МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В. Н. Каразіна

Факультет математики і інформатики

Кафедра комп'ютерних наук

Пояснювальна записка
до дипломної роботи
магістра

(освітній ступінь)

на тему «Машинне навчання в управлінні та запобіганні фінансовим ризикам»

Виконав: студент 2 курсу групи
МФ-61

Спеціальність
122 «Комп'ютерні науки»

Федосов В. А.

(прізвище й ініціали студента)

Керівник: Власенко Д. І.

(прізвище й ініціали)

Рецензент:

(прізвище й ініціали)

Харків – 2024

Міністерство освіти і науки України

Харківський національний університет імені В. Н. Каразіна

Факультет математики та інформатики
Кафедра інформатики
Рівень вищої освіти другий (магістр)
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____
(підпис) (ініціали та прізвище)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Федосов Вадим Андрійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Машинне навчання в управлінні та запобіганні фінансовим ризикам
керівник кваліфікаційної роботи Власенко Дмитро Іванович старший викладач, кандидат фізико-математичних наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом Університету № від «__» ____2024 року
2. Термін подання студентом роботи _____
3. Вихідні дані до проекту (роботи) Дані про позики за період з 2007 по 4 квартал 2018 року
4. Зміст пояснювальної записки (перелік завдань, які потрібно розв'язати)
Аналіз предметної області дослідження та даних, вибір доцільних методів та моделей для роботи, моделювання прогнозу та класифікації обраними методами, оцінка отриманих результатів та вибір найкращої моделі

5. Перелік графічного матеріалу аналіз даних, побудова моделей ,порівняння результатів

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

Нормоконтроль _____ _____ «___» червня 2024 р.
(підпис) (ініціали та прізвище)

7. Дата видачі завдання «____» _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Формування теми	30.10.2023	Виконано
2.	Вивчення літератури за темою	01.11.2023 - 29.12.2023	Виконано
3.	Підготовка першого розділу	08.01.2024 - 31.01.2024	Виконано
4.	Підготовка другого розділу	01.02.2024 - 29.02.2024	Виконано
5.	Розробка програмного продукту	01.03.2024 - 19.03.2024	Виконано
6.	Підготовка третього розділу	20.03.2024 - 15.04.2024	Виконано
7.	Оформлення розділів	15.04.2024 - 25.04.2024	Виконано

8.	Підготовка презентації	26.04.2024	Виконано
9.	Оформлення дипломної роботи	29.04.2024 - 30.04.2024	Виконано

Студент _____ **Федосов В.А.**

(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ **Власенко Д.І.**

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Дипломна робота: 131 с., 62 рис., 17 джерел.

КРЕДИТУВАННЯ, АНАЛІЗ ДАНИХ, МОДЕЛІ МАШИННОГО НАВЧАННЯ, АНАЛІЗ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ.

Об’єкт дослідження - позики взяті за період з 2007 по 4 квартал 2018 року.

Предмет дослідження - обробка даних і підготовка даних до моделювання. Навчання моделей машинного навчання на основі підготовлених даних.

Мета роботи - побудова моделі машинного навчання, здатної передбачити ймовірність непогашення кредиту.

Розглянуто здатність моделей машинного навчання прогнозувати успішність кредитування. Проведено ретельний аналіз і підготовку кредитних даних, щоб зрозуміти ключові показники успіху чи невдачі кредитування. Маючи чисті та структуровані дані, у роботі досліджено створення моделей машинного навчання для прогнозування результатів кредитів. Оцінюються різні алгоритми, включаючи логістичну регресію, дерева рішень і ансамблеві методи, щоб визначити найбільш ефективний підхід для прогнозування успіху кредитування. Результатом є набір моделей машинного навчання, створених для прогнозування успіху кредитування з високим ступенем точності. Серед цих моделей обрана найефективніша на основі її показників ефективності та придатності для практичного застосування в сценаріях кредитування.

ABSTRACT

Diploma work: 131 pages, 62 figures, 17 sources.

LENDING, DATA ANALYSIS, MACHINE LEARNING MODELS, ANALYSIS OF MODELING RESULTS.

The object of the study is loans taken for the period from 2007 to the 4th quarter of 2018.

The subject of research is data processing and data preparation for modeling. Training of machine learning models based on prepared data.

The purpose of the work is to build a machine learning model capable of predicting the probability of loan default.

The ability of machine learning models to predict lending success is considered. Thorough analysis and preparation of credit data to understand key indicators of lending success or failure. Given clean and structured data, the paper explores the creation of machine learning models for predicting loan outcomes. Various algorithms, including logistic regression, decision trees, and ensemble methods, are evaluated to determine the most effective approach for predicting lending success. The result is a set of machine learning models built to predict lending success with a high degree of accuracy. Among these models, the most effective one is chosen based on its performance indicators and suitability for practical application in lending scenarios.

ЗМІСТ

ВСТУП

Сучасні фінансові установи стикаються зі значними проблемами в управлінні фінансовими ризиками та їх зменшенні через складний і динамічний характер глобального економічного ландшафту. Експоненціальне зростання фінансових даних зумовило необхідність впровадження передових технологій для отримання значущої інформації та прийняття обґрунтованих рішень. У цій дипломній роботі розглядається перетин машинного навчання (ML) і управління фінансовими ризиками (FRM), щоб вирішити ці проблеми та покращити прогноз потенційних ризиків.

Розділ 1 містить вичерпний огляд управління фінансовими ризиками, наголошуючи на важливій ролі, яку воно відіграє в стабільності та прибутковості фінансових установ. Основна увага буде зосереджена на розумінні проблем, пов’язаних з фінансовими ризиками, і дослідженні того, як методи машинного навчання можуть бути інструментальними для ефективного управління та запобігання цим ризикам.

Розділ 2 містить поглиблений аналіз різних методів і алгоритмів ML, які можна застосувати до управління фінансовими ризиками. Цей розділ має на меті детально вивчити сильні та слабкі сторони різних методів ML, продемонструвавши їх придатність для вирішення конкретних типів фінансових ризиків. Розуміючи нюанси цих алгоритмів, фінансові професіонали можуть приймати обґрунтовані рішення щодо їх застосування в практиці управління ризиками.

Третій розділ присвячений практичному застосуванню моделей ML у прогнозуванні фінансових ризиків. Це передбачає поглиблений аналіз функціонування та оцінку якості моделей прогнозування фінансових ризиків, розроблених за допомогою машинного навчання. Критично оцінюючи продуктивність і надійність цих моделей, аналіз прагне дати

розуміння ефективності ML у прогнозуванні та запобіганні фінансовим ризикам, що зрештою сприяє загальному вдосконаленню стратегій управління ризиками.

Таким чином, ця робота присвячена вивченню синергії між машинним навчанням і управлінням фінансовими ризиками, щоб забезпечити повне розуміння того, як передові методи, керовані даними, можуть революціонізувати процеси оцінки ризиків і прийняття рішень у фінансовому секторі. Завдяки дослідженню поточного ландшафту, аналізу методів ML та оцінці моделей прогнозування це дослідження робить внесок у поточний дискурс щодо підвищення фінансової стабільності та стійкості перед лицем ризиків, що розвиваються.

РОЗДІЛ 1. ОГЛЯД УПРАВЛІННЯ ТА ЗАПОБІГАННЯ ФІНАНСОВИМ РИЗИКАМ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ

1.1 Поняття управління фінансовими ризиками

Управління фінансовими ризиками є критично важливим компонентом сучасних фінансових установ, спрямованим на виявлення, оцінку та зменшення потенційних ризиків, які можуть негативно вплинути на їх фінансовий стан. Це передбачає стратегічне використання різноманітних інструментів, технік і методологій для мінімізації впливу невизначеності на фінансових ринках.

Основні поняття та терміни управління фінансовими ризиками:

- *Ризик* - у фінансовому плані ризик означає невизначеність, пов’язану з потенційними фінансовими втратами. Воно може виникати з різних джерел, включаючи коливання ринку, неповернення кредитів, операційні збої та економічні зміни.

- *Фінансовий ризик* — це ризик потенційних фінансових втрат, які можуть виникнути через нестабільність ринку, кредитні дефолти, зміни процентних ставок, коливання валют та інші невизначеності у фінансовому ландшафті.

- *Управління фінансовими ризиками* - передбачає ідентифікацію, оцінку та контроль потенційних фінансових ризиків. Мета полягає в мінімізації впливу негативних подій на прибутки та капітал фінансової установи.

Види фінансових ризиків:

- *Ринковий ризик* - ризик збитків через зміни ринкових цін, наприклад, коливання процентних ставок, обмінних курсів і цін на товари.

- *Кредитний ризик* - ризик збитків, що виникає внаслідок невиконання позичальником або контрагентом своїх фінансових зобов'язань.

- *Операційний ризик* - ризик збитків внаслідок неадекватних внутрішніх процесів, систем, людей або зовнішніх подій.

- *Ризик ліквідності* - ризик неспроможності виконати короткострокові фінансові зобов'язання через брак ринкової ліквідності.

