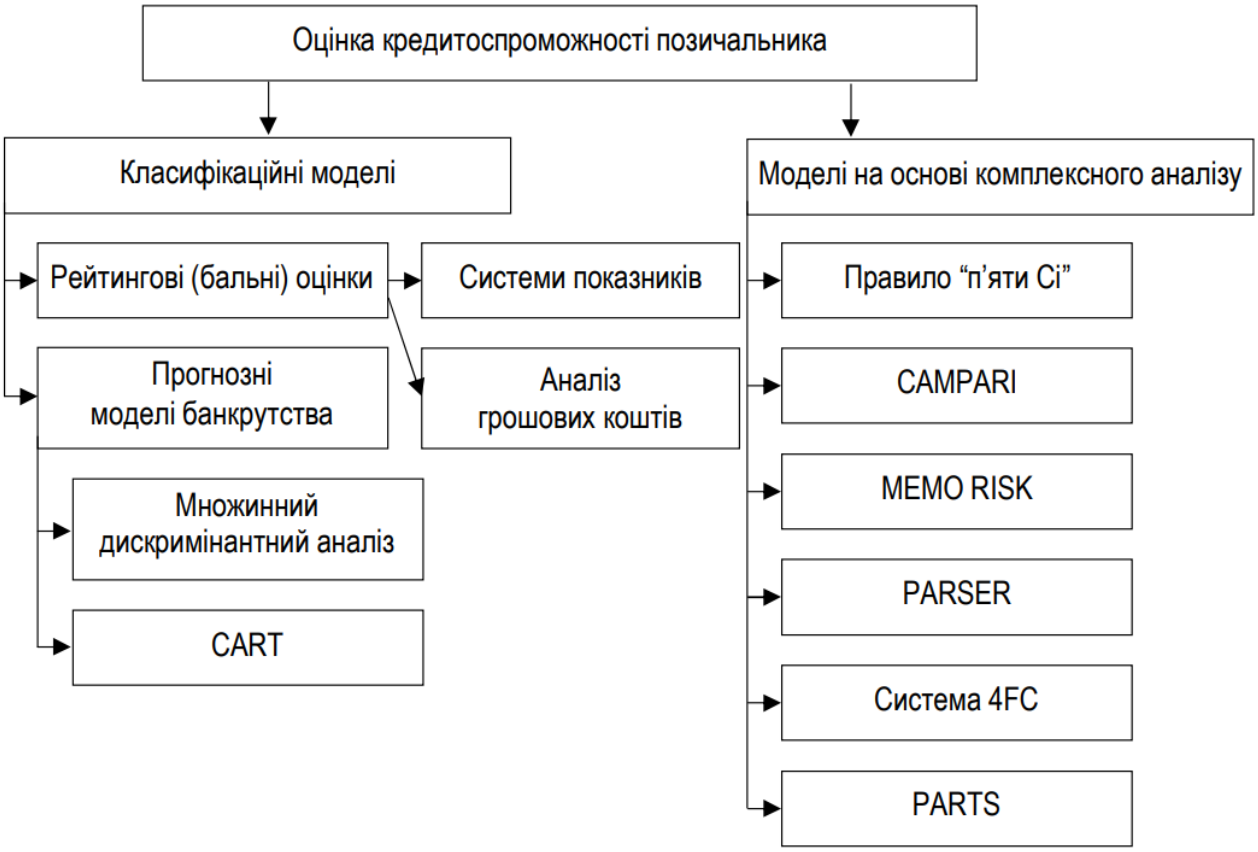

Рис. 1.1. Класифікація підходів до оцінки платоспроможності позичальників [2]

Серед наявних методів і моделей оцінки платоспроможності позичальників виділяють:

• категоріальні (статистичні) методи оцінки, рейтингові (бальні) системи оцінювання та моделі прогнозування;

• комплексного аналізу (засновані на «емпіричних» підходах, тобто на експертних оцінках аналізу економічної доцільності кредитування): «Правило п’яти Сі», система 4FC, PARTS, MEMO RISK, CAMPARI, PARSER тощо [40].

Застосування методів класифікації для оцінки платоспроможності позичальника, розробка стандартизованих підходів до орієнтації на позичальників, постановка мети пошуку кількісних орієнтирів поділу потенційних клієнтів на надійних і ненадійних за даними, які вони надають, тобто за ризиком банкрутства.

Бальне оцінення ставки дає можливість прогнозувати терміни майбутніх платежів, ліквідність та реальність платоспроможності позичальників, оцінення їх загального фінансового стану сільськогосподарського підприємства та його стійкість до ризику. Вирішення таких задач забезпечує повернення частини прибутку, забезпечення фіксованих платежів тощо. Перевагами бальних рейтингових моделей є простота, а також можливості виконання розрахунків на основі показників. На підставі цієї методики є змога оцінити ініціативи сільськогосподарських підприємств щодо кредитної спроможності як позичальника.

Заслуговують на увагу моделі комплексного аналізу, які використовуються у розвинутих країнах. У них використовують досить складну систему показників оцінки платоспроможності позичальників. Вони змінюються в залежності від категорії позичальника (компанія, фізична особа, вид діяльності тощо), а також обсягів надходжень готівки на рахунки підприємства. Кількісні та якісні характеристики сільськогосподарських підприємств, позичальників, визначаються моделями, що передбачають комплексний аналіз («Правило п’яти Сі», система 4FC, PARTS, MEMO RISK, CAMPARI, PARSER тощо) існуючої негативної кредитної історії [2]. Зазначені

методики оцінки оцінки платоспроможності позичальників набули популярності через вдале поєднання аналізу характеристик окремих позичальників.

1.2. Інформаційні системи та технології для фінансових установ

Провідні країни світу мають значний досвід використання та проектування інформаційних систем та технологій для фінансових установ. Зокрема, це стосується управління захисту від витоку даних, підвищення рівня прозорості та оновлення механізмів надання фінансових послуг, оцінення кредитоспроможності тощо. Кращий досвід використання інформаційних систем та технологій розвинутих країн є досить цінним для вітчизняних установ, а його практичне впровадження сприяє підвищенню рівня конкуренції на ринках фінансових послуг та рівня довіри населення до банківських установ (рис. 1.2).

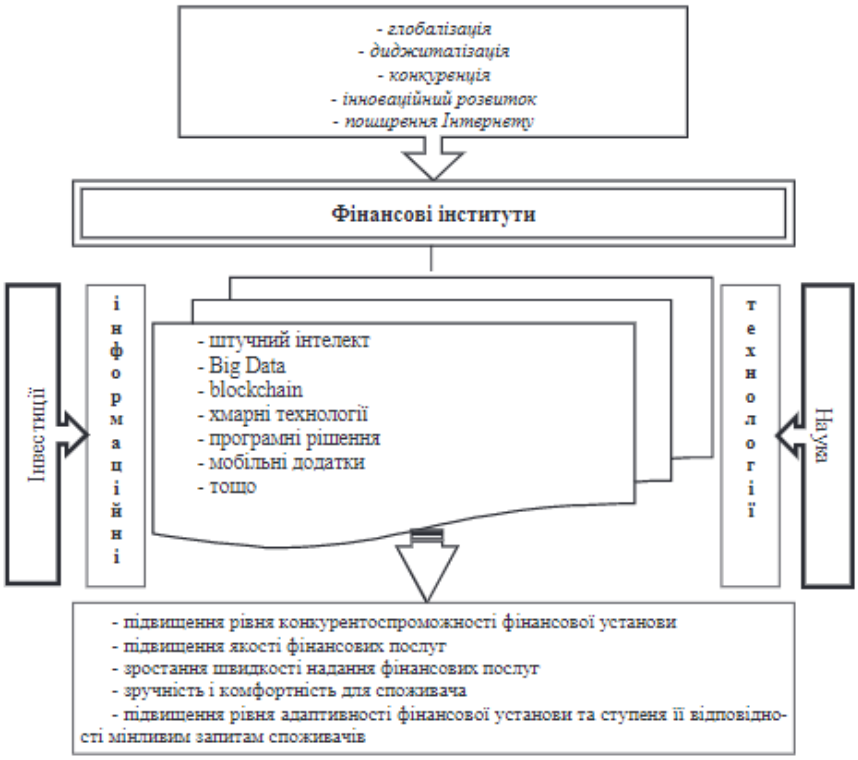

Рис. 1.2. Складові впливу на розробку інформаційних систем та технологій для фінансових установ

Великі дані відіграють важливу роль серед інформаційних систем та технологій, які активно використовуються в діяльності у фінансових установах. Вони можуть прискорити процес збору та аналізу інформації, а також покращити контроль над інформаційною безпекою та системами запобігання шахрайству. На основі технології великих даних британський комерційний банк «NatWest» створив цифрові сервіси «Mettle» для малого та середнього бізнесу [35]. Використовуючи штучний інтелект і аналіз великих даних, він розробив додаток «Mito» для особистого фінансового обліку [32]. Запустив цифровий помічник «» в онлайн-банкінгу, який надає клієнтам інформацію з понад 200 запитань [35].

Gnipo Santander Banking Group (Іспанія) – одна з перших європейських країн, яка впровадила інноваційну інформаційну технологію блокчейн. Використання такої технології в роботі фінансових установ може значно прискорити процес платежів, працювати без посередників і забезпечити прозорість транзакцій. Технологія блокчейн є основою платіжної системи «One Pay FX», розробленої «Santander Group», яка спрямована на оптимізацію платежів між Європою та Південною Америкою [32].

Використання роботів у діяльності фінансових установ поступово поширюється. По суті, це спеціальна інженерна розробка, пристрій з певним набором команд, прописаних фахівцями, який сьогодні зазвичай не є революційним, але в останні роки заслужили нові форми впливу та сфери застосування. Впровадження роботів дозволить банкам заощадити витрати на оплату праці касирів та інших спеціалістів, які виконують рутинні, монотонні та монотонні технічні завдання.

Викладене свідчить про актуальність використання великих даних, а на їх основі технологій машинного навчання для дослідження та розробки інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання.

1.3. Методи класифікації машинного навчання із бібліотекою Scikit-learn

Існує багато методів класифікації, які використовують різний математичний апарат і різні підходи до реалізації [40]. Однак ефективність цих методів залежить від предметної галузі та вирішуваних задач. За останні 10 років комерційні компанії зайняті підвищенням якості машинного навчання, але поки не знайдено універсальної технології, яка дійсно розв’язує усі задачі класифікації із високою точністю. Тому необхідно проаналізувати використання різних алгоритмів класифікації за допомогою бібліотеки Scikit-Learn, яка використовується для мови Python.

За останні роки проведено велику кількість демонстраційних досліджень та впроваджено багато методів машинного навчання в різних областях [3]. В основному використовується машинне навчання для вирішення задач, які є надто складні для вирішення і потребують адаптації до середовища. Тобто це такий клас задач, які неможливо вирішити за певним чітким алгоритмом, необхідно враховувати вже отримані результати. Про аналіз літературних джерел свідчить про те, що використання цих методів є обмежене малим обсягом інформації про стан досліджуваного об’єкту.

Існує багато бібліотек для машинного навчання, написаних на Python. Розглянемо одну з найпопулярніших Scikit-Learn.

Scikit-Learn – це бібліотека Python, створена Девідом Курнапеу в 2007 році. Ця бібліотека містить велику кількість алгоритмів для задач класифікації. Scikit-Learn базується на бібліотеці SciPy і її потрібно встановити перед початком роботи. Впроваджуючи чітку, добре задокументовану та надійну бібліотеку Scikit-Learn, можна допомагати спростити процес побудови класифікаторів і більш чітко підкреслити концепції машинного навчання. Вона містить багато методів, які охоплюють все, що може знадобитися під час аналізу даних: алгоритми класифікації та регресії тощо. Окрім того, її можна

використовувати для кластеризації, перевірки та вибору моделі. Також її можна використовувати для зменшення розмірності даних і ознак (рис. 1.3).

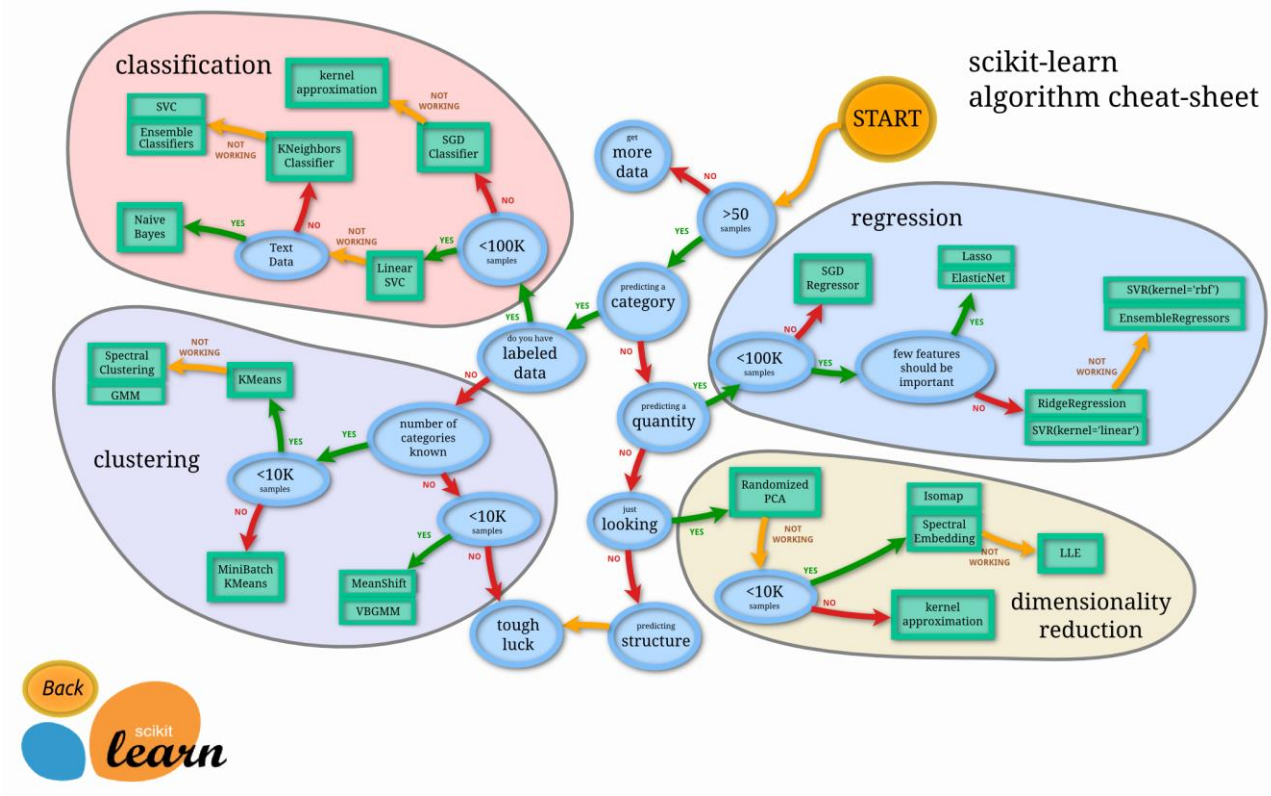

Рис. 1.3. Наявні алгоритми у бібліотеці Scikit-learn

Система машинного навчання має входи та виходи. Дані, які подають на вхід, називається ознакою. Коли ознака передається як вхідні дані в систему машинного навчання, система намагається знайти збіги та шаблони серед ознак [6]. Цей результат зазвичай називають міткою, оскільки результат має певну позначку, надану системою, тобто прогноз, до якої категорії належить вихід після класифікації.

Scikit-Learn надає доступ до різноманітних алгоритмів класифікації. Основні з них:

- k найближчих сусідів (K Nearest Neighbours);

- Метод опорного вектора;

- Класифікатор дерева рішень (Decision Tree класифікатор) / random forest (випадковий ліс);

– Наївний метод Байєса;

- лінійний дискримінантний аналіз (лінійний дискримінантний аналіз);

– логічна регресія (логістична регресія).

Завдання класифікації – це завдання, яке вимагає визначення типу об’єкта з двох або більше існуючих класів. Залежно від завдання класифікації доступні різні типи класифікаторів. Наприклад, логістична регресія найкраща, якщо ваша класифікація включає двійкову логіку.

Процес машинного навчання включає етапи підготовки даних, створення навчального набору, створення класифікатора, навчання класифікатора, прогнозування, оцінку ефективності класифікатора та налаштування параметрів.

Спочатку нам потрібно підготувати набір даних для класифікатора. Перетворення даних у правильний формат для класифікації та обробки аномалій у даних. Необхідно обробляти будь-які відсутні значення даних або інші викиди. Невиконання цього може негативно вплинути на продуктивність класифікатора. Цей етап називається попередньою обробкою даних.

Наступним кроком є розділення даних на навчальний набір і тестовий набір. Для цього у ScikitLearn є функція traintestsplit. Як було сказано раніше, класифікатор створює навчальний набір даних і виконує навчання на них. Після цих кроків модель може робити прогнози. Порівнюючи продуктивність класифікатора з реальними відомими даними, ми можемо зробити висновки про точність класифікатора.

Класифікатор навряд чи задовольнить усі вимоги з першого запуску, тому параметри класифікатора слід «налаштувати», поки не буде досягнута бажана точність [17].

Оцінка: Існує кілька варіантів оцінки класифікатора. Точність класифікації є найбільш часто використовуваним параметром, оскільки його найлегше виміряти. Значення точності – це кількість правильних передбачень, поділена на кількість усіх передбачень, або просто відношення правильних передбачень до всіх. Хоча цей показник відразу може виявити очевидне з точки зору продуктивності класифікатора. Його найкраще використовувати, коли

кожен клас має принаймні приблизно однакову кількість прикладів. Оскільки це трапляється рідко, ми рекомендуємо використовувати інші класифікатори.

Значення логарифмічних втрат (logloss) вказує на достовірність прогнозів класифікатора. Logloss повертає ймовірності того, що об’єкт належить до певного класу, і підсумовує їх, щоб отримати загальне уявлення про «правильність» класифікатора. Цей показник коливається від 0 до 1 – «не впевнений» і «повністю впевнений». Втрати Logloss різко падають, коли класифікатор надто «впевнений» у неправильній відповіді.

Площа ROC-кривої (AUC) використовується як показник лише для двійкової класифікації. Площа під ROC-кривою – це здатність класифікатора розрізняти придатні та непридатні об'єкти для будь-якого заданого класу. Значення 1.0: уся область під кривою представляє ідеальний класифікатор. Отже, 0,5 означає, що точність класифікатора відповідає випадковості. Крива розраховується з урахуванням точності та специфіки моделі.

На підставі виконаного аналізу зауважено, що бібліотека ScikitLearn має потрібні методи машинного навчання, які забезпечують оцінювання платоспроможності сільськогосподарських підприємств на підставі наявних даних. Це лежить в основі розробки інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання.

1.4. Завдання кваліфікаційної роботи

На виконаного аналізу встановлено, що використання наявного інструментарію оцінювання платоспроможності сільськогосподарських підприємств знижує точність отриманих рішень через їх недоліки. Це зумовлює потребу розроблення інтелектуальної інформаційної системи оцінювання

платоспроможності сільськогосподарських підприємств. Для цього у кваліфікаційній роботі існує потреба у розв’язанні таких завдань:

- провести аналіз існуючих підходів до оцінювання платоспроможності сільськогосподарських підприємств та сформулювати завдання роботи;
- описати особливості вирішення задач класифікації та здійснити вибір методів машинного навчання для оцінювання платоспроможності сільськогосподарських підприємств;
- виконати дослідження методів машинного навчання та обґрунтувати раціональний для оцінювання платоспроможності сільськогосподарських підприємств;
- запропонувати архітектуру інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств;
- розробити заходи щодо охорони праці та безпеки у надзвичайних ситуаціях;
- виконати розрахунок економічної ефективності від використання інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств.

РОЗДІЛ 2.
ОСОБЛИВОСТІ ВИРІШЕННЯ ЗАДАЧ КЛАСИФІКАЦІЇ ТА ВИБІР МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ОЦІНЮВАННЯ ПЛАТОСПРОМОЖНОСТІ СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВ

2.1. Особливості вирішення задач класифікації

Оцінювання платоспроможності сільськогосподарських підприємств належить до задач класифікації. Задачі класифікації у машинному навчанні (ML) – це ті задачі, які вимагають, щоб даний набір даних класифікувався за двома або більше категоріями. Наприклад, чи особисте селянське господарство є платоспроможним (відповідь «Так» або «Ні») можна назвати задачею класифікації.

Задачі класифікації можуть бути наступних різних типів:

Бінарна класифікація – класифікує дані за двома класами, наприклад «Так»/«Ні», «добре/погано», «високий/низький», «платоспроможний чи неплатоспроможний» тощо. На рис. 2.1. показано модель класифікації, що представляє лінії, що розділяють два різні класи. Залежно від типу задач і лінійних/нелінійних даних межа, що розділяє класи, матиме різний характер.

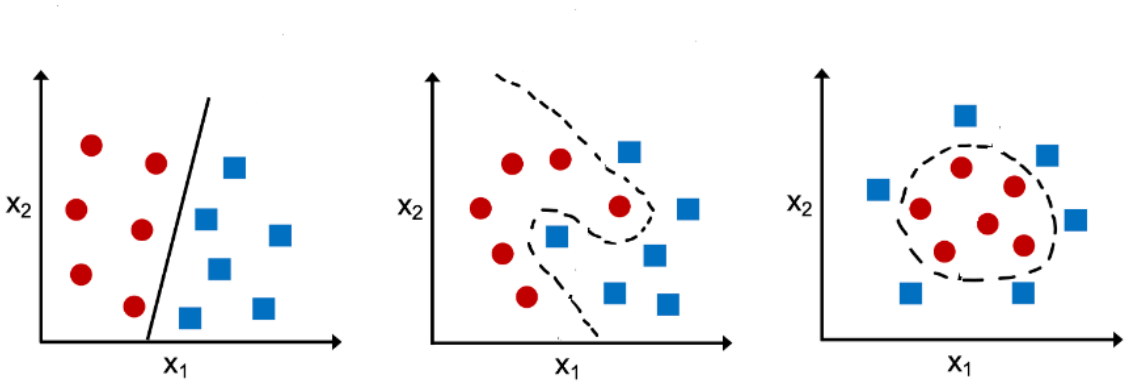

Рис. 2.1. Види поділу даних на класи

Наявна мультиноміальна класифікація, яка класифікує дані на три або більше класів. При цьому дані класифікуються як м’які призначення, наприклад, ймовірність того, що кожна категорія або клас застосовується до даних. Таким чином, якщо є три класи 10 для розпізнавання цифр від 0 до 9, результатом класифікації буде ймовірність того, що кожен клас застосовується до певної цифри. Таким чином, 0 може бути 0,01, а 1 може бути 0,3, 3 як 0,05, 4 як 0,02, 5 як 0,0,8, 6 як 0,1, 7 як 0,5 і т. д. Таким чином, цифру можна класифікувати як належну до 7 класу на основі значення ймовірності.

Задачі класифікації – це контрольовані навчальні задачі, у яких навчальний набір даних складається з даних, пов’язаних із незалежними змінними та відповідними змінними (мітка). Моделі класифікації навчаються за допомогою деяких із наведених нижче алгоритмів:

- Логістична регресія;
- Дерева рішень;
- Випадковий ліс;
- XGBoost;
- Light GBM;
- Класифікатори голосування;
- Штучні нейронні мережі;

Під час навчання моделі для задач класифікації на основі контрольованого навчання існують різні стратегії, які застосовуються для призначення міток відповіді або вихідних змінних.

Призначення цілого числа виводу: якщо є два класи, наприклад платоспроможний, неплатоспроможний, можна виконати {1, 2} відповідно для {платоспроможний, неплатоспроможний}.

Існує поняття призначення вектора одноразового кодування. Можна призначити вектор одноразового кодування для позначення вихідних даних. Одночасне кодування представляє вектор, який має стільки компонентів, скільки класів або категорій. Компонент, що відповідає класу або категорії конкретного екземпляра, встановлюється на 1, а всі інші компоненти – на 0.

Таким чином, одноразовий вектор кодування для класу платоспроможний буде (1, 0). Для класу неплатоспроможний це буде (0, 1). Таким чином, вихідні мітки можуть бути представлені як {(1, 0), (0, 1)}.

Отже, у Deep Learning задачі класифікації вирішуються шляхом навчання моделей класифікації. Моделі класифікації навчаються шляхом надання об’єктів та їхніх міток. Моделі навчають і ідентифікують схожі характеристики об’єктів у класі. Після навчання модель тестується на окремих даних, які вона навчила. Для перевірки надається лише об’єкт для класифікації без його мітки. Модель класифікації передбачає мітку об'єкта. Точність моделі визначається на основі правильно передбачених міток.

2.2. Оцінення якості класифікації

Для оцінювання якості бінарної класифікації використовують ROC-криві, які являють собою графіки, що відображають співвідношення між часткою об'єктів від сумарної кількості носіїв певної ознаки. При цьому вони повинні бути правильно класифіковані до сумарної кількості об'єктів, які немають ознаки, тобто помилково класифіковані, як такі що мають ознаку.

Крива ROC (крива робочих характеристик приймача) – це графік, що показує ефективність моделі класифікації за всіх порогів класифікації. Ця крива відображає два параметри:

- справжня позитивна оцінка;
- хибно позитивний рівень.

Справжній позитивний показник (TPR) є синонімом запам’ятовування, і тому визначається таким чином:

$$TPR = \frac{TP}{TP + FN}, \qquad (2.1)$$

Частота помилкових позитивних результатів (FPR) визначається таким чином:

$$FPR = \frac{FP}{FP + TN}, \tag{2.2}$$

Крива ROC відображає TPR проти FPR за різних порогів класифікації. Зниження порогу класифікації класифікує більше елементів як позитивні, таким чином збільшуючи кількість помилкових і справжніх позитивних результатів. На рис. 2.2 показано типову криву ROC.

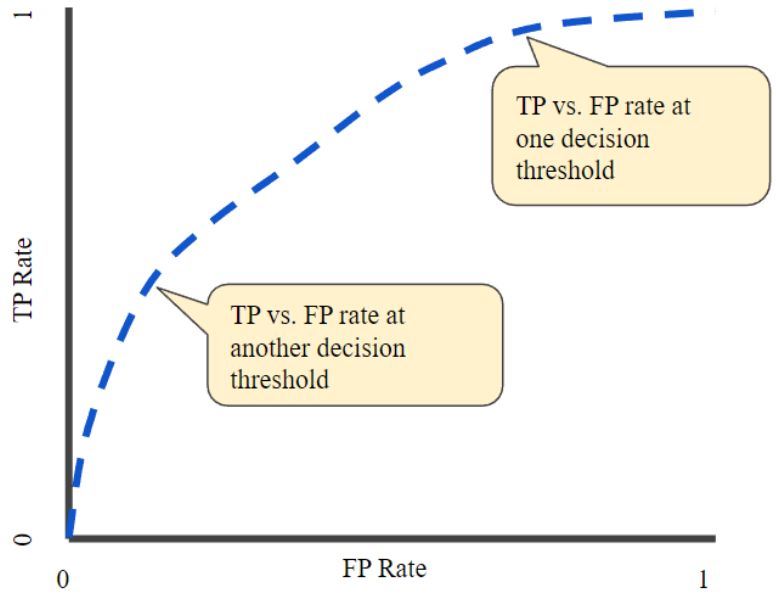

Рис. 2.2. Залежність частоти TP від FP за різних порогів класифікації

Щоб обчислити точки на кривій ROC, ми могли б багато разів оцінити модель логістичної регресії з різними пороговими значеннями класифікації, але це було б неефективно. При цьому існує ефективний алгоритм на основі сортування, який може надати нам цю інформацію, називається AUC – площа під кривою ROC.

AUC означає «Площа під кривою ROC». Тобто AUC вимірює всю двовимірну площу під усією кривою ROC (інтегральне числення) від (0,0) до (1,1).

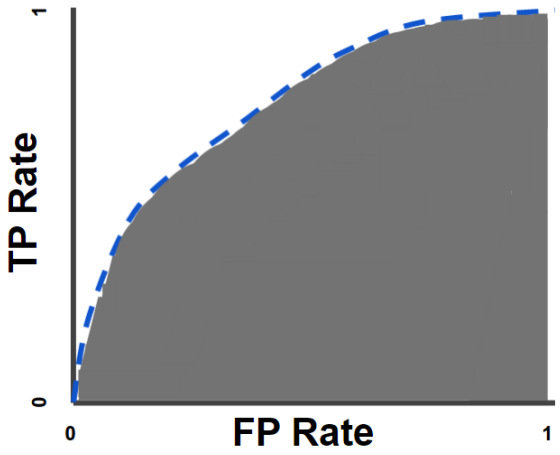

Рис. 2.3. AUC «Площа під кривою ROC»

AUC забезпечує сукупний показник продуктивності за всіма можливими пороговими значеннями класифікації. Одним із способів інтерпретації AUC є ймовірність того, що модель оцінює випадковий позитивний приклад вище, ніж випадковий негативний приклад. Наприклад, наведені нижче приклади, які розташовані зліва направо в порядку зростання прогнозів логістичної регресії (рис. 2.4).