- *Оцінка ризику* - оцінка потенційного впливу та ймовірності виникнення ризиків. Це передбачає кількісний та якісний аналіз, щоб зрозуміти природу та величину різних ризиків.

- *Зменшення ризику* - розробка стратегій і політики для мінімізації впливу виявлених ризиків. Це може включати диверсифікацію, хеджування та використання фінансових інструментів для передачі ризику.

Традиційне управління фінансовими ризиками:

Фінансові установи традиційно використовували звичайні методи управління ризиками, які часто включали ручні процеси та значною мірою покладалися на історичні дані та статистичні моделі. Незважаючи на цінність цих методів, вони мали обмеження щодо пристосованості до умов ринку, що швидко змінювалися.

Основні аспекти традиційного управління фінансовими ризиками включають:

- *Аналіз історичних даних* - традиційне управління ризиками значною мірою покладається на аналіз історичних даних для виявлення закономірностей і тенденцій. Хоча це дає цінну інформацію, воно може недостатньо відобразити динаміку ринків, що розвиваються, особливо під час безпрецедентних подій.

- *Статистичні моделі* - Статистичні моделі, такі як Value at Risk (VaR), широко використовуються для кількісного визначення та вимірювання різних ризиків. Однак ці моделі часто припускають статичні

ринкові умови і можуть не повністю врахувати хвостові ризики або екстремальні події.

- *Відповідність нормативним вимогам* - Фінансові установи пов'язані нормативними вимогами, які диктують мінімальні стандарти практики управління ризиками. Відповідність часто передбачає застосування стандартизованих підходів, які можуть не пристосовуватися до конкретного профілю ризику окремих установ.

- *Ручні процеси* - Багато процесів управління ризиками були ручними та займали багато часу, вимагаючи значного втручання людини. Це може призвести до затримок у прийнятті рішень і може бути недостатнім для вирішення ризиків у реальному часі.

Проблеми традиційного управління фінансовими ризиками:

- *Обмеження моделі* - традиційним моделям може бути важко охопити складні, нелінійні відносини на фінансових ринках, особливо під час швидких змін або криз.

- *Якість і своєчасність даних* - точність і своєчасність даних мають вирішальне значення в управлінні ризиками. Традиційні системи можуть зіткнутися з проблемами в обробці та аналізі величезних наборів даних у режимі реального часу, що потенційно може призвести до затримки або неточності оцінки ризиків.

- *Інтеграція кількох ризиків* - фінансові установи стикаються з багатьма ризиками одночасно. Традиційні методи часто розглядали ризики ізольовано, не звертаючи уваги на взаємозв'язок ринкових, кредитних, операційних ризиків і ризиків ліквідності.

- *Поява нових ризиків* - фінансовий ландшафт постійно змінюється, породжуючи нові типи ризиків, наприклад загрози кібербезпеці та системні ризики. Традиційні методи управління ризиками можуть мати проблеми з ефективним вирішенням цих нових проблем.

1.2 Статистичні моделі

Статистичні моделі відіграють вирішальну роль в управлінні фінансовими ризиками, надаючи кількісні інструменти для аналізу та управління різними типами ризиків, з якими стикаються фінансові установи та інвестори. Статистичні моделі допомагають ідентифікувати та кількісно оцінити різні типи фінансових ризиків, таких як ринковий ризик, кредитний ризик, операційний ризик і ризик ліквідності. Ці моделі аналізують історичні дані, щоб визначити закономірності та тенденції, які можуть вказувати на потенційні ризики. Статистичні моделі широко використовуються для оцінки ринкового ризику, який включає потенційні втрати через зміни ринкових цін, процентних ставок, обмінних курсів та інших відповідних факторів. Для оцінки максимального потенційного збитку в межах певного довірчого інтервалу зазвичай використовуються моделі Value at Risk (VaR). Статистичні моделі допомагають аналізувати операційний ризик, який виникає через внутрішні процеси, системи, людські помилки та зовнішні події. Моделі кредитного ризику оцінюють ймовірність дефолту позичальників і кількісно визначають потенційні збитки, пов'язані з кредитним ризиком. Моделі кредитного рейтингу та моделі ймовірності дефолту (PD) використовуються для оцінки кредитоспроможності окремих осіб, компаній або фінансових інструментів. Таким чином, статистичні моделі є важливими інструментами в управлінні фінансовими ризиками, надаючи кількісну інформацію, яка дозволяє установам приймати обґрунтовані рішення, оптимізувати портфелі та проактивно керувати широким спектром фінансових ризиків.

1.2.1 VaR, CVaR, RVaR, EVaR моделі

Ризикова вартість (VaR) служить показником для кількісної оцінки потенційної втрати інвестицій або капіталу шляхом оцінки максимальної суми, яку портфель може втратити протягом визначеного періоду часу, враховуючи нормальні ринкові умови та заздалегідь визначену ймовірність. Ця ймовірність, позначена як p, представляє ймовірність падіння вартості портфеля нижче певного порогу.

Для визначеного портфеля, часового горизонту та ймовірності p, p VaR характеризується як найвища можлива втрата протягом цього періоду, за винятком результатів, сукупна ймовірність яких перевищує p. Цей розрахунок передбачає ціноутворення за ринковою ціною та відсутність торгової діяльності в портфелі (рис. 1.1).

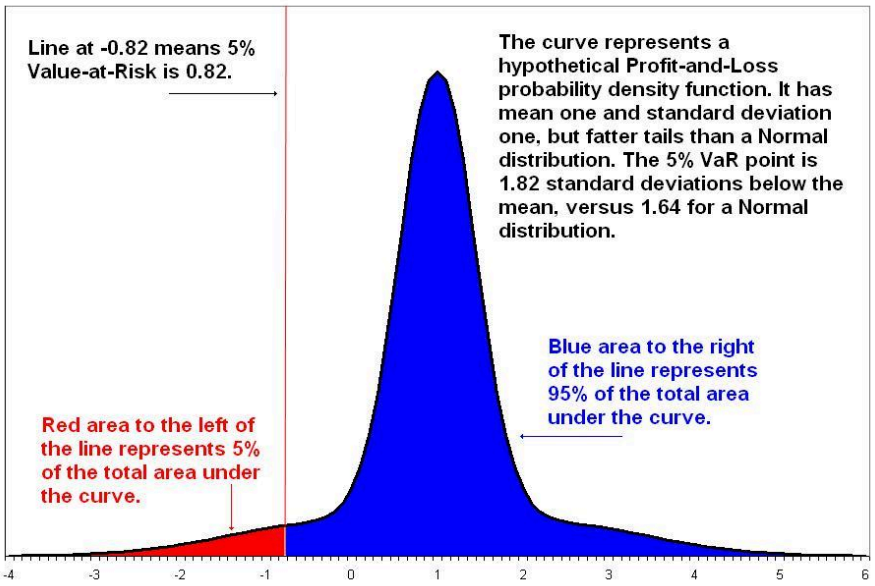

Рисунок 1.1 - Ризикова вартість у 5% гіпотетичної функції щільності ймовірності прибутків і збитків.[1]

95% VaR у розмірі 1 мільйона доларів США означає, що існує 5% ймовірність втрати понад 1 мільйона доларів США протягом зазначеного періоду часу. VaR допомагає ризик-менеджерам та інвесторам оцінити ризик зниження своїх портфелів і прийняти обґрунтовані рішення щодо ризику.

Математичне визначення VaR представляє максимальну потенційну втрату в межах даного рівня довіри протягом певного часового горизонту.

Математично VaR визначається так:

$$VaR_{\alpha}(X) = -\inf\{x \in R: F_X(x) > \alpha\} = F_Y^{-1}(1 - \alpha) \qquad (1.1)$$

Де X — вартість або збиток портфеля, α — рівень довіри (наприклад, 95% або 99%), а P(X ≤ x) — кумулятивна функція розподілу вартості або збитку портфеля.

Історичне моделювання VaR можна розрахувати, використовуючи історичні дані для моделювання можливих майбутніх сценаріїв. Збитки ранжуються, а VaR визначається на основі історичного процентиля. Параметричні моделі VaR можна обчислити за допомогою статистичних методів, припускаючи певний розподіл прибутковості портфеля і оцінюючи параметри. Моделювання за методом Монте-Карло передбачає створення кількох випадкових сценаріїв для моделювання майбутньої вартості портфеля, що дозволяє більш гнучко моделювати складні відносини.

CVaR, також відомий як очікуваний дефіцит (ES), виходить за межі VaR, вимірюючи очікувані збитки за умови, що збитки перевищують порогове значення VaR. Він забезпечує більш повне уявлення про екстремальні втрати.

95% CVaR у 1,5 мільйона доларів США означає, що в найгірших 5% сценаріїв середній збиток становитиме 1,5 мільйона доларів США.

CVaR корисний, коли екстремальні події викликають особливе занепокоєння, оскільки він фіксує інформацію про серйозність збитків за межами рівня VaR.

Математичне визначення CVaR представляє очікуване значення втрат, що перевищують порогове значення VaR.

Математично CVaR визначається так:

$$CVaR_{\alpha}(X) = \frac{1}{1-\alpha}\int_{\alpha}^{1} VaR_{p}(X)dp \quad (1.2)$$

CVaR надає додаткову інформацію про хвостовий ризик, окрім VaR, і вважається більш узгодженим з перевагами ризику.

Математичне інтегрування CVaR можна розрахувати шляхом інтегрування краю розподілу втрат за межами рівня VaR. Методи моделювання подібно до VaR, CVaR можна обчислити за допомогою історичного моделювання або моделювання Монте-Карло, зосереджуючись на краю розподілу.

RVaR — це показник ризику, який виражає VaR як відсоток від загальної вартості портфеля. Це допомагає стандартизувати показники ризику для портфелів різного розміру. Якщо VaR портфеля становить 2 мільйони доларів, а загальна вартість портфеля становить 100 мільйонів доларів, то RVaR становить 2%. RVaR корисний для порівняння ризикованості різних портфелів і розуміння пропорційного впливу потенційних втрат.

Математичне визначення: RVaR вимірює ризик відносно розміру портфеля.

Математично RVaR визначається так:

$$RVaR_{\alpha}(X) = \frac{VaR_{\alpha}(X)}{Portfolio\ Value} \quad (1.3)$$

RVaR корисний для порівняння ризикованості різних портфелів у нормалізованій шкалі.

Щоб отримати відносну міру RVaR треба розділити окремо обчислені VaR однієї інвестиції на VaR іншої.

EVaR — це міра ризику, яка поєднує очікуваний дохід і VaR портфеля. Він забезпечує більш цілісне уявлення, враховуючи як потенційні прибутки, так і втрати. Якщо портфель має очікувану прибутковість 8% і VaR 5%, EVaR становить 3%. EVaR корисний для інвесторів, які хочуть збалансувати ризик і прибутковість, оскільки він враховує як середню прибутковість, так і потенційний ризик зниження.

Математичне визначення EVaR — це гібридна метрика, яка поєднує в собі очікувану вартість і міркування щодо ризику.

Математично EVaR визначається так:

$$EVaR_{\alpha}(X) = E\left[X | X \leq VaR_{\alpha}(X)\right] \quad (1.4)$$

Він представляє очікувану вартість портфеля за умови, що він знаходиться в крайній області, охопленій VaR.

Інтеграція EVaR з VaR – це, добуток ймовірності події (нижчого за VaR) і очікуваних збитків у зв’язку з цією подією.

В контексті стандартного нормального розподілу (рис. 1.2).

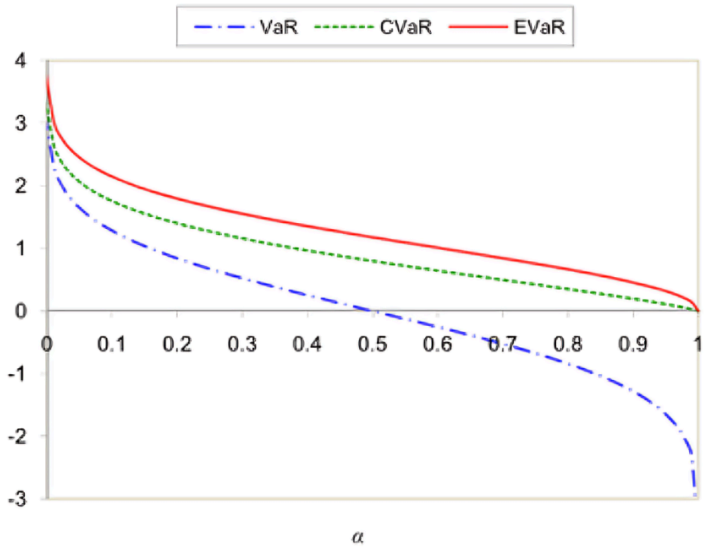

Рисунок 1.2 - Порівняння VaR, CVaR і EVaR для стандартного нормального розподілу.

Стандартний нормальний розподіл, позначений Z, має середнє значення (μ) 0 і стандартне відхилення (σ) 1:

$$Z \sim N(0,1) \tag{1.5}$$

VaR представляє максимальну потенційну втрату в межах заданого рівня довіри протягом певного періоду часу. Для стандартного нормального розподілу VaR на рівні довірчої ймовірності α можна розрахувати за допомогою зворотної інтегральної функції розподілу стандартного нормального розподілу:

$$VaR_{\alpha} = -\ \sigma \times Z_{\alpha} \tag{1.6}$$

Де Zα — (1−α) процентиль стандартного нормального розподілу.

CVaR, також відомий як очікуваний дефіцит (ES) або очікувана вартість під ризиком (EVaR), виходить за межі VaR, враховуючи середній збиток понад порогове значення VaR. Математично CVaR можна визначити як:

$$CVaR_{\alpha} = \frac{1}{\alpha} \int_{-\infty}^{-VaR_{\alpha}} x \times f(x)dx \tag{1.7}$$

де f(x) – функція щільності ймовірності стандартного нормального розподілу.

Інша поширена формула для очікуваного дефіциту (ES) або очікуваної вартості під ризиком (EVaR):

$$CVaR_{\alpha} = \mu + \frac{1}{\alpha} \int_{-\infty}^{-VaR_{\alpha}} x \times f(x)dx \tag{1.8}$$

Це середнє значення всіх втрат, що перевищують VaR на заданому рівні довіри.

Ці формули забезпечують основу для розрахунку показників ризику для портфеля або інвестицій зі стандартним нормальним розподілом. На практиці можуть бути розглянуті інші розподіли, і чисельні методи можуть бути використані для інтеграції, яка бере участь у розрахунках CVaR.

Розглянемо рівномірний розподіл на інтервалі (0,1), позначеному U(0,1) (рис. 1.3).

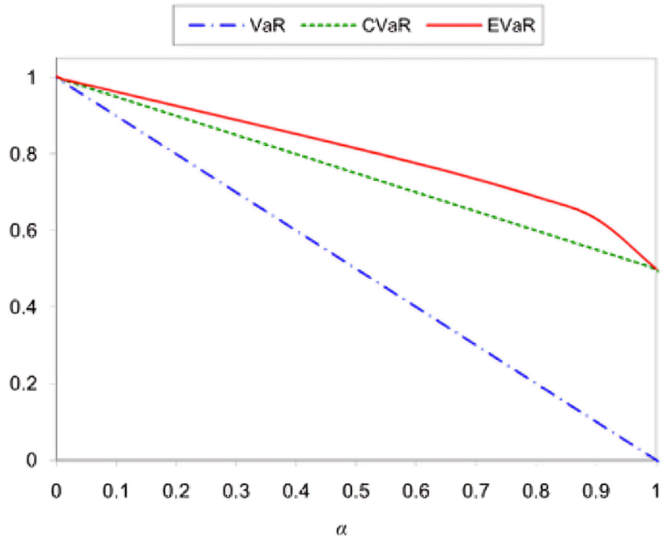

Рисунок 1.2 - Порівняння VaR, CVaR та EVaR для рівномірного розподілу в інтервалі (0,1).

Функція щільності ймовірності рівномірного розподілу на інтервалі (0,1) постійна в цьому інтервалі та дорівнює нулю в інших місцях:

$$f(x) = 1 \; for \; 0 < x < 1 \tag{1.9}$$

Значення ризику (VaR) на рівні довіри α для рівномірного розподілу можна розрахувати шляхом знаходження α-квантиля, який відповідає $F^{-1}(\alpha)$, де F(x) є кумулятивною функцією розподілу. Для рівномірного розподілу за (0,1) визначається як:

$$F(x) = x \; for \; 0 < x < 1 \tag{1.10}$$

Отже, VaR на рівні довірчої ймовірності α становить:

$$VaR_{\alpha} = F^{-1}(\alpha) = \alpha \quad (1.11)$$

Умовне значення ризику (CVaR) або очікуваний дефіцит (ES) для рівномірного розподілу можна обчислити як середнє значення втрат, що перевищують порогове значення VaR. Для рівномірного розподілу за (0,1) CVaR визначається як:

$$CVaR_{\alpha} = \frac{1}{\alpha}\int_{\alpha}^{1} x\,dx \quad (1.12)$$

Оцінка цього інтеграла дає:

$$CVaR_{\alpha} = \frac{1}{2\alpha}(1 - \alpha)^2 \quad (1.13)$$

Очікувана вартість під ризиком (EVaR) іноді використовується як взаємозамінна з CVaR. У разі рівномірного розподілу зазвичай розглядається CVaR.

Для рівномірного розподілу на інтервалі (0,1) це формули для VaR і CVaR. Зверніть увагу, що у випадку рівномірного розподілу обчислення відносно прості порівняно зі складнішими розподілами, оскільки функція щільності ймовірності і кумулятивна функція розподілу мають прості форми.