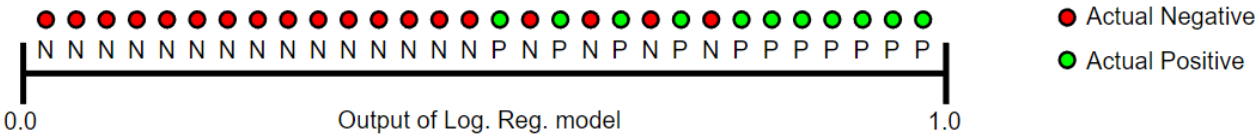

Рис. 2.4. Прогнози, ранжовані в порядку зростання балів логістичної регресії

AUC представляє ймовірність того, що випадковий позитивний (зелений) зразок розташований праворуч від випадкового негативного (червоний) зразок. Значення AUC коливається від 0 до 1. Модель, прогнози якої на 100% помилкові, має AUC 0,0; а та яка забезпечує прогнози на 100% правильні, має AUC 1,0.

2.3. Класифікатор випадкового лісу (Random Forest Classifier)

Наука про дані надає досить багато алгоритмів класифікації, таких як опорна векторна машина, наївний класифікатор Байєса, логістична регресія, дерева рішень тощо. Але на вершині ієрархії класифікаторів знаходиться у класифікаторі випадкового лісу (існує також регресор випадкового лісу). Щоб зрозуміти роботу класифікатора випадкового лісу, нам потрібно спочатку зрозуміти концепцію дерев рішень.

Випадкові ліси або ліси випадкових рішень – це метод ансамблевого навчання для класифікації, регресії та інших завдань, який працює шляхом побудови багатьох дерев рішень під час навчання. Для класифікаційних завдань результатом випадкового лісу є клас, вибраний більшістю дерев. Для завдань регресії повертається середнє або середнє прогнозування окремих дерев (рис. 2.1).

Випадкові ліси часто використовуються як моделі «чорної скриньки» в бізнесі, оскільки вони генерують обґрунтовані прогнози для широкого діапазону даних, вимагаючи невеликої конфігурації.

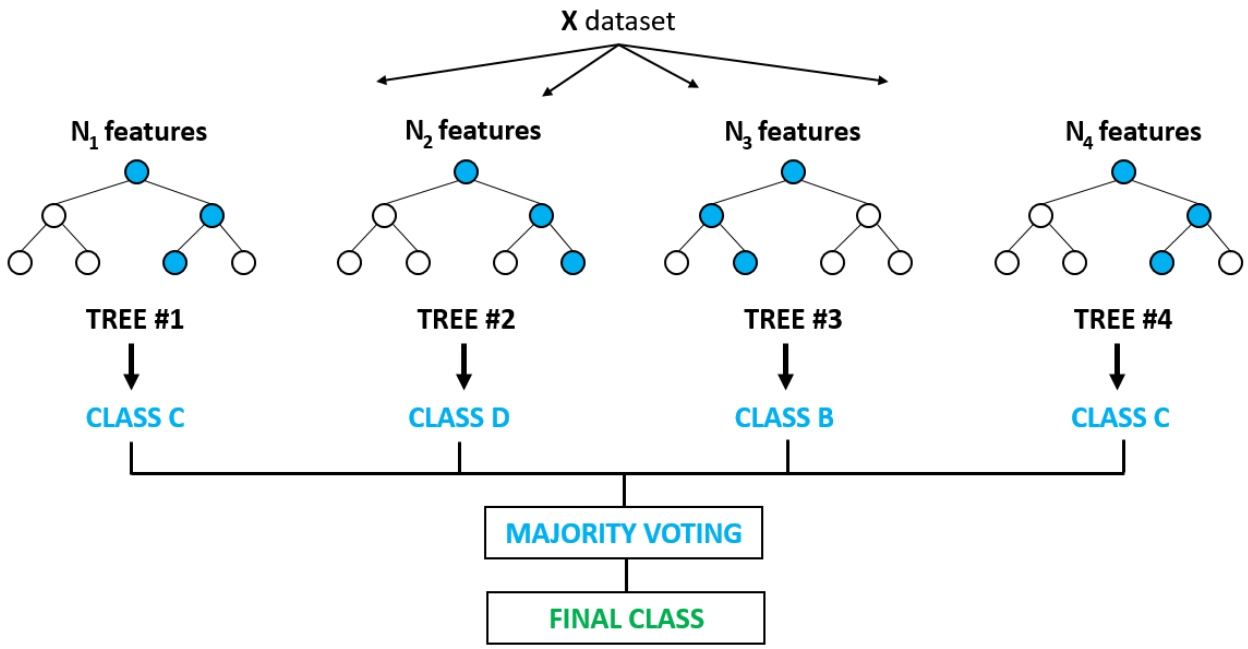

Рис. 2.1. Схема класифікатора випадкового лісу (Random Forest Classifier)

Алгоритм класифікатора випадкового лісу (Random Forest Classifier) виглядає наступним чином:

1. Виконується побудова трьох прикладів початкового завантаження з оригіналу даних.

2. Для кожного із зразків початкового завантаження виростити необрізане дерево класифікації з наступною модифікацією: на кожному вузлі виконується вибір найкращого розподілу серед усіх предикторів, формується випадкова вибірка m_{try} предикторів і вибирається із них найкращий пул змінних. (укладання можна розглядати як окремий випадок випадкових лісів, отриманих, коли $m_{try} = p$, кількість предикторів).

3. Прогнозуються нові дані шляхом агрегування прогнозів n-дерев (тобто більшість голосів за класифікацію, середнє для регресії).

Можна отримати оцінку рівня помилок, на основі даних навчання наступним чином:

1. На кожній ітерації початкового завантаження прогнозуються дані не з прикладу початкового завантаження, а за допомогою дерева вирощеного за зразком початкового завантаження.

2. Агрегування передбачень OOB. (У середньому, кожна точка даних буде поза пакетом приблизно у 36% випадків, тож слід об’єднати їх прогнозовані). Обчислюється частота помилок і визначається OOB щодо оцінки частоти помилок. Відомо, що оцінка OOB частоти помилок досить точна, враховуючи, що дерев мають достатню глибину вирощування [27].

Ми будемо використовувати модуль sklearn для навчання нашої регресійної моделі випадкового лісу, зокрема функції RandomForestRegressor. Документація RandomForestRegressor показує багато різних параметрів, які ми можемо вибрати для нашої моделі. Деякі з важливих параметрів виділено нижче:

- *n_estimators* – кількість дерев рішень, які ви будете запускати в моделі;

- *criterion* – ця змінна дозволяє вибрати критерій (функцію втрат), який використовується для визначення результатів моделі. Ми можемо вибрати з таких функцій втрат, як середня квадратична помилка (MSE) і середня абсолютна помилка (MAE). Значенням за замовчуванням є MSE;
- *max_depth* – встановлює максимально можливу глибину кожного дерева;
- *max_features* – максимальна кількість функцій, які модель враховуватиме при визначенні розбиття;
- *bootstrap* – значенням за замовчуванням є True, тобто модель дотримується принципів початкового завантаження (визначених раніше);
- *max_samples* – цей параметр припускає, що для початкового завантаження встановлено значення True, якщо ні, цей параметр не застосовується. У випадку True це значення встановлює найбільший розмір кожної вибірки для кожного дерева.

2.4. Підсилення градієнта для класифікації (Gradient Boosting Classifier)

Протягом багатьох років підсилення градієнта знайшло застосування в різних галузях техніки. На перший погляд алгоритм може здатися складним, але в більшості випадків використовує лише одну попередньо визначену конфігурацію для класифікації та одну для регресії, які можна змінити відповідно до вимог. Зосередимося на підсиленні градієнта для задач класифікації. Цей алгоритм працює за налаштуваннями, інтуїтивно та математично. Gradient Boosting складається з трьох основних компонентів:

Функція втрат. Роль функції втрат полягає в тому, щоб оцінити, наскільки добре модель робить прогнози на основі заданих даних. Це може змінюватися залежно від наявної проблеми. Наприклад, якщо ми намагаємося передбачити платоспроможність селянських господарств залежно від деяких

вхідних змінних, тоді функція втрат забезпечить знаходження різниці між прогнозованою платоспроможністю та спостережуваною платоспроможністю. Для цього потрібна функція втрат, яка допоможе зрозуміти, наскільки запропонована модель точна для класифікації платоспроможності селянських господарств.

Слабкі учні– погане навчання, яке класифікує наші дані, але робить це погано порівняно із випадковим вгадуванням. Іншими словами, спостерігається високий рівень помилок. При цьому отримуються дерева рішень (також їх називають пнями рішень, оскільки вони менш складні, ніж типові дерева рішень).

Адитивна модель – це ітеративний і послідовний підхід додавання дерев (слабких учнів) крок за кроком. Після кожної ітерації нам потрібно бути ближче до нашої остаточної моделі. Іншими словами, кожна ітерація повинна зменшувати значення нашої функції втрат.

Найпростіший спосіб використовувати логарифм (коефіцієнти) для класифікації – це перетворити його на ймовірність. Для цього ми скористаємося такою формулою:

$$P\left(surviving\right)=\frac{e^{log(odds)}}{1+e^{log(odds)}}. \qquad (2.3)$$

Якщо ймовірність правдивого прогнозу перевищує 0,5, виконується класифікація всіх даних у навчальному наборі. (0,5 – це загальний поріг, який використовується для класифікаційних рішень, прийнятих на основі ймовірності. Поріг можна легко змінити і прийняти інше його значення). Після цього обчислюється псевдозалишок, тобто різницю між спостережуваним і прогнозованим значенням. Отримування залишків представлено на рис. 2.2.

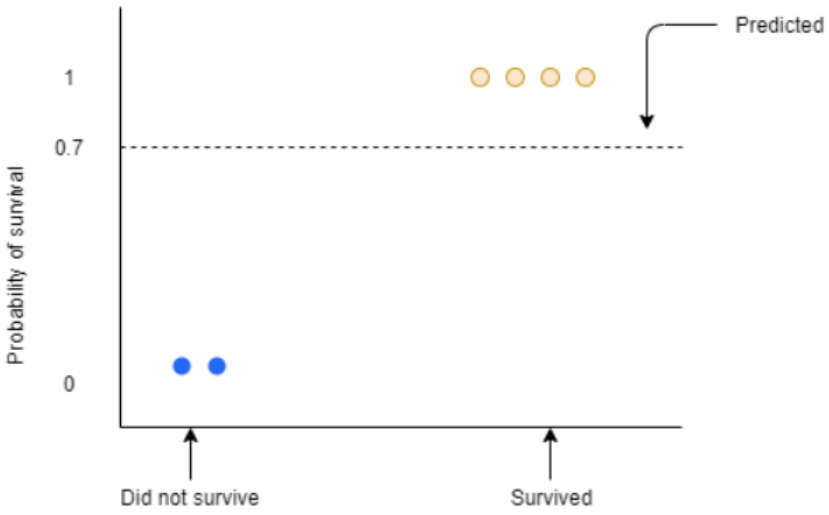

Рис. 2.2. Схема отримування залишків

Сині та жовті крапки є спостережуваними значеннями. Сині крапки позначають позичальників, які не спроможні повернути кредит з ймовірністю 0, а жовті крапки позначають позичальників, які спроможні повернути кредит і є платоспроможними з ймовірністю 1. Пунктирна лінія тут представляє прогнозовану ймовірність, яка становить 0,7. Залишок можна визначити за формулою:

$$Residual = Observed - Predicted\,. \qquad (2.4)$$

При цьому 1 означає, що позичальник є платоспроможним, а 0 означає що позичальник є неплатоспроможним.

Використовується цей залишок, щоб отримати наступне дерево. При цьому розглядається залишкова, а не фактична платоспроможність, але це лежить в основі майбутнього прогнозування.

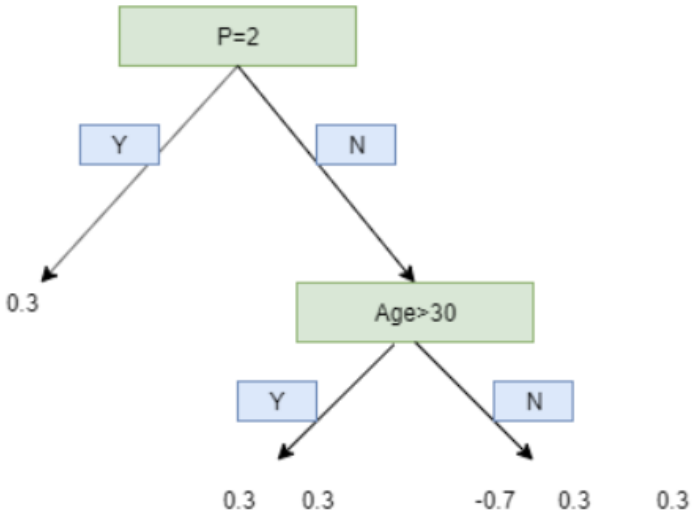

Рис. 2.3. Розгалуження точок даних за допомогою залишкових значень

У прикладі використовується обмеження у дві гілки, однак Gradient Boost має діапазон від 8 до 32 гілок. Прогнози складаються з точки зору логарифму (коефіцієнтів), але ці гілки походять від ймовірності, яка спричиняє невідповідність. Отже, ми не можемо просто додати одне дерево, щоб отримати нові прогнози, оскільки вони отримані з різних джерел. При цьому слід використовувати трансформацію. Найпоширенішою формою перетворення, яка використовується в Gradient Boost for Classification є:

$$\frac{\sum Residual}{\sum [PreviousProb \cdot (1 - PreviousProb)]}. \qquad (2.5)$$

Чисельник у рівнянні (2.5) є сумою залишків у цьому конкретному дереві. Знаменник є сумою (попередня ймовірність передбачення для кожного залишку) * (1 – така сама попередня ймовірність передбачення).

Тепер трансформоване дерево виглядає так, як показано на рис. 2.4.

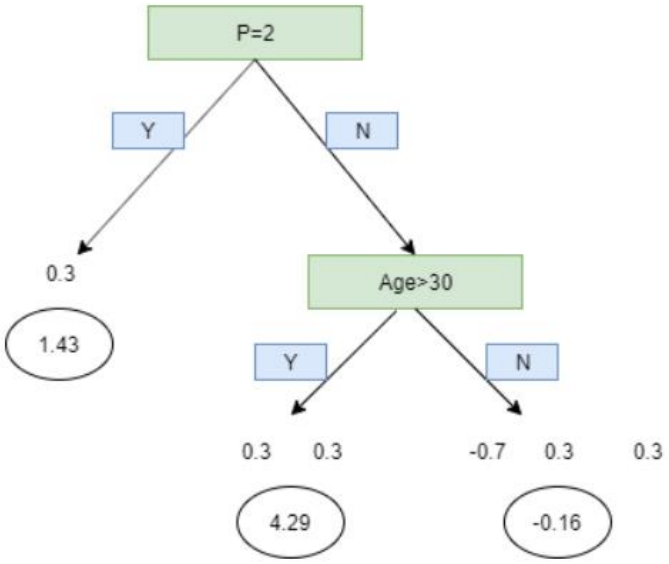

Рис. 2.4. Трансформоване дерево

Тепер, коли дерево трансформували, можна додати до початкового дерева елементи нового дерева зі швидкістю навчання.

$$OldTree + Learning\ Rate \cdot NewTree\,. \qquad (2.6)$$

Швидкість навчання використовується для масштабування внеску у нового дерева. Це призводить до невеликого кроку у бажаному напрямку прогнозування. Емпіричні дані довели, що виконання багатьох маленьких кроків бажаному напрямку призводить до кращого прогнозу за допомогою тестового набору даних, тобто набору даних, який модель ніколи не бачила, порівняно з ідеальним прогнозом на першому кроці. Швидкість навчання зазвичай є невеликим числом, наприклад становить 0,1. Тепер можна розрахувати новий логарифмічний прогноз (коефіцієнти) та відповідно нову ймовірність.

2.5. Градієнтний бустинг для класифікації (XGB Classifier)

XGBoost , що розшифровується як Extreme Gradient Boosting, – це масштабована, розподілена бібліотека машинного навчання з підсиленим градієнтом дерева рішень (GBDT). Він забезпечує паралельне прискорення дерева та є провідною бібліотекою машинного навчання для проблем регресії, класифікації та ранжування.

Для розуміння XGBoost потрібно спершу зрозуміти концепції та алгоритми машинного навчання, на яких базується XGBoost: контрольоване машинне навчання, дерева рішень, ансамблеве навчання та посилення градієнта.

Контрольоване машинне навчання використовує алгоритми для навчання моделі пошуку шаблонів у наборі даних із мітками та функціями, а потім використовує навчену модель для прогнозування міток на функціях нового набору даних (рис. 2.5).

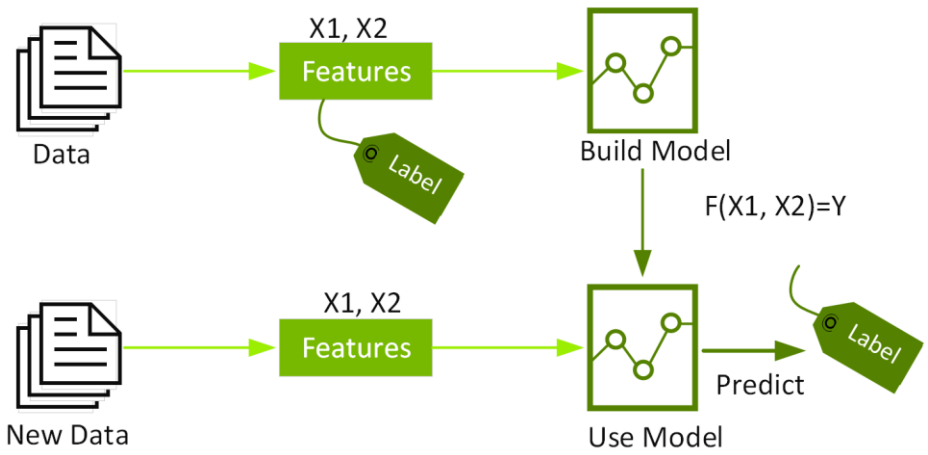

Рис. 2.5. Навчання моделі пошуку шаблонів у наборі даних із мітками та функціями

Дерева рішень створюють модель, яка передбачає мітку шляхом оцінки дерева запитань щодо функції if-then-else true/false та оцінки мінімальної кількості запитань, необхідних для оцінки ймовірності прийняття правильного рішення. Дерева рішень можна використовувати для класифікації, щоб

передбачити категорію, або регресії, щоб передбачити безперервне числове значення. У простому прикладі нижче дерево рішень використовується для оцінки вартості будинку (мітка) на основі розміру та кількості спалень (об'єкти) (рис. 2.6).

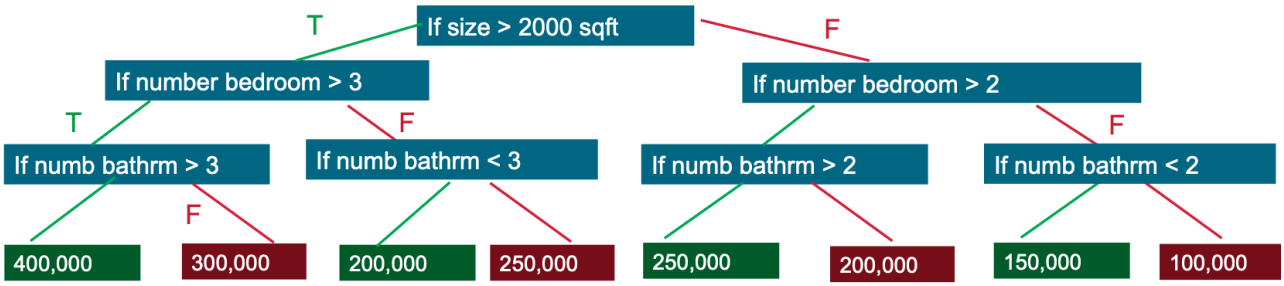

Рис. 2.6. Приклад дерева рішень для оцінки вартості будинку

Градієнтне підсилення дерев рішень (GBDT) – це алгоритм навчання ансамблю дерева рішень, подібний до випадкового лісу, для класифікації та регресії. Алгоритми ансамблевого навчання поєднують кілька алгоритмів машинного навчання, щоб отримати кращу модель.

І випадковий ліс, і GBDT будують модель, що складається з кількох дерев рішень. Різниця полягає в тому, як будуються і комбінуються дерева (рис. 2.7).

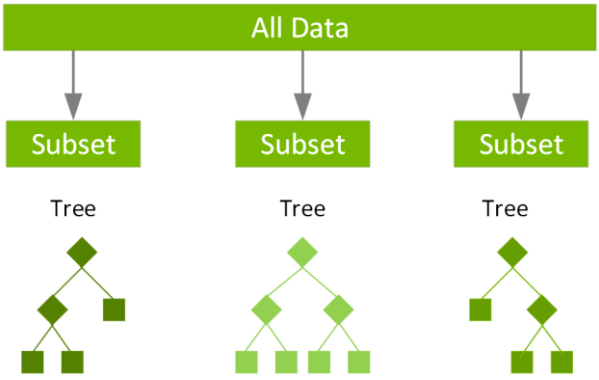

Рис. 2.7. Схема побудови комбінованих дерев

Випадковий ліс використовує техніку під назвою bagging, щоб паралельно побудувати повні дерева рішень із випадкових початкових зразків

набору даних. Остаточний прогноз є середнім значенням усіх передбачень дерева рішень.

Термін «посилення градієнта» походить від ідеї «посилення» або вдосконалення однієї слабкої моделі шляхом поєднання її з кількома іншими слабкими моделями для створення спільної сильної моделі.

Градієнтний бустинг – це розширення бустингу, де процес адитивного генерування слабких моделей формалізовано як алгоритм градієнтного спуску над цільовою функцією.

Підсилення градієнта встановлює цільові результати для наступної моделі з метою мінімізації помилок. Цільові результати для кожного випадку базуються на градієнті помилки (звідси й назва підвищення градієнта) щодо прогнозу.

GBDT ітеративно навчає ансамбль неглибоких дерев рішень, при цьому кожна ітерація використовує залишки помилок попередньої моделі, щоб відповідати наступній моделі. Остаточний прогноз є зваженою сумою всіх прогнозів дерева. Випадкове «розміщення» лісу мінімізує дисперсію та надмірне облаштування, тоді як «посилення» GBDT мінімізує зміщення та недообладнання.

XGBoost – це масштабована та високоточна реалізація посилення градієнта, яка розширює межі обчислювальної потужності для вдосконалених деревоподібних алгоритмів, створена в основному для підвищення продуктивності моделі машинного навчання та швидкості обчислень. За допомогою XGBoost дерева будуються паралельно, а не послідовно, як у GBDT. Він дотримується порівневої стратегії, скануючи значення градієнта та використовуючи ці часткові суми для оцінки якості розбиття на кожному можливому розбиванні в навчальному наборі.

Оцінки передбачень кожного окремого дерева рішень потім підсумовується, щоб отримати потрібний результат. Важливим фактом є те, що два дерева намагаються доповнювати одне одного. Можна записати модель у вигляді:

$$\hat{y}i = \sum_{k=1}^{K} f_k(xi),\ f_k \in F. \qquad (2.7)$$

де K – кількість дерев, f – функціональний простір F – набір можливих CART.

Цільова функція для наведеної вище моделі визначається як:

$$obj(\Theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k). \qquad (2.8)$$

де $\sum_{i}^{n} l(y_i, \hat{y}_i)$ – це функція втрат, $\sum_{k=1}^{K} \Omega(f_k)$ – параметр регуляризації.

Тепер замість того, щоб вивчати дерево відразу, що ускладнює оптимізацію, застосовується адитивна стратегія, що забезпечує мінімізацію втрати на навчання та забезпечує додавання нового дерева, яке можна підсумувати нижче:

$$\begin{aligned}
&\hat{y}_i^{(0)} = 0 \\
&\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
&\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i). \qquad (2.9) \\
&\ldots \\
&\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

Цільову функцію наведеної вище моделі можна визначити як:

$$\begin{aligned}
&obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_i) = \\
&= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant \\
&obj^{(t)} = \sum_{i=1}^{n} \left(y_i - \left(\hat{y}_i^{(t-1)} + f_t(x_i)\right)\right)^2 + \sum_{k=1}^{t} \Omega(f_i) = \\
&= \sum_{i=1}^{n} \left[2\left(\hat{y}_i^{(t-1)} - y_i\right) f_t(x_i) + f_t(x_i)^2\right] + \Omega(f_t) + constant
\end{aligned} \quad . \qquad (2.10)$$

РОЗДІЛ 3.
РЕЗУЛЬТАТИ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОЦІНЮВАННЯ ПЛАТОСПРОМОЖНОСТІ СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВ

3.1. Підготовка та аналіз даних для оцінювання платоспроможності сільськогосподарських підприємств

З метою виконання оцінювання платоспроможності сільськогосподарських підприємств обрано загальнодоступний набір даних [14]. У цих даних подано показники, що лежать в основі оцінювання платоспроможності позичальників банків. Щодня користувачі заповнюють тисячі заявок на кредит, з яких є представники сільськогосподарських підприємств, в тому числі селянських фермерських господарств. Банки можуть окремі замовлення схвалюють, а інші відхиляють. Логіка прийняття їх рішення базується на оцінюванні платоспроможності заявників, що потребує використання відповідних моделей. Для використання технологій машинного навчання слід використовувати історичні дані, що забезпечує знаходження зв’язку між показниками опитування банківських клієнтів та їх окремими характеристика (атрибутами даних).

Із вибраного аналітичного набору даних виділено наступні їх атрибути, що характеризують платоспроможність позичальників:

1. id – ідентифікатор заявника;
2. application_dt – дата подавання заявки на кредит;
3. sample_cd – категорія вибору позичальника;
4. education_cd – освіта позичальника;
5. gender_cd – стать позичальника;
6. age – вік замовника;
7. car_own_flg – наявність у позичальника автомобіля;
8. car_type_flg – наявність у позичальника іноземної техніки;

9. appl_rej_cnt – кількість попередньо відмовлених запитів на кредит;
10. good_work_flg – наявність у позичальника хорошої роботи;
11. Score_bki – бал за даними попередніх кредитних історій;
12. out_request_cnt – кількість попередньо виконаних запитів;
13. region_rating – рейтинг регіону, де проживає позичальник;
14. home_address_cd – домашня адреса позичальника;
15. work_address_cd – робоча адреса позичальника;
16. income – доходи позичальника;
17. SNA – зв'язок позичальника із банком;
18. first_time_cd – наявність історичної інформації про позичальника;
19. Air_flg – наявність закордонного паспорта позичальника;
20. default_flg – показник дефолту за кредитом.

Зазначений набір даних передбачає використання 19 вхідних змінних (X1… X19) та 1 вихідну змінну (Y1). При цьому опрацьовується вибірка наявних заявок на отримання кредитну та оцінення платоспроможності позичальника із аналізом результатів користування кредитом (відбувся дефолт чи ні). При цьому вирішується задача класифікації позичальників завдяки отриманню залежності із наявними даними про позичальників банків та існуючими фактами наявності дефолту, що забезпечує прогнозування на тестових даних. Як метрика використовується ROC-крива. Площа під ROC-кривими є однією із найпопулярніших функціоналів якості у задачах машинного навчання, які стосуються бінарної класифікації.

Для вирішення завдання підготовки та аналізу даних з метою оцінювання платоспроможності сільськогосподарських підприємств використано Jupyter Notebook.

Jupyter являє собою проєкт, мета якого є забезпечення розробки програмного забезпечення, що має відкритий код на підставі використання відкритих стандартів та служб, що забезпечує проведення інтерактивних обчислень на багатьох мовах програмування.

Представлений набір даних щодо оцінювання платоспроможності сільськогосподарських підприємств завантажуємо у блокнот Jupyter Notebook. Для цього виконуємо імпорт потрібних бібліотеки, що представлено на рис. 3.1.

In [4]:
```
import pandas as pd
import numpy as np
from matplotlib import pylab as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn import pipeline
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, FunctionTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from xgboost import XGBClassifier
from sklearn.svm import LinearSVC
from sklearn.model_selection import GridSearchCV
from scipy import stats
from sklearn.metrics import roc_auc_score, roc_curve
import seaborn
from sklearn.utils import shuffle
from matplotlib import pylab as plt
%matplotlib inline
```

Рис. 3.1. Імпорт бібліотек у блокноті Jupyter Notebook

На наступному етапі виконаємо завантаження та виведення масиву початкових даних (рис. 3.2).

In [8]:
```
data = pd.read_csv('application_info.csv') # завантажуємо вибірки
label = pd.read_csv('default_flg.csv') # завантажуємо цільові мітки
```

In [9]:
```
data.head()
```

Out[9]:

	id	application_dt	sample_cd	education_cd	gender_cd	age	car_own_flg	car_type_flg	appl_rej_cnt	good_work_flg	Score_bki	out_request_cnt	region_rati
0	1	01JAN2014	train	SCH	M	27	Y	Y	0	0	-1.917831	0	
1	2	01JAN2014	train	GRD	F	26	N	N	0	0	-1.153144	2	
2	3	01JAN2014	train	SCH	M	35	N	N	0	1	-1.732810	0	
3	4	01JAN2014	train	GRD	F	35	N	N	0	1	-2.552133	2	
4	5	01JAN2014	train	UGR	F	24	N	N	0	0	-1.914581	1	

In [11]:
```
data.drop('sample_cd', axis=1, inplace=True)
data.drop('id', axis=1, inplace=True)
```

Рис. 3.2. Завантаження та виведення масиву початкових даних

На підставі попереднього аналізу встановлено, що є зайві дані в таблиці (sample_cd та id), які у подальшому видалено. Також виконано заміну пропущених даних у полях на «NaN» (не задано). Встановлено, що ознака education_cd (освіта) має перепустки.