Такі моделі вимірювання ризику, як Value at Risk (VaR), Conditional Value at Risk (CVaR), Relative Value at Risk (RVaR) і Expected Value at Risk (EVaR), дають цінну інформацію про ризики портфеля. Однак вони не позбавлені обмежень. Припущення щодо нормальності VaR можуть бути неточними під час екстремальних подій на ринку. Відсутність хвостової інформації VaR може призвести до недооцінки втрат за межі зазначеного рівня довіри. Відсутність адитивності у VaR може призвести до викривлення загального ризику портфеля. Так само CVaR чутливий до

припущень щодо розподілу, і його обчислювальна складність може бути високою для складних портфелів із нелінійними зв’язками між активами.

Залежність RVaR від вибраного контрольного показника вносить варіативність в оцінку ризику, що робить критично важливим розглянути вибір контрольного показника. Крім того, інтерпретація RVaR може викликати труднощі для неспеціалістів, і метрика може не охоплювати всі аспекти ризику. З іншого боку, розрахунок EVaR є більш складним, ніж традиційні вимірювання ризику, і вимагає складних методів моделювання. Суб’єктивність в оцінці параметрів для EVaR, як і в інших моделях, може вплинути на результати, засновані на суб’єктивному виборі.

Загалом, ці моделі вимірювання ризику пропонують цінну інформацію, але критика зосереджена навколо припущень, обчислювальної складності та суб’єктивної оцінки параметрів. Керівники ризиків повинні ретельно розглянути ці фактори та потенційно використовувати комбінацію моделей для отримання більш надійної оцінки ризику.

Ці показники ризику є ключовими інструментами для управління ризиками, що дозволяє інвесторам та установам оцінювати, кількісно визначати та керувати потенційними фінансовими втратами, пов’язаними з їхніми портфелями. Важливо зазначити, що ці моделі мають свої обмеження та припущення, і їх слід використовувати в поєднанні з іншими методами управління ризиками для комплексного підходу.

1.2.2 GARCH моделі

GARCH (узагальнена авторегресійна умовна гетероскедастичність) — це статистична модель, яка використовується для аналізу та прогнозування даних часових рядів зі зміною дисперсії з часом. Моделі GARCH припускають, що дисперсія помилки в часовому ряді відповідає

процесу авторегресії. Моделі GARCH в основному широко використовуються у фінансовій економетриці для моделювання кластеризації волатильності та умовної гетероскедастичності прибутковості активів.

Математично GARCH визначається так:

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \quad (1.14)$$

Де σ_t^2 — умовна дисперсія в момент часу t, ε_t — стандартизований залишок у момент часу t, ω — постійний терм, α_i — параметри авторегресії, β_j — коефіцієнти для минулих відхилень, а p і q — параметри порядку.

Критика GARCH:

- Припускає, що волатильність є єдиним фактором, що впливає на майбутню волатильність.
- Може не фіксувати раптові та великі зміни волатильності, відомі як кластеризація волатильності.

NGARCH розширює модель GARCH, враховуючи нелінійність у процесі умовної дисперсії. Він може вловлювати асиметрію та використовувати ефекти. NGARCH корисний у ситуаціях, коли реакція волатильності на минулі шоки не є симетричною, і існує потреба моделювати нелінійності в процесі волатильності.

Математично NGARCH визначається так:

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i f(\varepsilon_{t-i})^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \quad (1.15)$$

Де $f(\varepsilon_{t-i})$ - нелінійна функція попередніх квадратичних помилок прогнозу.

Критика NGARCH:

- Підвищена складність може призвести до переобладнання.

- Вибір нелінійної функції вводить додаткові параметри, які необхідно оцінити.

IGARCH є розширенням моделі GARCH, яка включає параметр порядку інтеграції. Це корисно, коли дані часових рядів є нестаціонарними та потребують різниці для досягнення стаціонарності. IGARCH використовується при роботі з нестаціонарними фінансовими даними часового ряду для моделювання та прогнозування волатильності.

Математично IGARCH визначається так:

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 + \gamma \sigma_{t-1}^2 \quad (1.16)$$

Де γ - параметр інтеграції.

Критика IGARCH:

- Оцінка може бути обчислювально інтенсивною.

- Інтерпретація параметра довгої пам'яті може бути складною.

EGARCH моделює умовну дисперсію як функцію минулих квадратичних інновацій і минулих шоків волатильності. Це допускає асиметричний вплив на волатильність. EGARCH особливо корисний для визначення ефекту кредитного плеча, коли негативні шоки мають інший вплив на волатильність порівняно з позитивними шоками.

Математично EGARCH визначається так:

$$log(\sigma_t^2) = \omega + \sum_{i=1}^{p} \alpha_i \left(\frac{\varepsilon_{t-i}}{\sigma_{t-i}}\right) + \sum_{j=1}^{q} \beta_j log(\sigma_{t-j}^2) \quad (1.17)$$

Де α_i - коефіцієнти для стандартизованих минулих помилок прогнозу.

Критика EGARCH:

- Інтерпретація параметрів може бути складною.

- Може бути чутливим до викидів.

GARCH-M розширює модель GARCH до багатовимірних часових рядів, дозволяючи моделювати матрицю умовної коваріації між кількома активами. GARCH-M зазвичай використовується в управлінні портфелем і аналізі ризиків, де кореляції та коваріації між різними активами важливі для оцінки ризику.

Модель GARCH-M додає до середнього рівняння член гетероскедастичності. Він має специфікацію:

$$y_t = \beta x_t + \lambda \sigma_t + \epsilon_t \tag{1.18}$$

Залишок ϵ_t визначається як:

$$\epsilon_t = \sigma_t \times z_t \tag{1.19}$$

Критика GARCH-M:

- Складність зростає з включенням середньої динаміки.
- Може не фіксувати нелінійні зв’язки між середнім значенням і волатильністю.

Кожна зі згаданих моделей, включаючи GARCH та її варіації, такі як NGARCH, IGARCH, EGARCH, GARCH-M, належить до категорії моделей ARCH (авторегресивна умовна гетероскедастичність) і GARCH (узагальнена авторегресійна умовна гетероскедастичність). Ці моделі широко використовуються у фінансовій економетриці для моделювання та прогнозування нестабільності фінансових часових рядів.

Підводячи підсумок, можна сказати, що ці моделі пропонують різні розширення та вдосконалення базової структури GARCH для вирішення різних характеристик і закономірностей, що спостерігаються в даних часових рядів, особливо в контексті фінансової економетрики. Вибір конкретної моделі залежить від характеристик даних і характеристик, які потрібно охопити.

1.3 Підходи до управління фінансовими ризиками в машинному навчанні

Управління фінансовими ризиками є критично важливим аспектом сучасного фінансового ландшафту, де невизначеність і ринкові коливання створюють значні проблеми. Інтеграція методів машинного навчання зробила революцію в традиційних практиках управління ризиками, запропонувавши передові інструменти для прогнозування, аналізу та запобігання. У цьому розділі ми розглянемо різні підходи до управління фінансовими ризиками з акцентом на застосування методів машинного навчання.

Ідентифікація та класифікація ризиків:

Машинне навчання чудово розпізнає закономірності та аномалії у великих наборах даних. В управлінні фінансовими ризиками ця можливість використовується для визначення та класифікації різних типів ризиків, таких як кредитний ризик, ринковий ризик, операційний ризик і ризик ліквідності. Завдяки аналізу історичних даних моделі машинного навчання можуть навчитися розрізняти нормальну поведінку ринку та потенційні ризикові події.

Кредитний рейтинг і прогнозування невиконання зобов'язань:

Однією з основних проблем фінансових установ є кредитний ризик. Алгоритми машинного навчання відіграють вирішальну роль в оцінці кредитоспроможності фізичних осіб і компаній шляхом аналізу різноманітних джерел даних. Ці алгоритми враховують такі змінні, як кредитна історія, дохід та інші фінансові показники, щоб передбачити ймовірність дефолту, допомагаючи в більш точному прийнятті рішень щодо кредитування та схвалення кредиту.

Управління ринковими ризиками:

Моделі машинного навчання можуть аналізувати ринкові тенденції та прогнозувати потенційні ризики, пов’язані зі змінами цін на активи, процентних ставок і коливань валют. Використовуючи такі алгоритми, як аналіз часових рядів і нейронні мережі, фінансові установи можуть покращити свою здатність прогнозувати зміни ринку та відповідно коригувати свої портфелі для пом’якшення ринкових ризиків.

Виявлення та запобігання шахрайству:

Фінансові установи постійно стикаються з загрозами шахрайства. Завдяки своїй здатності виявляти аномалії та незвичайні шаблони машинне навчання значно сприяє виявленню та запобіганню шахрайству. Аналізуючи дані транзакцій у режимі реального часу, ці моделі можуть ідентифікувати підозрілу діяльність, потенційно запобігаючи фінансовим втратам і захищаючи цілісність фінансової системи.

Зменшення операційного ризику:

Машинне навчання допомагає визначати та зменшувати операційні ризики, що виникають через внутрішні процеси, системи або людські помилки. Завдяки аналізу історичних інцидентів і моніторингу в реальному часі ці моделі можуть надати інформацію про потенційні вразливості та рекомендувати профілактичні заходи для підвищення операційної стійкості.

Стрес-тестування та аналіз сценаріїв:

Алгоритми машинного навчання полегшують стрес-тестування шляхом імітації різних сценаріїв і оцінки впливу на фінансові портфелі. Цей підхід допомагає фінансовим установам оцінити свою стійкість перед лицем несприятливих умов, дозволяючи їм приймати обґрунтовані рішення, щоб протистояти економічним спадам або іншим непередбаченим подіям.

Управління ризиком ліквідності:

Управління ризиком ліквідності має вирішальне значення для фінансових установ, щоб забезпечити достатні кошти для виконання своїх зобов’язань. Моделі машинного навчання можуть аналізувати моделі грошових потоків, прогнозувати потреби в ліквідності та оптимізувати стратегії фінансування. Цей проактивний підхід покращує управління ризиком ліквідності та зменшує ймовірність фінансової нестабільності.

Підходи до машинного навчання:

Прогнозна аналітика:

- Опис: моделі машинного навчання можуть аналізувати величезні обсяги історичних даних і даних у реальному часі, щоб передбачити майбутні рухи ринку та визначити потенційні ризики.
- Переваги: моделі ML можуть фіксувати нелінійні моделі та адаптуватися до мінливих ринкових умов, забезпечуючи точніші прогнози в певних ситуаціях.
- Обмеження: природа «чорної скриньки» деяких алгоритмів машинного навчання може ускладнити розуміння обґрунтування конкретних прогнозів, що викликає занепокоєння щодо інтерпретації моделі.

Кластеризація та класифікація:

- Опис: методи ML, такі як кластеризація та класифікація, можуть групувати подібні фінансові інструменти або ідентифікувати активи зі схожими профілями ризику.
- Переваги: ці методи покращують диверсифікацію портфеля та допомагають ідентифікувати активи з порівнянними характеристиками ризику.
- Обмеження: якість результатів залежить від вибору функцій і актуальності навчальних даних, тому вкрай важливо ретельно розробляти ці аспекти.

Виявлення аномалії:

- Опис: моделі ML можна навчити виявляти аномалії або незвичні шаблони у фінансових даних, які можуть вказувати на потенційні ризики.
- Переваги: ефективний у виявленні рідкісних подій або викидів, які можуть бути пропущені традиційними методами.
- Обмеження: можливі помилкові спрацьовування, і моделі потрібно постійно оновлювати для адаптації до мінливих ринкових умов.

Інтеграція традиційного та машинного навчання:

- *Гібридні моделі* - поєднання традиційних підходів і підходів машинного навчання може використовувати сильні сторони обох. Наприклад, використання машинного навчання для покращення прогнозної аналітики, покладаючись на традиційні моделі для відповідності нормативним вимогам.

- *Аналіз сценарію ризику* - інтеграція машинного навчання в аналіз сценаріїв допомагає симулювати широкий спектр потенційних майбутніх подій та їхній вплив на фінансовий ландшафт.

- *Безперервне навчання* - моделі машинного навчання можуть адаптуватися та вчитися на нових даних, забезпечуючи більш динамічний і оперативний підхід до управління ризиками порівняно зі статичними традиційними моделями.

Проблеми:

- *Якість даних і зміщення* - традиційний, і машинний підходи значною мірою покладаються на дані. Питання, пов’язані з якістю, повнотою та упередженістю даних, можуть вплинути на точність оцінки ризиків.

- *Відповідність нормативним вимогам* - фінансові установи повинні дотримуватися нормативних вказівок. Інтеграція моделей машинного

навчання в управління ризиками вимагає забезпечення відповідності нормативним стандартам і пояснення модельних рішень.

- *Можливість тлумачення* - розуміння та пояснення результатів моделей машинного навчання залишається проблемою. Щоб вирішити цю проблему, розробляються методи штучного інтелекту (AI).

- *Динамічні ринкові умови* - фінансові ринки динамічні і можуть зазнавати швидких змін. Щоб залишатися актуальними, моделі машинного навчання мають бути адаптованими та постійно оновлюватися.

Підсумовуючи, комплексна стратегія управління ризиками часто передбачає поєднання традиційного та машинного навчання. Традиційні методи забезпечують міцну основу, тоді як машинне навчання додає гнучкості та здатності обробляти величезні обсяги даних для більш точної оцінки ризиків. Ключ полягає в ефективній інтеграції цих підходів для створення надійної та адаптивної системи управління ризиками.

1.4 Сучасний стан управління фінансовими ризиками в машинному навчанні

Управління фінансовими ризиками все більше використовує машинне навчання (ML) і передову аналітику для покращення процесів прийняття рішень. Детальний огляд поточного стану управління фінансовими ризиками в машинному навчанні:

Оцінка кредитного ризику: Моделі машинного навчання зробили революцію в оцінці кредитного ризику, використовуючи різноманітні джерела даних, крім традиційних кредитних балів. Ці моделі включають не лише фінансову історію, але й нетрадиційні дані, такі як активність у соціальних мережах, поведінка в Інтернеті та навіть психографічна інформація. Цей цілісний підхід дозволяє більш детально оцінювати кредитоспроможність. Однак проблеми, пов’язані з інтерпретацією моделі та потенційними упередженнями в альтернативних джерелах даних,

потребують постійної уваги, щоб забезпечити справедливі та етичні практики кредитування.

Виявлення шахрайства: Використання машинного навчання для виявлення шахрайства розвинулося, щоб охопити більш складні методи. Аналіз даних у реальному часі, виявлення аномалій і адаптивні алгоритми покращують здатність виявляти та запобігати шахрайським діям. Однак перегони між шахраями та алгоритмами виявлення потребують постійного вдосконалення. Оскільки тактика шахрайства розвивається, моделі машинного навчання повинні швидко адаптуватися, щоб підтримувати свою ефективність.

Аналіз ринкового ризику: Методи машинного навчання в аналізі ринкових ризиків виходять за рамки традиційних методів, обробляючи великі обсяги даних для виявлення складних закономірностей і кореляцій. Завдання полягає в тому, щоб вирішити динамічну природу фінансових ринків і можливість непередбачуваних подій. Поточні дослідження зосереджені на покращенні надійності моделі та впровадженні пояснюваності для підвищення довіри до прогнозів, особливо в умовах екстремальних ринкових умов.

Управління операційними ризиками: Роль ML в управлінні операційним ризиком виходить за рамки виявлення закономірностей в історичних даних. Моделі прогнозованого технічного обслуговування на основі машинного навчання дозволяють проактивно виявляти потенційні збої в обладнанні, скорочуючи час простою та збої в роботі. Проблема полягає в тому, щоб інтегрувати ці моделі в існуючі операційні процеси та забезпечити їх надійність у різноманітних середовищах.

Алгоритмічна торгівля: Алгоритми машинного навчання стали невід’ємною частиною алгоритмічної торгівлі, оптимізуючи стратегії на основі історичних даних і ринкових умов у реальному часі. Навчання з підкріпленням набуло популярності, дозволяючи алгоритмам адаптуватися

до мінливої динаміки ринку. Проблеми включають потребу в надійних стратегіях управління ризиками для пом’якшення непередбачуваних наслідків і потенційних системних ризиків, пов’язаних з алгоритмічною торгівлею.

Відповідність нормативним вимогам: Машинне навчання полегшує автоматизацію процесів дотримання нормативних вимог у фінансовій галузі. Обробка природної мови (NLP) відіграє вирішальну роль в аналізі та розумінні нормативних текстів, допомагаючи в дотриманні відповідності. Проте залишаються проблеми із забезпеченням прозорості та можливості інтерпретації моделей ML, особливо в складних нормативних умовах, де підзвітність і можливість пояснення є найважливішими.

Кредитний рейтинг і затвердження кредиту: Вплив ML на оцінку кредитоспроможності поширюється на динамічні та персоналізовані оцінки, що включають ширший діапазон даних. Автоматизовані процеси схвалення кредитів, керовані машинним навчанням, підвищують швидкість і точність рішень про кредитування. Тим не менш, питання, пов’язані зі справедливістю, підзвітністю та потенційними упередженнями, вимагають постійного контролю, щоб уникнути посилення існуючої нерівності в екосистемі кредитування.

Управління портфелем: Машинне навчання допомагає оптимізувати інвестиційні портфелі, враховуючи безліч факторів, у тому числі стійкість до ризику, ринкові умови та попередні показники. Робо-консультанти на основі ML надають автоматизовані інвестиційні консультації на основі даних. Проблеми включають усунення потенційної концентрації ризиків у портфелях, керованих ML, і забезпечення прозорості в процесі прийняття рішень для зміцнення довіри інвесторів.