Після підготовки даних виконуємо аналіз їх ознак на мінливість за допомогою Q-Q графіка (рис. 3.3).

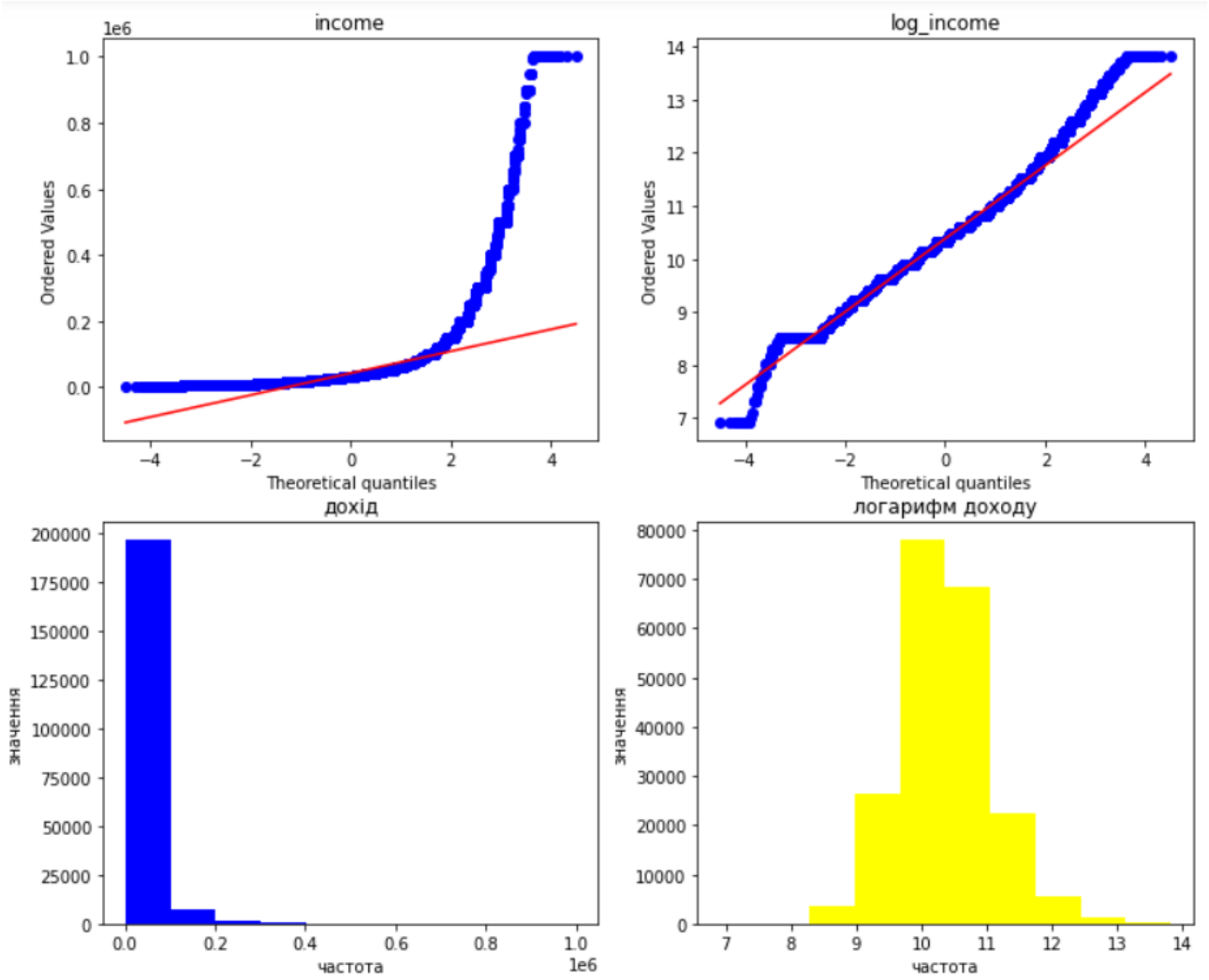

Рис. 3.3. Результати аналізу ознак на мінливість із використанням Q-Q графіка

Отриманий Q-Q графік показує, що логарифмування ознаки income (доходи позичальника) дозволяє можливість її описати нормальним розподілом. При цьому не використовувався критерій Шапіра-Вілка, так як маємо велику вибірку даних, а він застосовується на даних, що не перевищують 5000 спостережень.

У подальшому виконано кодування для бінарних та категоріальних даних та вирішено задачу визначення наявних взаємозв’язків між окремими ознаками даних. Це дало можливість визначити наявні максимальні взаємозв’язки між окремими ознаками даних (табл. 3.1).

Таблиця 3.1. Результати визначення наявних взаємозв'язків між окремими ознаками даних

Ознака	Наявний max коефіцієнт кореляції
home_address_cd	0.740874
work_address_cd	0.740874
car_own_flg	0.700206
car_type_flg	0.700206
log_income	0.367215

У результаті аналіз даних для оцінювання платоспроможності сільськогосподарських підприємств виконано візуалізацію у вигляді кореляційної матриці. Для цього нами побудовано теплову карту із використанням бібліотеки Seaborn (рис. 3.4).

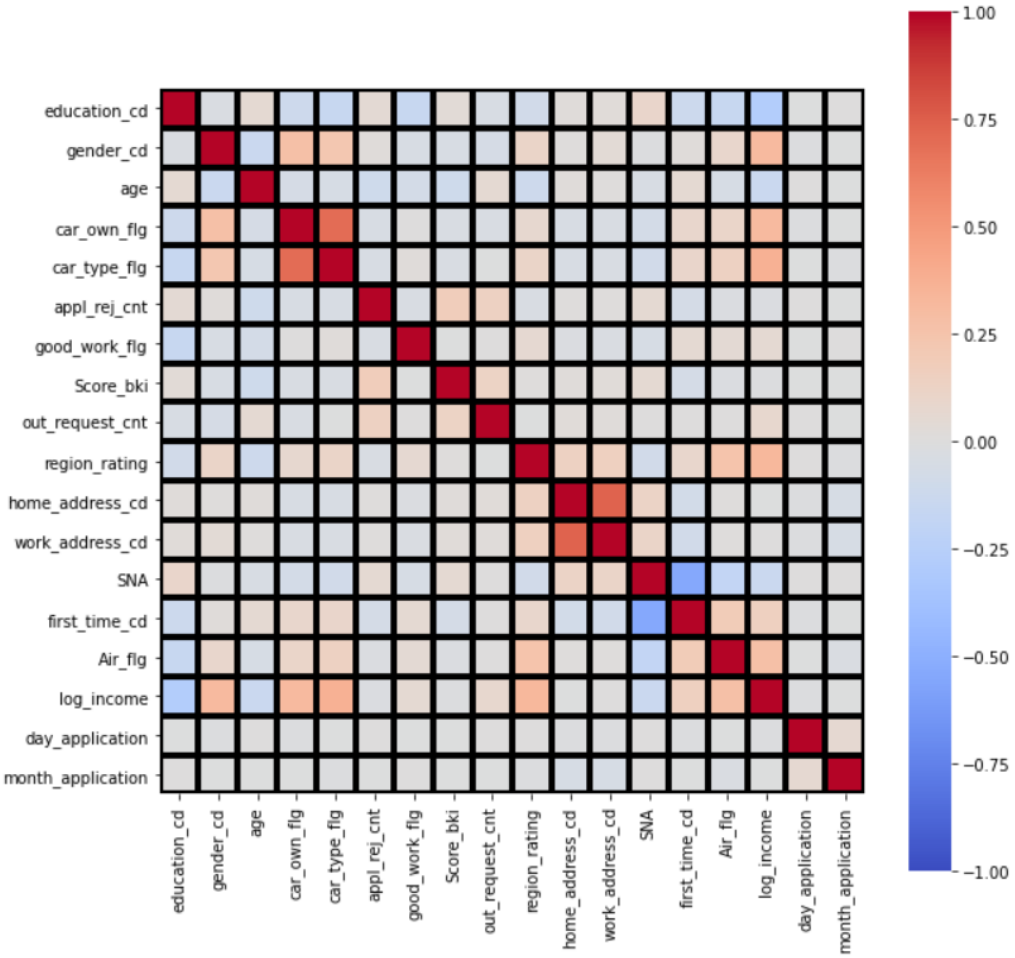

Рис. 3.4. Кореляційна матриця взаємозв'язків між атрибутами даних для оцінювання платоспроможності сільськогосподарських підприємств

Отримана кореляційна матриця взаємозв’язків між атрибутами даних для оцінювання платоспроможності сільськогосподарських підприємств наглядно свідчить про те, що ознак для оцінювання платоспроможності сільськогосподарських підприємств досить мало і ми не викидатимемо взаємозалежні, тим більше в лінійній моделі використовується регуляризація. Однак такі ознаки, як наявність автомобіля та іноземної техніки, а також домашня та робоча адреса сильно корелюються між собою.

На наступному етапі нами виконано розбивку та підготовку даних для оцінювання платоспроможності сільськогосподарських підприємств.

3.2. Створення конвейєра та навчання моделей

На підставі виконаного аналізу представленого набору даних для оцінювання платоспроможності сільськогосподарських підприємств пропонується створити конвеєр, підібрати моделі та виконати їх навчання.

Конвеєр, який використовується у нашій роботі, є основою для оцінювання платоспроможності сільськогосподарських підприємств. Конвеєр використовує компонент «Імпорт даних» замість використання набору даних, щоб показати, як навчати моделі, застосовуючи власні дані.

Параметри конвеєра використовуються для створення універсальних конвеєрів, які можна повторно надіслати пізніше за допомогою різних значень параметрів. Деякі найпоширеніші сценарії – оновлення наборів даних або деяких гіперпараметрів для повторного навчання. Створіть параметри конвеєра для динамічного встановлення змінних під час виконання.

Розподілимо параметри запропонованого конвеєра відносно параметрів джерела даних або модуля конвеєра (рис. 3.5). При повторному надсиланні конвеєра можна вказати значення цих параметрів.

```
In [35]: # для бінарних ознак
binary_data_columns = ['gender_cd', 'car_own_flg', 'car_type_flg', 'good_work_flg', 'Air_flg']
binary_data_indices = np.array([(column in binary_data_columns) for column in X_train.columns], dtype = bool)

In [36]: # для кількісних ознак
numeric_data_columns = ['education_cd','day_application','month_application','appl_rej_cnt','out_request_cnt',
                        'region_rating','home_address_cd','work_address_cd','SNA','first_time_cd']
numeric_data_indices = np.array([(column in numeric_data_columns) for column in X_train.columns], dtype = bool)

In [37]: # для дійсних ознак
float_data_columns = ['Score_bki','log_income','log_age']
float_data_indices = np.array([(column in float_data_columns) for column in X_train.columns], dtype = bool)
```

Рис. 3.5. Параметри джерела даних

У цьому прикладі змінюємо шлях даних для навчання з фіксованого значення на параметр, щоб можна було перенавчити модель на основі інших даних. Можна також додати інші параметри компонента як параметри конвеєра відповідно до варіанта використання (рис. 3.6).

```
In [39]: '''функція конвеєра'''
def PipModel(model):
    pip = pipeline.Pipeline([
        # крок 1: розбивка на види та передобробка
        ('feature_processing', pipeline.FeatureUnion([
            # крок 1.1: речові ознаки (виділення)
            ('binary_variables_processing', FunctionTransformer(lambda d: d[:, binary_data_indices])),
            # крок 1.2: цілі ознаки (виділення)
            ('numeric_variables_processing', pipeline.Pipeline([
                    ('selecting', FunctionTransformer(lambda d: d[:, numeric_data_indices])),
#                     ('scaling', StandardScaler(with_mean = 0))
                    ('scaling', MinMaxScaler())
            ])),
            # крок 1.3: речові ознаки (виділення та стандартизація)
            ('float_variables_processing', pipeline.Pipeline([
                    ('selecting', FunctionTransformer(lambda d: d[:, float_data_indices])),
#                     ('scaling', StandardScaler(with_mean = 0))
                    ('scaling', MinMaxScaler())
            ]))
        ])),
        # крок 2: навчання моделі
        ('model_fitting', model)
    ])
    return pip
```

Рис. 3.6. Результати створення коду для побудови конвеєра

У подальшому вибираємо та задаємо моделі машинного навчання, які будемо використовувати для оцінювання платоспроможності сільськогосподарських підприємств. Зокрема, пропонується використовувати для навчання три моделі (рис. 3.7):

- ✓ RandomForestClassifier (Класифікатор випадкового лісу);
- ✓ GradientBoostingClassifier (Підсилення градієнта для класифікації);
- ✓ XGBClassifier (Градієнтний бустинг для класифікації).

Модель випадкового лісу:

```
In [42]: %%time
         cv_3 = cross_val_score(model_3, X, y, scoring='roc_auc', cv=5)
         print('mean: {:.5f}, std: {:.5f}'.format(cv_3.mean(), cv_3.std()))

         mean: 0.70549, std: 0.00289
         Wall time: 1min 59s
```

Модель градієнтного бустингу sklearn:

```
In [43]: %%time
         cv_4 = cross_val_score(model_4, X, y, scoring='roc_auc', cv=5)
         print('mean: {:.5f}, std: {:.5f}'.format(cv_4.mean(), cv_4.std()))

         mean: 0.73403, std: 0.00179
         Wall time: 1min 50s
```

Модель градиентного бустинга xgboost:

```
In [44]: %%time
         cv_5 = cross_val_score(model_5, X, y, scoring='roc_auc', cv=5)
         print('mean: {:.5f}, std: {:.5f}'.format(cv_5.mean(), cv_5.std()))

         mean: 0.72836, std: 0.00170
         Wall time: 46.4 s
```

Рис. 3.7. Створення моделей машинного навчання для оцінювання платоспроможності сільськогосподарських підприємств

На підставі створених моделей машинного навчання для оцінювання платоспроможності сільськогосподарських підприємств встановлено їх основні показники, які подано у табл. 3.2.