Кібербезпека у фінансах: Застосування ML у кібербезпеці для фінансової галузі виходить за рамки традиційних методів. Моделі

виявлення аномалій ідентифікують незвичайні шаблони в мережевому трафіку, підвищуючи здатність виявляти та запобігати кіберзагрозам. Постійний розвиток кіберзагроз вимагає постійного вдосконалення моделей машинного навчання, щоб випереджати складні атаки.

Незважаючи на прогрес у машинному навчанні для управління фінансовими ризиками, проблеми залишаються. Здатність інтерпретувати та пояснювати моделі залишаються вирішальними, особливо під час прийняття серйозних рішень. Усунення упереджень в алгоритмах для забезпечення справедливих результатів є постійною проблемою, що вимагає відданості етичним практикам ШІ. Нормативно-правова база все ще розвивається, що підкреслює потребу фінансових установ орієнтуватися в вимогах відповідності та активно брати участь у формуванні відповідальної політики штучного інтелекту в галузі. Етичні міркування, прозорість і підзвітність мають бути пріоритетними, щоб створити та зберегти довіру до впровадження машинного навчання в управлінні фінансовими ризиками.

1.5 Висновки до розділу 1

У розділі «Огляд управління фінансовими ризиками та їх запобігання за допомогою методів машинного навчання» розглядаються різні важливі аспекти управління фінансовими ризиками. Дослідження починається з повного розуміння фундаментальних концепцій управління фінансовими ризиками, а потім детально розглядаються статистичні моделі, включаючи моделі VaR, CVaR, RVaR і EVaR. Включення моделей GARCH ще більше збагачує дискусію, надаючи детальний погляд на волатильність і ризик.

У розділі також досліджується перетин машинного навчання та управління фінансовими ризиками, проливаючи світло на інноваційні підходи. Заглиблюючись у сучасні додатки та методології, він акцентує увагу на еволюції управління фінансовими ризиками у сфері машинного навчання. Синтез традиційних моделей і передових технологій демонструє адаптивність, необхідну в сучасному фінансовому середовищі.

Цей розділ є вичерпним посібником, який пропонує цілісне уявлення про управління фінансовими ризиками та їх запобігання, бездоганно поєднуючи загальноприйняту думку з трансформаційною силою методів машинного навчання. Він дає читачам детальне розуміння стану управління фінансовими ризиками, створюючи основу для прийняття обґрунтованих рішень у динамічному та складному світі фінансів.

1.6 Постановка задачі дослідження

Дослідницька проблема, розглянута в цьому дипломі, обертається навколо ефективного використання методів машинного навчання для запобігання та управління фінансовими ризиками. Дослідження складається з трьох основних розділів:

Огляд управління фінансовими ризиками та їх запобігання за допомогою методів машинного навчання - цей розділ містить повне розуміння управління фінансовими ризиками та досліджує інтеграцію методів машинного навчання для запобігання та пом’якшення цих ризиків. Він має на меті створити основу для подальших дискусій щодо застосування машинного навчання у фінансовій сфері.

Аналіз методів і алгоритмів машинного навчання - цей розділ, зосереджений на заглибленні в тонкощі машинного навчання, критично аналізує різні методи й алгоритми, пов’язані з управлінням фінансовими ризиками. Він намагається визначити сильні та слабкі сторони та

придатність різних підходів до машинного навчання в контексті прогнозування та запобігання фінансовим ризикам.

Аналіз роботи та оцінка якості моделей прогнозування фінансових ризиків - у цьому розділі детально розглядаються існуючі моделі прогнозування фінансових ризиків, які використовують машинне навчання. Метою є оцінка практичної реалізації цих моделей, їх ефективності в реальних сценаріях, а також встановлення критеріїв оцінки якості та надійності таких систем прогнозування.

Завдяки цим розділам дипломна робота має на меті зробити внесок у розвиток знань у сфері управління фінансовими ризиками, надаючи розуміння застосування машинного навчання, оцінюючи методології та критично оцінюючи ефективність і якість існуючих моделей прогнозування.

РОЗДІЛ 2. АНАЛІЗ МЕТОДІВ ТА АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

2.1 Загальні відомості

Машинне навчання — це галузь штучного інтелекту (ШІ), яка зосереджена на розробці алгоритмів і моделей, які дозволяють комп'ютерам навчатися на основі даних. Мета полягає в тому, щоб дозволити комп'ютерам покращити свою продуктивність у виконанні завдання з часом без явного програмування. По суті, системи машинного навчання використовують дані, щоб ідентифікувати закономірності, робити прогнози або оптимізувати процеси прийняття рішень.

Кероване навчання – це тип машинного навчання, де алгоритм навчається на позначеному наборі даних, тобто кожна точка вхідних даних пов'язана з відповідним бажаним виходом. Алгоритм навчається зіставляти вхідні дані з правильними вихідними, узагальнюючи приклади з мітками. Процес навчання передбачає коригування параметрів моделі для мінімізації різниці між прогнозованими та фактичними результатами. Після навчання модель може робити прогнози на основі нових, невідомих даних. Як приклад, класифікація зображень, де алгоритм навчається на наборі даних зображень із пов'язаними мітками, щоб навчитися класифікувати нові зображення.

Некероване навчання - передбачає навчання моделі на немаркованому наборі даних, де алгоритм повинен знаходити закономірності чи зв'язки в даних без явних вказівок. Мета полягає в тому, щоб виявити притаманну структуру даних, таку як кластери чи асоціації, без надання маркованих виводів. Некероване навчання часто використовується для дослідницького аналізу даних і отримання розуміння базових закономірностей у даних. Як приклад, алгоритми кластеризації,

коли алгоритм групує подібні точки даних без попереднього знання їхніх міток.

Навчання з підкріпленням — це парадигма машинного навчання, де агент вчиться приймати рішення, взаємодіючи з середовищем. Агент отримує зворотній зв'язок у вигляді винагород або покарань на основі дій, які він виконує. Мета навчання з підкріпленням полягає в тому, щоб агент засвоїв політику або стратегію, яка максимізує кумулятивну винагороду з часом. Це передбачає баланс між дослідженням (випробуванням нових дій) і використанням (вибором дій із відомими позитивними результатами). Як приклад, навчання комп'ютерної програми грі, де програма отримує винагороду за перемогу та штрафи за поразку, зрештою навчаючись приймати оптимальні рішення для максимізації загальної винагороди.

Ці три типи навчання (кероване, некероване та навчання з підкріпленням) представляють різні підходи до вирішення проблем машинного навчання, кожен з яких підходить для різних типів завдань і сценаріїв даних.

2.2 Задачі машинного навчання в управлінні та запобіганні фінансовими ризиками

Фінансові установи стикаються з безліччю проблем, пов'язаних зі зменшенням фінансових ризиків, включаючи кредитний ризик, ринковий ризик і операційний ризик. Ескалація складності фінансових ринків у поєднанні з великою кількістю доступних даних зробила традиційні підходи до управління ризиками недоцільними. Методи та алгоритми машинного навчання стали потужними інструментами, що покращують управління фінансовими ризиками та їх запобігання, надаючи точніші прогнози, глибшу інформацію та розширені можливості прийняття рішень.

Оцінка кредитного ризику - оцінка ймовірності дефолту позичальника за кредитом.

Методи та алгоритми:

- *Логістична регресія* - оцінка ймовірності дефолту на основі історичних даних.

- *Дерева рішень і випадкові ліси* - фіксація нелінійних зв'язків і взаємодії функцій.

- *Машини підтримки векторів* - класифікація позичальників за категоріями ризику.

- *Нейронні мережі* - робота зі складними шаблонами та залежностями.

Прогноз ринкового ризику - прогнозування коливань на фінансових ринках та інвестиційних портфелях.

Методи та алгоритми:

- *Аналіз часових рядів* - аналіз історичних ринкових даних для визначення тенденцій і закономірностей.

- *Процеси Гауса* - моделювання невизначеності та прогнозування цін на активи.

- *Методи ансамблю* - поєднання прогнозів з кількох моделей для підвищення точності.

- *Підкріплення навчання* - адаптація інвестиційних стратегій на основі мінливих ринкових умов.

Виявлення шахрайства - виявлення шахрайських дій і транзакцій.

Методи та алгоритми:

- *Виявлення аномалій* - виявлення незвичайних моделей і відхилень від нормальної поведінки.

- *Алгоритми кластеризації* - групування подібних транзакцій для виявлення викидів.

- Нейронні мережі - вивчення складних моделей, що вказують на шахрайську поведінку.

- *Обробка природної мови* - аналіз текстових даних на предмет ознак шахрайства.

Управління операційними ризиками - зменшення ризиків, пов’язаних із внутрішніми процесами, системами та людськими помилками.

Методи та алгоритми:

- *Процесна аналітика* - аналіз журналів подій для виявлення неефективності та потенційних ризиків.

- *Байєсівські мережі* - моделювання залежностей між різними операційними змінними.

- *Дерева рішень* - визначення критичних точок прийняття рішень та їх впливу на операції.

- *Прогнозна аналітика* - передбачення системних збоїв або помилок до їх виникнення.

Відповідність нормативним вимогам - забезпечення дотримання фінансових правил і стандартів відповідності.

Методи та алгоритми:

- *Системи на основі правил* - впровадження попередньо визначених правил для дотримання нормативних вимог.

- *Обробка природної мови* - аналіз нормативних текстів і оновлення протоколів відповідності.

- *Моделі машинного навчання* - передбачення змін у нормах і адаптація стратегій відповідності.

- *Інтелектуальний аналіз даних* - виявлення потенційних проблем відповідності з великими наборами даних.

Методи та алгоритми машинного навчання відіграють ключову роль у підвищенні ефективності та результативності управління фінансовими ризиками та їх запобігання. Завдяки використанню розширеної аналітики

та автоматизованого прийняття рішень фінансові установи можуть вміло орієнтуватися в складнощах сучасного фінансового ландшафту та завчасно вирішувати потенційні ризики. Оскільки технологія продовжує розвиватися, інтеграція машинного навчання в процеси управління ризиками, ймовірно, стане ще більш невід'ємною частиною підтримки фінансової стабільності та дотримання нормативних вимог.

2.3 Моделі та алгоритми машинного навчання

У цьому сегменті буде окреслено різноманітні застосування машинного навчання у сфері фінансів, охоплюючи різні типи методів машинного навчання. Три основні категорії машинного навчання, що використовуються у фінансовому контексті, включають кероване навчання, некероване навчання та навчання з підкріпленням (рис. 2.3.1).

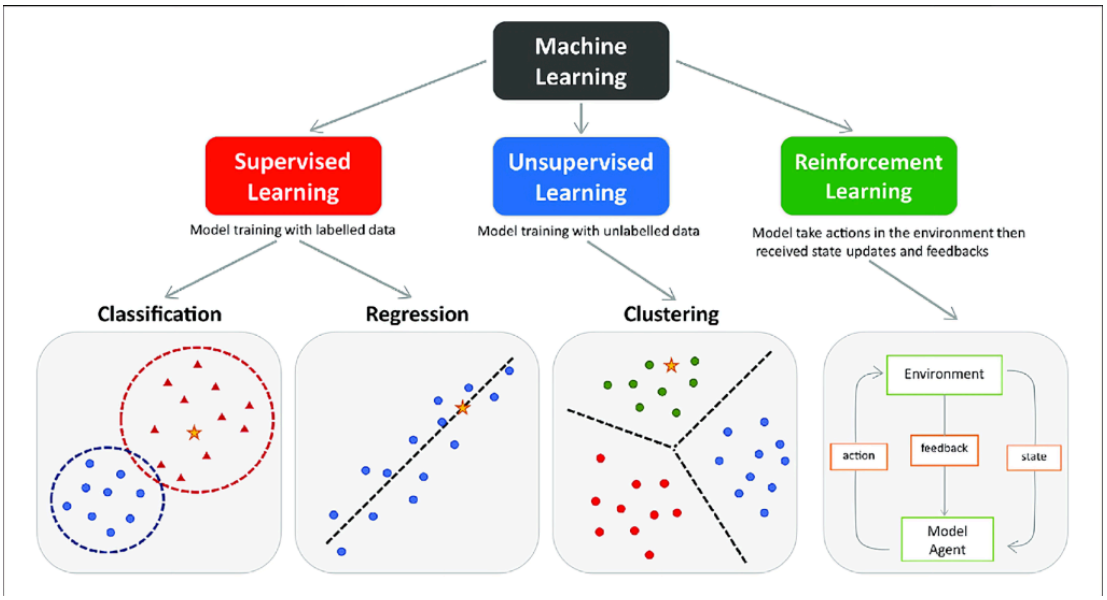

Рисунок 2.3.1 - Типи машинного навчання, кероване навчання, некероване навчання та навчання з підкріпленням.[2]

Основна мета керованого навчання – розробити модель з використанням позначених даних, що дозволяє прогнозувати невидимі або

майбутні дані. У керованому навчанні термін «кероване» означає набір даних із відомими бажаними вихідними сигналами. Два поширених типи алгоритмів навчання під наглядом – це класифікація, яка призначає мітки даним, і регресія, яка передбачає числові значення.

Класифікація, підмножина керованого навчання, спрямована на прогнозування категорійних міток класу для нових екземплярів на основі історичних спостережень. І навпаки, регресія, інша підмножина керованого навчання, зосереджена на прогнозуванні постійних результатів. У регресії змінні предиктора використовуються для встановлення зв'язку зі змінною безперервної реакції.

Некероване навчання – це підхід до машинного навчання, який використовується для отримання інформації з наборів даних, у вхідних даних яких немає позначених відповідей. Ця категорія охоплює два основних типи: зменшення розмірності та кластеризація.

Зменшення розмірності передбачає мінімізацію кількості функцій або змінних у наборі даних із збереженням інформації та загальної продуктивності моделі. Цей метод особливо корисний для керування наборами даних із високою розмірністю. З іншого боку, кластеризація, підмножина методів некерованого навчання, спрямована на виявлення прихованих структур у даних. Основна мета кластеризації полягає в тому, щоб визначити природні групи в даних, де елементи в одному кластері виявляють більшу схожість один з одним, ніж елементи в різних кластерах.

Навчання з підкріпленням - зосереджується навколо фундаментальної концепції отримання знань із досвіду разом із відповідними винагородами чи покараннями. Основна мета полягає в тому, щоб вжити відповідних дій, щоб максимізувати винагороду в конкретному контексті. Система навчання, яку називають агентом, бере участь у процесі спостереження за навколишнім середовищем, вибору та виконання дій і

подальшого отримання винагороди (або покарань у формі негативної винагороди), як показано на рисунку 2.1.

Навчання з підкріпленням відрізняється від керованого навчання у вирішальному аспекті. На відміну від керованого навчання, де навчальні дані надають моделі остаточний ключ відповіді, навчання з підкріпленням не має чіткої відповіді. У навчанні з підкріпленням агент визначає свої дії для даного завдання і дізнається правильність цих дій на основі отриманих винагород. Алгоритм на основі свого досвіду встановлює ключ відповіді.

Процес навчання з підкріпленням розгортається за такими етапами:

- Агент ініціює взаємодію з середовищем, виконуючи дію.
- Згодом агент отримує винагороду залежно від виконаної дії.
- Агент на основі винагороди отримує спостереження та розпізнає ефективність дії.

Якщо дія принесла позитивну винагороду, агент схильний повторити її, в іншому випадку він досліджує альтернативні дії в пошуках позитивної винагороди. Процес в собі втілює підхід до навчання методом проб і помилок.

2.4 Кероване навчання

У сфері фінансів моделі керованого навчання виділяються як широко використовувана категорія моделей машинного навчання. Чисельні алгоритми, які широко використовуються в алгоритмічній торгівлі, спираються на моделі керованого навчання завдяки їхнім ефективним можливостям навчання, стійкості до різних фінансових даних і надійним зв’язкам із теоретичними основами фінансів.

Проблеми класифікаційного прогнозного моделювання відрізняються від задач регресійного прогнозного моделювання, де класифікація передбачає прогнозування дискретної мітки класу, а регресія

передбачає прогнозування безперервної величини. Незважаючи на цю відмінність, обидві поділяють фундаментальну концепцію використання відомих змінних для прогнозів, що призводить до значного збігу між двома моделями. Певні моделі, включаючи K-найближчі сусіди, дерева рішень, опорні векторні машини, методи пакетування/підсилення ансамблю та штучні нейронні мережі, демонструють універсальність, оскільки їх можна адаптувати як для класифікації, так і для регресії з незначними модифікаціями. І навпаки, такі моделі, як лінійна регресія та логістична регресія, стикаються з проблемами у застосуванні до обох типів проблем.

2.4.1 Лінійна регресія

Лінійний регресійний аналіз використовується для прогнозування значення змінної на основі значення іншої змінної. Змінна, яку потрібно передбачити, називається залежною змінною, тоді як змінна, яка використовується для прогнозування її значення, називається незалежною змінною.

Цей аналітичний підхід передбачає визначення коефіцієнтів лінійного рівняння, яке містить одну чи більше незалежних змінних, щоб оптимізувати прогноз значення залежної змінної. Лінійна регресія встановлює пряму лінію або поверхню, яка мінімізує розбіжності між прогнозованими та фактичними вихідними значеннями (рис. 2.4.1.1).