Таблиця 3.2. Результати визначення показників використання створених моделей машинного навчання для оцінювання платоспроможності сільськогосподарських підприємств

Назва моделі	Показники використання моделі		
	mean	std	Wall time
Модель випадкового лісу	0.70549	0.00289	1min 59s
Модель градієнтного бустингу Sklearn	0.73403	0.00179	1min 50s
Модель градієнтного бустинга XGBoost	0.72836	0.00170	46.4 s

На підставі отриманих показників використання створених моделей машинного навчання для оцінювання платоспроможності сільськогосподарських підприємств встановлено, що найкращі результати забезпечує модель градієнтного бустингу XGBoost, навчання якої триває 46.4 сек, математичне сподівання – 0.72836, середньоквадратичне відхилення – 0.00170.

3.3. Підбір параметрів та аналіз якості моделі градієнтного бустингу XGBoost

Модель градієнтного бустингу (XGBoost) – це чудовий спосіб скористатися перевагами алгоритмів машинного навчання підвищення градієнта з відкритим кодом. Ця модель є високошвидкісна і вона добре зарекомендувала себе у різних задачах.

У подальшому виконуємо ансамблювання алгоритмом, тобто використовуватимемо а не одну модель XGBoost. Це забезпечить можливість підібрати найкращі базові алгоритми стекінгу. Стекінг (Stacked Generalization або Stacking) полягає у використанні базових класифікаторів, що забезпечують прогнозування (мета-ознак) та використання їх як ознак для деякого «узагальнюючого» алгоритму (мета-алгоритму). Тобто, основною ідеєю стекінгу є перетворення вихідної ознаки відповідно до завдання у нову ознаку, точками якої є прогнозування базових алгоритмів.

Для виконання стекінгу насамперед виконується вибір окремих пар довільних підмножин даних із навчальної вибірки. Після цього кожну із них слід навчити за базовими алгоритми та прогнозувати із їх використанням цільову змінну. Отримані значення стають об'єктами виконання нового прогнозування.

На наступному етапі виконуємо обґрунтування оптимальної кількості дерев та їх глибини (рис. 3.8).

In [38]:
```
%%time
'''підбір кількості дерев'''
estimators = list(range(30, 305, 5))
scoring_estimators = []

for i in estimators:
    clf = XGBClassifier(n_estimators=i)
    score = cross_val_score(clf, X, y, scoring='roc_auc', cv=3).mean()
    scoring_estimators.append(score)
```

```
Wall time: 44min 35s
```

In [40]:
```
%%time
'''підбір глибини дерев'''
depth = list(range(1,11,1))
scoring_depth_1 = []
scoring_depth_2 = []

for i in depth:
    clf = XGBClassifier(max_depth=i, n_estimators=100)
    score = cross_val_score(clf, X, y, scoring='roc_auc', cv=3).mean()
    scoring_depth_1.append(score)
    clf = XGBClassifier(max_depth=i, n_estimators=200)
    score = cross_val_score(clf, X, y, scoring='roc_auc', cv=3).mean()
    scoring_depth_2.append(score)
```

```
Wall time: 13min 46s
```

Рис. 3.8. Обґрунтування оптимальної кількості дерев та їх глибини

У результаті обґрунтування оптимальної кількості дерев та їх глибини, нами побудовано залежності якості навчання від кількості дерев та їх глибини (рис. 3.9).

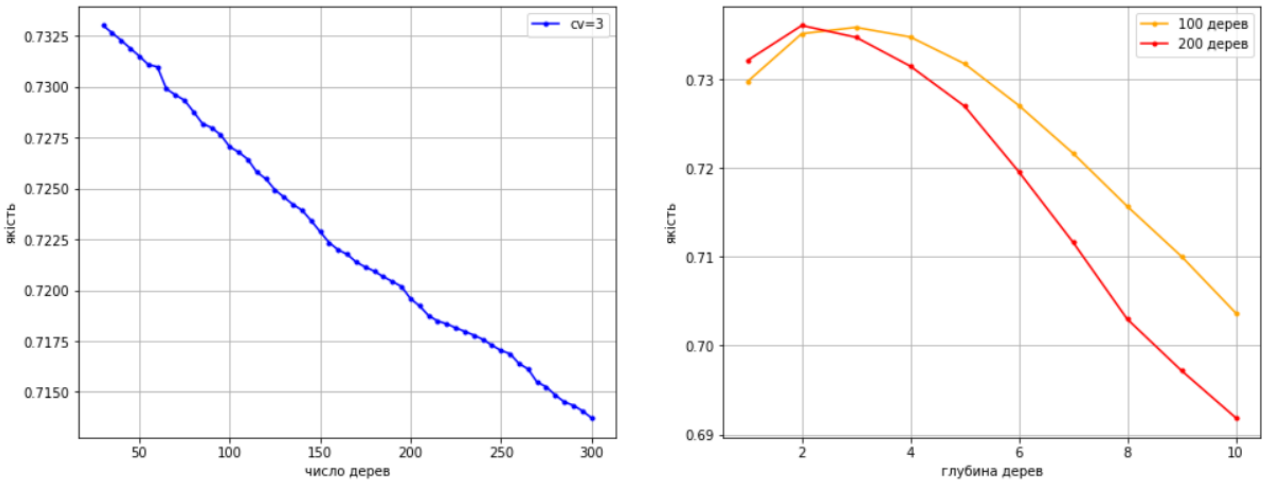

Рис. 3.9. Залежності якості навчання від кількості дерев та їх глибини

У результаті отриманих результатів встановлено, що зі зростанням кількості дерев, якість навчання знижується. Водночас, глибини дерев якість навчання залежить від кількості дерев. За використання 100 дерев оптимальна їх глибина становить – 3, а за 300 – 2.

У подальшому виконаємо ансамблювання алгоритмів за допомогою стекінгу. Нами виконано тестування ансамблю на навчальній та валідаційній вибірці. У результаті виконаних досліджень встановлено залежність AUC ROC від кількості фолдів (рис. 3.10).

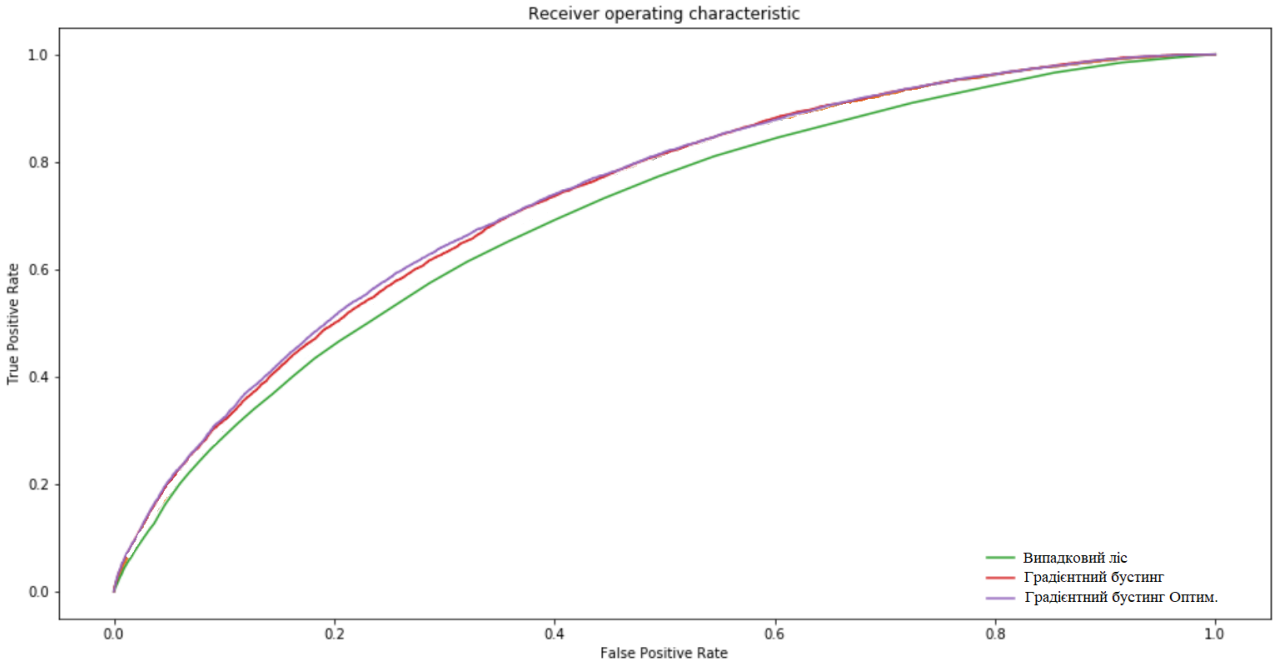

Рис. 3.10. Залежність AUC ROC від кількості фолдів

На підставі залежності AUC ROC від кількості фолдів встановлено, що ансамблювання не дозволило можливості покращити якість класифікації, у порівнянні із моделлю XGBoost із обґрунтованими параметрами. Можливо це пов'язано з тим, що ансамбль використовує тільки один вид алгоритму (градієнтний бустинг), проте якість інших алгоритмів сильно поступається цьому. Отже, приймаємо, що в інтелектуальній інформаційній системі оцінювання платоспроможності сільськогосподарських підприємств використовуватимемо просто модель XGBoost із обґрунтованими параметрами.

3.4. Архітектура інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств

Нами вибрано клієнт-серверну архітектуру інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств (рис. 3.11). При цьому сервер встановлений у регіональному відділенні і надає дані для ПК та програм, які встановлені у осіб, що приймають рішення (клієнтів).

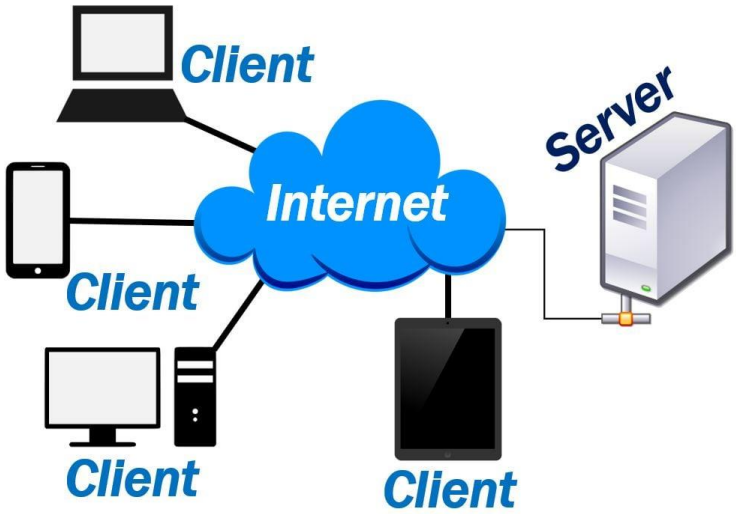

Рис. 3.11. Архітектура інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств

Дані щодо позичальників, надаюся клієнтам через глобальну мережу Інтернет. У запропонованій інтелектуальній інформаційній системі оцінювання платоспроможності сільськогосподарських підприємств сервер являє собою пристрій у мережі, який керує мережевими ресурсами. Сервер при цьому є віддаленим і не виконує жодних інших завдань, окрім завдань сервера – збереження та надання доступу до своїх обчислювальних і дискових ресурсів, а також доступу до наявних сервісів. При цьому він працює цілодобово, або ж у час роботи заданої групи користувачів, які оцінюють платоспроможність сільськогосподарських підприємств.

Пропонується у інтелектуальній інформаційній системі оцінювання платоспроможності сільськогосподарських підприємств встановити сервер бази

даних. Він забезпечить доступ і отримання даних із бази даних. При цьому можна отримати доступ до бази даних за допомогою «інтерфейсу» окремих уористувачів, який запускається локально на ПК користувачів, або ж «бекенду», який є запущеним у самій базі даних, до якої має доступ будь-який віддалений ПК користувача.

Рис. 3.12. Структура зв’язку між клієнтом та сервером бази даних

Потрібна для ОПР інформація із бази даних отримується шляхом запуску запиту, а потім виводиться користувачеві, який запитує дані у вигляді відповіді. Якщо система зростає у масштабах, то у сервері формується база даних для зберігання потрібної інформації, а користувачам можна отримати доступ до цих даних, виконавши запит за допомогою будь-якої мови запитів.

На ПК клієнтів встановлюється прикладне програмне забезпечення, яке базується на обґрунтованій моделі градієнтного бустингу XGBoost, що забезпечує оцінювання платоспроможності сільськогосподарських підприємств.

РОЗДІЛ 4.
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Аналіз небезпечних і шкідливих виробничих чинників та розробка заходів щодо покращення умов праці

Потенційно шкідливим чинником кабінету особи, яка приймає управлінські рішення, вважається небезпека враження людини електричним струмом. Важливим, але менш ймовірним чинником являється пожежна небезпека під час аварійної ситуації. Хімічні та біологічні джерела практично не мають впливу.

Перелік небезпечних та шкідливих виробничих чинників наведено у таблиці 5.1.

Таблиця 5.1. Небезпечні та шкідливі виробничі чинники

Фізичні	Електронебезпека, пожежа, шум, мікроклімат
Хімічні	Відсутні
Біологічні	Відсутні
Психофізіологічні	Відсутні

В приміщенні кабінету особи, яка приймає управлінські рішення, присутні небезпечні чинники, та за умов дотримання заходів безпеки, вони не є критичним.

4.2. Розробка логічно-імітаційної моделі процесу виникнення травм під час монтажу інтелектуальної інформаційної системи

Проаналізувавши кожну із логічних моделей процесів формування та можливого виникнення травмонебезпечних та аварійних ситуацій, завжди

можна знайти подію з якої починається небезпечний процес ще до виникнення небезпечних наслідків.

Методикою оцінки рівня небезпеки робочих місць, машин, виробничих процесів та окремих виробництв передбачено пошук об'єктивного критерію рівня небезпеки для конкретного об'єкта. Таким показником вибрана ймовірність виникнення аварії, травми залежно від досліджуваного явища.

Для оцінки рівня небезпеки певного об'єкта чи явища можна застосувати метод обчислення ймовірності виникнення будь-якого випадкового явища, який широко застосовують в зарубіжній інженерній практиці. Основні його принципи полягають в тому, що на основі обстеження робочого місця чи окремої машини виявляють виробничі небезпеки, можливі аварійні або травматичні ситуації. При оцінці ситуацій визначають події, які можуть стати головною подією при побудові логічно-імітаційної моделі травми. Після цього будують модель "дерева відмов і помилок оператора". При цьому важливе значення має правильний вибір головної події.

Головну подію (травма), модель якої нам необхідно побудувати, вибирають виходячи з оцінки відповідного об'єкта, виробництва чи окремої одиниці обладнання і змісту його найбільш небезпечного явища, яке за певних умов виробництва може виникнути.

Після вибору головного випадкового явища (події) розпочинаємо побудову моделі ("дерева"). Використовуючи оператора "і" та "або", використовуємо набір ситуацій (відомих до цього), які можуть призвести до подій, вибраної як головна.

Після визначення відповідних травмонебезпечних ситуацій та їх кількості, визначаємо інші події, що входять до кожної такої ситуації, логічним аналізом із застосуванням операторів "і", "або" та інших. Процес побудови моделі триває, поки не будуть знайдені усі базові події, що визначають межу моделі.

Слід мати на увазі, що кожна випадкова подія, до якої входять базові події, може формуватися й виникати при входженні у неї двох, трьох і більше базових подій за допомогою відповідних операторів.

Повністю побудована і перевірена модель підлягає математичній обробці для визначення ймовірності кожної випадкової події, що увійшла до моделі, починаючи з базових і закінчуючи головною.

Ймовірність базових подій визначаємо за даними виробництва. Наприклад, базова подія “стан контролю з охорони праці”. Для визначення ймовірності ми повинні встановити, наскільки (у відсотках) від ідеального рівня здійснюється відповідний контроль на об’єкті. Якщо буде встановлено, що такий рівень контролю становить 50% або 30%, то ймовірність відповідно дорівнює 0,5 і 0,3. При відсутності контролю ймовірність “не здійснення контролю” становитиме 1, якщо контроль ідеальний, то відповідно ймовірність дорівнює 0.

Після обчислення ймовірності всіх подій, розміщених у ромбах, і базових подій, починаючи з лівої нижньої гілки “дерева”, позначаємо номерами всі випадкові події, що увійшли до моделі.

На цьому можна вважати, що певна модель підготовлена до математичних обчислень ймовірностей випадкових подій логічно-імітаційної моделі

Отже, для побудови логіко-імітаційної моделі процесу, формування і виникнення аварії та травми під час монтажу інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств складемо список базових подій. Вони лежатимуть у основі даної моделі. Кожному пункту списку присвоюємо певне значення ймовірності виникнення. Нижче подано сам список:

1. Стан контролю з охорони праці ..... **$P_1 = 0,2;$**
2. Несерйозне відношення до проходження ТО інструменту **$P_2 = 0,1;$**
3. Відсутність комплектуючих установки..... **$P_3 = 0,2;$**
4. Невисока міцність ..... **$P_4 = 0,03;$**

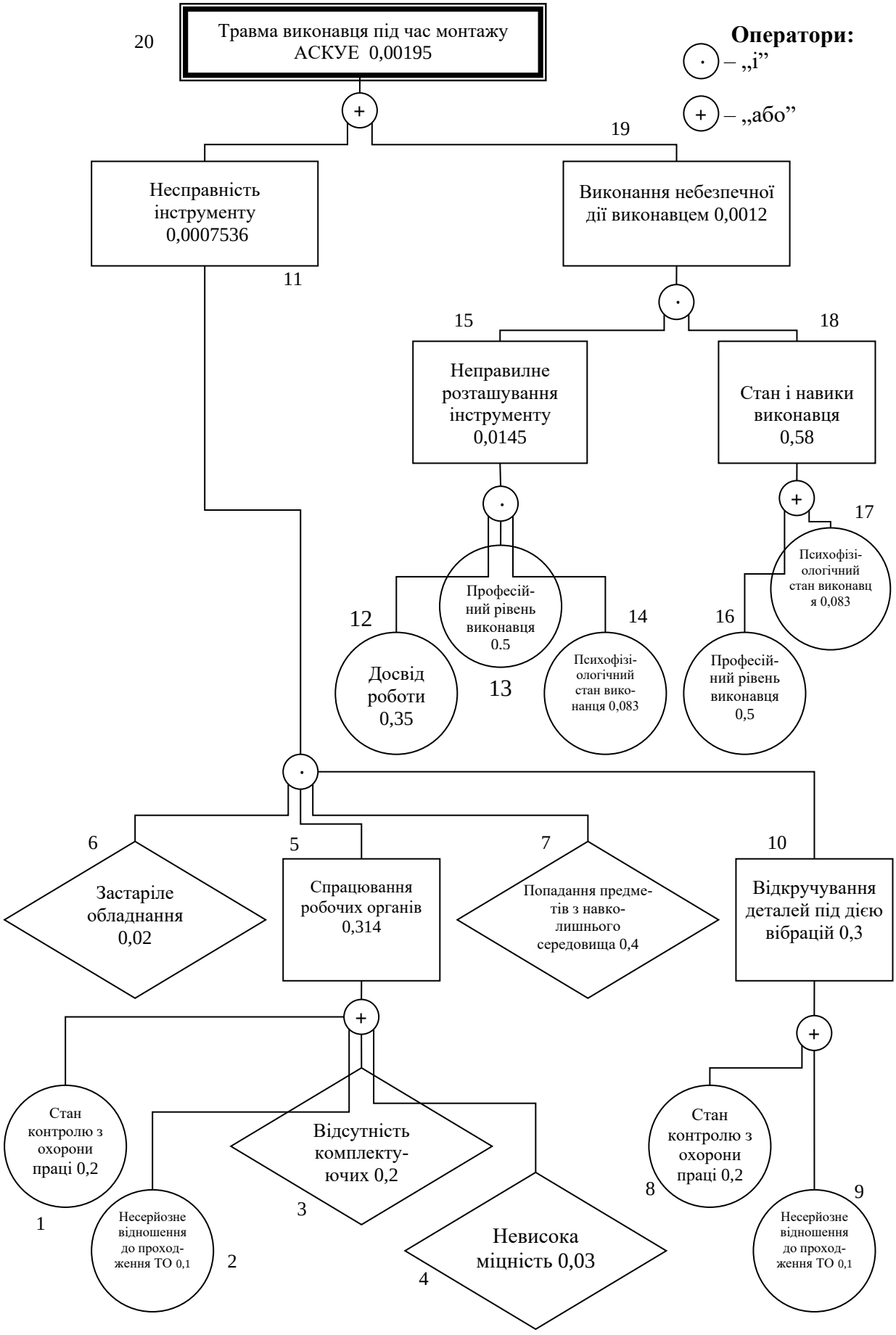

Рис. 4.1. Логіко-імітаційна модель процесу формування та виникнення аварії та травми під час монтажу інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств

1. Використання застарілого обладнання..... **P_6 = 0,02;**
2. Попадання сторонніх предметів ..... **P_7 = 0,4;**
3. Досвід роботи виконавця ..... **P_{12} = 0,35.**
4. Професійний рівень виконавця ..... **P_{13} = 0,5;**
5. Психофізіологічний стан виконавця..... **P_{14} = 0,083;**

На основі даного списку будуємо матрицю логічних взаємозв'язків між окремими пунктами, графічне представлення якої зображено на рис. 4.1.

Розрахуємо ймовірності виникнення подій, що входять у дану логіко-імітаційну модель процесу монтажу інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств (на прикладі ймовірності отримання травми виконавця).

Ймовірність виникнення події P_5 визначаємо наступним чином:

$$P_5 = 02 + 0{,}1 + 0{,}2 + 0{,}003 - 0{,}2 \cdot 0{,}1 - 0{,}2 \cdot 0{,}03 - 0{,}2 \cdot 0{,}03 - 0{,}1 \cdot 0{,}2 - 0{,}1 \cdot 0{,}03 -$$
$$- 0{,}2 \cdot 0{,}03 + 0{,}2 \cdot 0{,}1 \cdot 0{,}2 + 0{,}1 \cdot 0{,}2 \cdot 0{,}03 + 0{,}2 \cdot 0{,}1 \cdot 0{,}2 + 0{,}2 \cdot 0{,}1 \cdot 0{,}03 -$$
$$- 0{,}2 \cdot 0{,}1 \cdot 0{,}2 \cdot 0{,}03 = 0{,}314$$

Ймовірність виникнення події P_{10} визначаємо так:

$$P_{10} = 0{,}2 + 0{,}1 = 0{,}3.$$

Ймовірність виникнення події P_{11} визначаємо:

$$P_{11} = 0{,}02 \cdot 0{,}314 \cdot 0{,}4 \cdot 0{,}3 = 0{,}00075.$$

Ймовірність виникнення події P_{15} визначаємо наступним чином:

$$P_{15} = 0{,}35 \cdot 0{,}5 \cdot 0{,}083 = 0{,}0145.$$

Ймовірність події P_{18}:

$$P_{18} = 0{,}5 + 0{,}083 = 0{,}58.$$

Ймовірність події P_{19}:

$$P_{19} = 0{,}0145 \cdot 0{,}083 = 0{,}0012.$$

Ймовірність події P_{20}:

$$P_{20} = 0{,}00075 + 0{,}0012 = 0{,}00195.$$

Ймовірність травми рівна ймовірності виникнення аварії, бо остання можлива лише за умови монтажу автоматизованої системи управління енергоспоживанням людиною.

Логіко-імітаційні моделі аварій і травм допомагають зменшити ймовірність виникнення аварійних та травмонебезпечних ситуацій. Якщо необхідно оцінити рівень небезпеки будь-якого робочого місця, слід уважно вивчити і побудувати логічні моделі можливих небезпечних ситуацій, які охоплюють як стан обладнання і самого робочого місця, так і поведінку працюючого і обчислити ймовірність виникнення травми.

Після аналізу результатів моделювання ймовірність виникнення травми можна звести до дуже малої величини – достатньо зменшити вплив ймовірностей вихідних факторів, які до неї призводять.

4.3. Розробка заходів щодо безпеки у надзвичайних ситуаціях

Науково-технічний прогрес радикально змінив світ, породивши нові загрози для цивілізації. У житті сучасної людини все більше місце займають турботи, пов'язані з подоланням різних кризових явищ, що виникають в процесі розвитку земної цивілізації. В Україні, як і в усьому світі, в останні роки спостерігається зростання числа військових дій, катастроф природного та техногенного характеру. Це обумовлено, перш за все, прогресуючої урбанізацією територій, збільшенням щільності населення Землі, і, як наслідок, збільшенням антропогенного навантаження на навколишнє середовище. Захист природних систем і населення від надзвичайних ситуацій різного характеру сформувалася в останні роки як нагальна і об'єктивна потреба суспільства і держави.

Заходи щодо захисту цивільного населення плануються проводяться по населених пунктах де розміщені підприємства і охоплюють населення навколишніх сіл. Водночас характер та зміст захисних засобів встановлюються

вид ступеня загрози, місцевих умов з урахуванням важливості виробництва для безпеки населення і інших економічних і соціальних чинників.

Основні заходи щодо захисту населення плануються та здійснюються завчасно і мають випереджувальний характер, це стосується насамперед підготовки, підтримання у постійній готовності індивідуальних та колективних засобів захисту, їх накопичення, а також підготовки до проведення евакуації населення із зон підвищеного ризику.

Також раз в три роки проводяться навчання по підготовці близьких до військових дій, що в разі небезпеки могло би не дістати людину зненацька. Керівництво докладає максимум зусиль, щоб працівники підприємств були хоча би мінімально захищенні в разі будь-якої небезпеки пов’язаної з тими чи іншими обставинами.

РОЗДІЛ 5.
ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОЇ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Економічний ефект від використання запропонованої інтелектуальної інформаційної системи отримується завдяки точному оцінюванню платоспроможності сільськогосподарських підприємств, що забезпечує зниження втрат від несвоєчасного повернення кредитів та використання банками штрафних санкцій.

Початковими даними для проведення розрахунків економічного ефекту від використання запропонованої інтелектуальної інформаційної системи є характеристики цієї системи. Собівартість програмних та технічних засобів інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств передбачає врахування вартості розроблення програмного продукту, придбання основних технічних складових та виконання монтажних робіт. На підставі проведених розрахунків встановлено, що вартість програмних та технічних засобів інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств становить – 25800 грн.

Для забезпечення рентабельності запропонованої інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств $Р_м$=10% вартість цієї системи становить:

$$Ц_м = C_п + C_п \cdot (P_м / 100). \qquad (5.1)$$

Підставивши значення у (5.1) маємо:

$$Ц_м = 25800 + 25800 \cdot (10 / 100) = 28380 \text{ грн.}$$

Балансова вартість розробленої інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств визначається враховуючи її монтаж на налаштування:

$$C_{бал} = Ц \cdot K_{тп}, \qquad (5.2)$$

де $Ц$ – вартість складових інформаційної системи; $K_{тп}$ – коефіцієнт витрат ($K_{тп} = 1.05$).

Підставивши відповідні значення у формулу (5.2) отримаємо вартість інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств:

$$C_{бал} = 28380 \cdot 1,05 = 29799 \text{ грн.}$$

Використання розробленої інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств потребує поточних витрат (річних витрат на її утримання). Їх складові враховують заробітну плату, амортизаційні відрахування, витрати коштів на електричну енергію, а також її обслуговування.

Експлуатаційні витрати під час утримування інформаційної системи становлять 22364грн. Розроблення потрібного програмного забезпечення для інформаційної системи та її тестування становить 35600грн.

Повна собівартість інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств для окремої організації (банківської установи) складає 65399 грн.

Економічна ефективність інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств для організації (банківської установи), визначається із використанням формули:

Таблиця 5.1. Результати визначення економічної ефективності від використання інтелектуальної інформаційної системи оцінюванню платоспроможності сільськогосподарських підприємств для організації (банківської установи)

№ п/п	Назва показників	Одиниця вимірювання	Значення
1	Вартість інтелектуальної інформаційної системи	грн.	25800
2	Експлуатаційні витрати	грн.	22364
3	Розроблення програмного забезпечення та тестування	грн.	35600
4	Собівартість інтелектуальної інформаційної системи	грн.	65399
5	Приведені затрати на функціонування інтелектуальної інформаційної системи	грн./рік	28904
6	Економічна ефективність	грн./рік	117071
7	Термін окупності капіталовкладень	років	0,25

$$E_{ІСППР} = (П_1 - П_2) - З_{ІСППР}, \quad (5.3)$$

де $П_1$ – обсяг втрат організації (банківською установою) із інтелектуальною інформаційною системою оцінювання платоспроможності сільськогосподарських підприємств, грн.; $П_2$ – обсяг втрат організацією (банківською установою) без використання інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств, грн.; $З_{ІСППР}$ – річні приведені витрати на інтелектуальну інформаційну систему оцінюванню платоспроможності сільськогосподарських підприємств.

Річні приведені витрати на функціонування інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств становлять:

$$З_{ІСППР} = E_н \cdot C_{бал} + B_р . \quad (5.4)$$

Підставивши значення у формулу (5.4) маємо річні приведені витрати на функціонування інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств:

$$З_{ІСППР} = 0,1 \cdot 65399 + 22364 = 28904 \text{ грн.}$$

Підставивши значення у (5.3) маємо економічну ефективність від функціонування інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств:

$$E_{ІСППР} = (178320 - 62345) - 28904 = 117071 \text{ грн.}$$

Термін окупності капіталовкладень у інтелектуальну інформаційну систему оцінюванню платоспроможності сільськогосподарських підприємств визначається за формулою:

$$T_{ок} = \frac{З_{ІСППР}}{E_{ІСППР}} . \quad (5.5)$$

Підставивши значення у (5.5) маємо:

$$T_{ок} = \frac{28904}{117071} = 0,25 \text{ року.}$$

ВИСНОВКИ І ПРОПОЗИЦІЇ

Проведений аналіз стану оцінювання платоспроможності організацій та використовуваних нами інформаційних систем та технологій свідчить про те, що актуальними є технології використання великих даних, а на їх основі технологій машинного навчання для дослідження та розробки інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання.

На підставі виконаного аналізу зауважено, що бібліотека ScikitLearn має потрібні методи машинного навчання, які забезпечують оцінювання платоспроможності сільськогосподарських підприємств на підставі наявних даних. Це лежить в основі розробки інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств із вибором ефективних алгоритмів машинного навчання.

На виконаного аналізу встановлено, що використання наявного інструментарію оцінювання платоспроможності сільськогосподарських підприємств знижує точність отриманих рішень через їх недоліки. Це зумовлює потребу розроблення інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств. Для цього у кваліфікаційній роботі існує потреба у розв’язанні завдань, що стосуються розробки інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств.

Нами обґрунтовано особливості вирішення задач класифікації. Встановлено, що у Deep Learning задачі класифікації вирішуються шляхом навчання моделей класифікації. Моделі класифікації навчаються шляхом надання об’єктів та їхніх міток. Точність моделі визначається на основі правильно передбачених міток.

Для оцінювання якості бінарної класифікації використовують ROC-криві, які являють собою графіки, що відображають співвідношення між часткою об'єктів від сумарної кількості носіїв певної ознаки. При цьому вони повинні

бути правильно класифіковані до сумарної кількості об'єктів, які немають ознаки, тобто помилково класифіковані, як такі що мають ознаку. AUC означає «Площа під кривою ROC». Тобто AUC вимірює всю двовимірну площу під усією кривою ROC (інтегральне числення) від (0,0) до (1,1).

Нами виконано вибір та аналіз основних алгоритмів класифікації, які у подальшому використовуватимемо для дослідження та обґрунтування раціонального, що дозволить точно оцінювати платоспроможність сільськогосподарських підприємств.

Нами виконано оцінювання платоспроможності сільськогосподарських підприємств на підставі набору даних. У цих даних подано показники, що лежать в основі оцінювання платоспроможності позичальників банків. Цей набір даних передбачає використання 19 вхідних змінних (X1… X19) та 1 вихідну змінну (Y1), які характеризують заявки на отримання кредитну та оцінення платоспроможності позичальника із аналізом результатів користування кредитом (відбувся дефолт чи ні).

Для вирішення завдання підготовки та аналізу даних з метою оцінювання платоспроможності сільськогосподарських підприємств використано Jupyter Notebook. Це забезпечило побудову Q-Q графіка який показує, що логарифмування ознаки income (доходи позичальника) дозволяє можливість її описати нормальним розподілом. При цьому не використовувався критерій Шапіра-Вілка, так як маємо велику вибірку даних, а він застосовується на даних, що не перевищують 5000 спостережень.

У результаті аналіз даних для оцінювання платоспроможності сільськогосподарських підприємств виконано візуалізацію у вигляді кореляційної матриці. Для цього нами побудовано теплову карту із використанням бібліотеки Seaborn (рис. 3.4). Отримана кореляційна матриця взаємозв’язків між атрибутами даних для оцінювання платоспроможності сільськогосподарських підприємств наглядно свідчить про те, що ознак для оцінювання платоспроможності сільськогосподарських підприємств досить мало і ми не викидатимемо взаємозалежні, тим більше в лінійній моделі

використовується регуляризація. Однак такі ознаки, як наявність автомобіля та іноземної техніки сильно корелюються між собою.

У подальшому вибираємо та задаємо моделі машинного навчання, які будемо використовувати для оцінювання платоспроможності сільськогосподарських підприємств. Зокрема, пропонується використовувати для навчання три моделі (рис. 3.7): RandomForestClassifier (Класифікатор випадкового лісу); GradientBoostingClassifier (Підсилення градієнта для класифікації); XGBClassifier (Градієнтний бустинг для класифікації).

На підставі створених моделей машинного навчання для оцінювання платоспроможності сільськогосподарських підприємств встановлено їх основні показники, які подано у табл. 3.2. Встановлено, що найкращі результати забезпечує модель градієнтного бустингу XGBoost, навчання якої триває 46.4 сек, математичне сподівання – 0.72836, середньоквадратичне відхилення – 0.00170.

Нами виконано ансамблювання алгоритмом, тобто використовуватимемо а не одну модель XGBoost. Це забезпечить можливість підібрати найкращі базові алгоритми стекінгу. Стекінг (Stacked Generalization або Stacking) полягає у використанні базових класифікаторів, що забезпечують прогнозування (мета-ознак) та використання їх як ознак для деякого «узагальнюючого» алгоритму (мета-алгоритму).

У результаті обґрунтування оптимальної кількості дерев та їх глибини, нами побудовано залежності якості навчання від кількості дерев та їх глибини (рис. 3.9). Встановлено, що зі зростанням кількості дерев, якість навчання знижується. Водночас, глибини дерев якість навчання залежить від кількості дерев. За використання 100 дерев оптимальна їх глибина становить – 3, а за 300 – 2.

На підставі залежності AUC ROC від кількості фолдів встановлено, що ансамблювання не дозволило покращити якість класифікації, у порівнянні із моделлю XGBoost із обґрунтованими параметрами. Можливо це пов'язано з тим, що ансамбль використовує тільки один вид алгоритму (градієнтний

бустинг), проте якість інших алгоритмів сильно поступається цьому. Отже, приймаємо, що в інтелектуальній інформаційній системі оцінювання платоспроможності сільськогосподарських підприємств використовуватимемо просто модель XGBoost із обґрунтованими параметрами.

Нами вибрано клієнт-серверну архітектуру інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств (рис. 3.11). При цьому сервер встановлений у регіональному відділенні і надає дані для ПК та програм, які встановлені у осіб, що приймають рішення (клієнтів).

Обґрунтовані заходи щодо охорони праці та безпеки у надзвичайних ситуаціях передбачають створення безпечних умов праці під час роботи з інтелектуальною інформаційною системою оцінювання платоспроможності сільськогосподарських підприємств.

На підставі виконаних розрахунків економічної ефективності встановлено, що використання запропонованої інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств у окремій фінансовій організації дасть можливість отримати економію коштів у розмірі 117071 грн. Показник ефективності запропонованої інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств становить 0,56.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрушків Т. Проблеми оцінки кредитоспроможності позичальників в управлінні кредитним ризиком банку. *Світ фінансів*. 2008. № 2(15). С. 113-118.

2. Бордюг В.В. Теоретичні основи оцінки кредитоспроможності позичальника банку. Вісник *Університету банківської справи Національного банку України*. 2008. № 3. С. 112-115.

3. Введення в машинне навчання за допомогою Python и Scikit-Learn. URL: https://habr.com/ua/company/mlclass/blog/247751/ (дата звернення: 20.05.2022).

4. Григорович О.В. Застосування багатошарових персептронів для класифікації позичальників юридичних осіб. Нейронечіткі технології моделювання в економіці. Науково-аналітични журнал. Київ, 2019. №8. С.48-64.

5. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.

6. Класифікація в Python з Scikit-Learn та Pandas. URL: https://stackabuse.com/classification-in-pythonwith-scikit-learn-and-pandas/ (дата звернення: 17.05.2022).

7. Клебан Ю.В. Дослідження способів трансформації даних в контексті підвищення ефективності моделей кредитного скорингу. Нейронечіткі технології моделювання в економіці. Науково-аналітични журнал. Київ, 2019. №8. С.94-123.

8. Кравченко В.П., Кравченко В.І. Удосконалення оцінки кредитоспроможності позичальника. *Наукові праці КНТУ Економічні науки*. 2010. № 17. С. 11-15.

9. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.

10. Лутц М. Программирование на Python. I том. СПб.: Символ-плюс, 2015. 992 с.

11. Марк Саммерфилд. Программирование на Python 3. Подробное руководство. Пер. с англ. СПб.: Символ-Плюс, 2013. 608 с.

12. Матвійчук А. В. Штучний інтелект в економіці: нейронні мережі, нечітка логіка: монографія. Київ, КНЕУ, 2011. 439 с.

13. Методичні вказівки з інспектування банків «Система оцінки ризиків»: Постанова Правління Національного банку України від 15.03.2004 № 104. URL: https://zakon.rada.gov.ua/laws/sho w/v0104500-04.

14. Набір даних щодо анкетних даних заявників та факту наявності дефолту. URL: https://www.kaggle.com/competitions/fintech-credit-scoring/data

15. Навчання нейромережі з учителем, без вчителя, з підкріпленням – у чому відмінність? URL: https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/(дата звернення: 22.10.2022).

16. Новоселецький О.М., Якубець О.В. Моделювання кредитоспроможності юридичних осіб на основі дискримінантного аналізу та нейронних мереж. Нейронечіткі технології моделювання в економіці. Науково-аналітични журнал. Київ, 2014. №3. С.120-151.

17. Огляд методів класифікації у машинному навчанні за допомогою Scikit-Learn. URL: https://tproger. ru/translations/scikit-learn-in-python/s://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/ (дата звернення: 10.05.2022).

18. Остафіль О., Рубаха М. Комплексна оцінка кредитоспроможності позичальника як інструмент управління кредитним ризиком банку. *Формування ринкової економіки в Україні*. 2009. № 19. С. 387–396.

19. Плескач В.Л., Рогушина Ю.В., Кустова Н.П. Інформаційні технології та системи. К.: Книга, 2004. 519 с.

20. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. СПб.: БХВ-Петербург, 2012. 704 с.

21. Смолева Т. Сучасні методи оцінки кредитоспроможності позичальників банками України. Фінанси, облік, банки. 2014. №1(20). С. 241-245.

22. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. М.: Альт Линукс, 2010. 126 с.

23. Abdou H., Pointon J. Credit scoring, statistical techniques and evaluation criteria: a review of the literature, Intell. Syst. Account., Finance Manage. 18 (2–3) (2011) 59–88.

24. Ahmed A.M., Rizaner A., and Ulusoy A.H., A Decision Tree Algorithm Combined with Linear Regression for Data Classification, 2018 Int. Conf. Comput. Control. Electr. Electron. Eng. ICCCEEE 2018, no. August, pp. 1–5, doi: 10.1109/ICCCEEE.2018.8515759.

25. Bevans R. Simple Linear Regression: An Easy Introduction & Examples. 2020. URL: https://www.scribbr.com/statistics/multiple-linear-regression/

26. Breiman L., Friedman J., Charles J. Stone, OlshenTaylor & Francis R.A. Classification and Regression Trees. 1984. P. 368.

27. Bylander T. Estimating generalization error on twoclass datasets using out-of-bag estimates. Machine Learning, 2002. 48, 18, pp. 287–297.

28. Cavus, N. & Chingoka, D., N., C. (2015). Information technology in the banking sector: Review of mobile banking. Global Journal of Information Technology, 5(2), pp. 62-70.

29. Crook J., Banasik J. Forecasting and explaining aggregate consumer credit delinquency behaviour, Int. J. Forecasting 28 (1) (2012) 145–160.

30. Downey A., Elkner J., Meyers Ch. How to Think Like a Computer Scientist: Learning with Python. - Wellesley, Massachusetts: Green Tea Press, 2002. 290 pp.

31. Elektronnyi zhurnal «Bloomberg». Zamina robotamy spivrobitnykiv u Shvetsii [Bloomberg ezine. Replacement of employees in Sweden] (n.d.). Retrieved 19.01.2021 https://www.finanz.ru/novosti/aktsii/krupneyshiy-bank-shvecii-reshil-zamenit-robotami-6-tysyach- sotrudnikov-1027409738.

32. Fast Blockchain as the core of new banking technology [Fast Blockchain as the core of new banking technology] (n.d.). Retrieved 16.01.2021 https://guland.com.ua/kryptovalyutablockchain blokcheyn-i-banky.htm.

33. Hertzmann A., Fleet D. J., and Brubaker M. Linear Regression. 2015. URL: http://www.cs.toronto.edu/~mbrubake/teaching/C11/Handouts/LinearRegression.pdf

34. How Five Robots Replaced Seven Employees at a Swiss Bank. (n.d.). Retrieved 20.01.2021 from https://www.bloomberg.eom/news/articles/2018-05-04/how-five-robots-replaced-seven-employees-at-a- swiss-bank.

35. Iak banky vykorystovuiut AI ta Big Data dlia stvorennia novykh servisiv [How banks use AI and Big Data to create new services] (n.d.). Retrieved 20.01.2021 https://www.everest.ua/yak-banky- vykorystovuyut-ai-ta-big-data-dlya-stvorennya-novyh-servisiv.

36. Jordan M. Constrained supervised learning. Journal of Mathematical Psychology. 1992. 36. P. 396–452.

37. Mandic D.P., Chambers J.A. Recurrent Neural Networks for Prediction. Chichester: John Wiley&Sons, 2001. 285 p.

38. Moeedlodhi. How Outliers Can Pose a Problem in Linear Regression. URL: https://medium.com/swlh/how-outlierscan-pose-a-problem-in-linear-regression-1431c50a8e0

39. Molnar C. Interpretable machine learning. A Guide for Making Black Box Models Explainable,2019. URL: https://christophm.github.io/interpretable-ml-book/.

40. Rao C., Govindaraju V. Handbook of Statistics: Machine Learning: Theory and Applications, 2013. 552 c.

41. Tibshirani S. and Friedman H. Random Forest Regression model explained in depth, GDCoder, Jun. 04, 2019. https://gdcoder.com/random-forest-regressor-explained-in-depth/ (accessed Nov. 30, 2019). Valerie and Patrick Hastie, p. 764.

42. Tsanas A. and Xifara A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy Build*. vol. 49, pp. 560–567, Jun. 2012, doi: 10.1016/j.enbuild.2012.03.003.

43. Tutorialspoint. Python [Електронний ресурс]. Режим доступу: http://www.tutorialspoint.com/python/

44. Williams R.J., D. A. Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation. 1989. 1. P. 270–280.

45. Yu Z., Haghighat F., Fung B. C. M., and Yoshino H., A decision tree method for building energy demand modeling. *Energy Build*, vol. 42, no. 10, pp. 1637–1646, Oct. 2010, doi: 10.1016/j.enbuild.2010.04.006.

46. Zenzerović R. Credit scoring models in estimating the creditworthiness of small and medium and big enterprises. Croatian Operational Research Review, Vol. 2 No. 1, 2011.