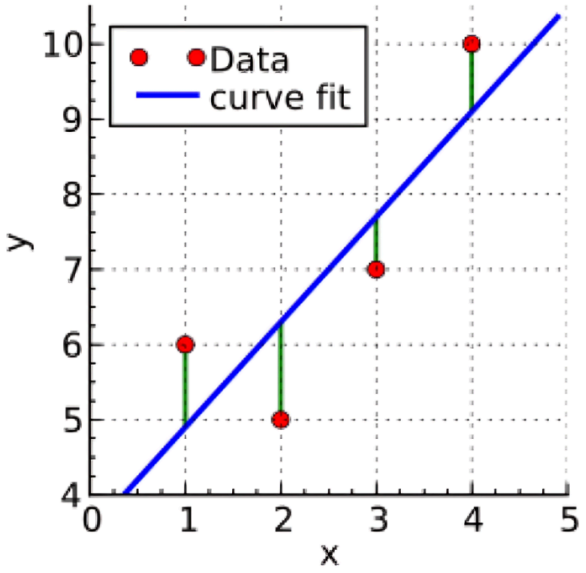

Рисунок 2.4.1.1 - Лінійна регресія.[3]

Візуально, у двох вимірах, це дає найкраще відповідне лінійне зображення, проілюстроване на малюнку 2.3.1.1. У вищих вимірах ми мали б гіперплощини вищих вимірів. Математично ми оцінюємо дисперсію між кожною фактичною точкою даних (y) і прогнозом нашої моделі (ŷ). Щоб запобігти негативним значенням і підкреслити більші відхилення, ми зводимо ці різниці в квадрат, підсумовуємо їх і обчислюємо середнє значення. Цей процес служить індикатором того, наскільки добре наші дані відповідають запропонованій лінії.

Рівняння для простої лінійної регресії задається так:

$$Y = \beta_0 + \beta_1 X + \varepsilon \qquad (2.4.1.1)$$

Де, Y - залежна змінна або змінна, яку ми намагаємося передбачити, X — незалежна змінна або змінна, яка використовується для прогнозування, β_0 - перетин, значення Y, коли X дорівнює нулю, β_1 - нахил, представляє зміну Y для зміни X на одну одиницю, ε - помилка, що представляє неспостережувані фактори, що впливають на Y, які не пояснюються X.

Метою простої лінійної регресії є оцінка значень β_0 і β_1, які мінімізують суму квадратів різниць між спостережуваними (Y_i) і прогнозованими ($\widehat{Y_i}$) значеннями, відомими як залишки (ε_i):

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2 \qquad (2.4.1.2)$$

Метод найменших квадратів використовується для знаходження оптимальних значень β_0 і β_1 шляхом мінімізації цієї суми.

Для знаходження цих оптимальних значень виводяться формули для β_0 і β_1:

$$\beta_1 = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n} (X_i - \overline{X})^2} \qquad (2.4.1.3)$$

$$\beta_0 = \overline{Y} - \beta_1 \overline{X} \qquad (2.4.1.4)$$

Де $\overline{X}$ - середнє значення незалежної змінної X, $\overline{Y}$ - середнє значення залежної змінної Y.

Ці формули надають коефіцієнти, які визначають лінійний зв’язок між змінними.

Для множинної лінійної регресії з n незалежними змінними (X_1, X_2, …, X_n) рівняння виглядає так:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon \qquad (2.4.1.5)$$

Відповідні формули для β_0, β_1, … , β_n отримані з використанням матричного запису, і мета залишається мінімізувати суму квадратів залишків. Рішення включає в себе методи лінійної алгебри, які зазвичай розв’язуються за допомогою таких методів, як нормальне рівняння або градієнтний спуск.

Переваги лінійної регресії полягають у її легкій зрозумілості та інтерпретованості, що робить їх доступними для широкого кола користувачів. Цей тип моделі вимагає менше обчислень порівняно зі складнішими моделями, що робить її ефективною для обробки великих об'ємів даних та в реальному часі. Коли зв'язок між змінними приблизно лінійний, лінійна регресія може забезпечити точні прогнози та надійні результати.

Лінійна регресія надає оцінки коефіцієнтів лінійного рівняння, що дозволяє кількісно зрозуміти взаємозв'язок між змінними. Коефіцієнти в моделі лінійної регресії можуть вказувати на важливість кожної ознаки для прогнозування залежної змінної, що допомагає ідентифікувати ключові фактори впливу. Таким чином, лінійна регресія є ефективним інструментом для аналізу та моделювання залежностей між змінними в даних.

Недоліки лінійної регресії включають кілька аспектів. По-перше, цей метод передбачає лінійний зв’язок між залежною та незалежною змінними, і в разі порушення цього припущення модель може надавати неточні результати. Також слід враховувати, що лінійна регресія чутлива до викидів, і вже одна екстремальна точка даних може значно вплинути на коефіцієнти та прогнози моделі. Лінійна регресія передбачає постійну дисперсію залишків на всіх рівнях незалежних змінних. Порушення цього припущення може призвести до неефективних оцінок параметрів. У випадку сильної кореляції між незалежними змінними, оцінки коефіцієнтів можуть бути нестабільні, ускладнюючи інтерпретацію внесків кожної змінної.

Важливо зауважити, що лінійна регресія не є відповідною для фіксації складних нелінійних зв’язків між змінними, і для таких випадків може знадобитися використання більш досконалих моделей. Крім того,

лінійна регресія може страждати від недостатнього пристосування, якщо складність моделі обрана невірно.

2.4.2 Регуляризована регресія

Регуляризована регресія — це тип лінійної регресії, який включає умови регуляризації в цільову функцію. Мета регуляризованої регресії полягає в тому, щоб знайти коефіцієнти характеристик, які найкраще відповідають даним, одночасно штрафуючи великі коефіцієнти, щоб запобігти перенавчанню. Існує два поширених типи регуляризованої регресії: регресія Ласо (регуляризація L1) і регресія Ріджа (регуляризація L2).

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \qquad (2.4.2.1)$$

Де J(θ) – функція вартості, m – кількість прикладів навчання, $h_\theta(x^{(i)})$ – прогнозоване значення для і-го прикладу навчання, $y^{(i)}$ – фактичне значення для і-го прикладу навчання.

Регресія ласо додає член регуляризації до цільової функції лінійної регресії, використовуючи норму L1 вектора коефіцієнтів (θ). Цільова функція регресії ласо визначається як:

$$J_{lasso}(\theta) = J(\theta) + \lambda\sum_{j=1}^{n}\left|\theta_j\right| \qquad (2.4.2.2)$$

Де λ – параметр регуляризації, n – кількість ознак.

Гребнева регресія додає член регуляризації до цільової функції лінійної регресії за допомогою норми L2 вектора коефіцієнтів (θ). Цільова функція гребневої регресії визначається як:

$$J_{ridge}(\theta) = J(\theta) + \lambda\sum_{j=1}^{n}\theta_j^2 \qquad (2.4.2.3)$$

Еластично-сіткова регуляризація - це комбінація регуляризації L1 і L2. Цільова функція є лінійною комбінацією цільових функцій ласо та гребневої:

$$J_{elastic\ net}(\theta) = J(\theta) + \lambda_1 \sum_{j=1}^{n} \left|\theta_j\right| + \lambda_2 \sum_{j=1}^{n} \theta_j^2 \qquad (2.4.2.4)$$

Де λ_1 і λ_2 є параметрами регуляризації для регулярізації L1 і L2 відповідно.

Параметр регуляризації (λ або λ1, λ2) контролює силу регуляризації. Більший λ призводить до більш агресивної регулярізації, яка допомагає запобігти перенавчанню, штрафуючи великі коефіцієнти. Вибір параметра регуляризації є вирішальним і часто визначається за допомогою методів перехресної перевірки.

Далі розглянемо алгоритм, який ідеально підходить для визначення зв’язку між ознаками чи характеристиками та конкретним результатом: логістична регресія.

2.4.3 Логістична регресія

Логістична регресія — це статистичний метод, який широко використовується в машинному навчанні для задач бінарної класифікації, де метою є прогнозування ймовірності приналежності екземпляра до певного класу. Незважаючи на свою назву, логістична регресія використовується для класифікації, а не регресії.

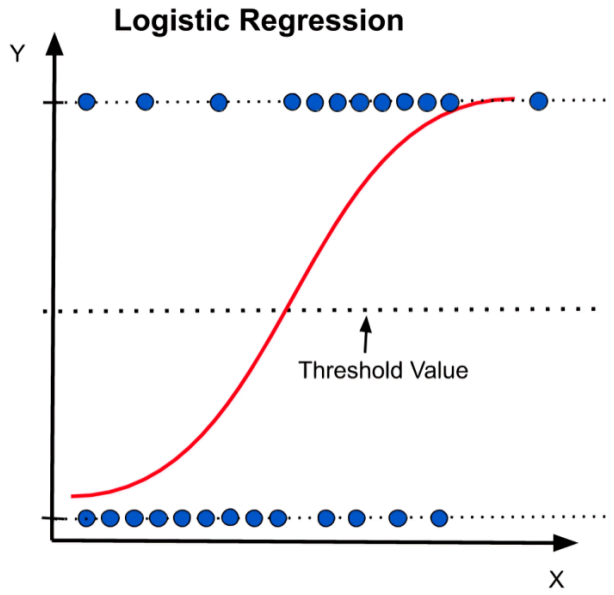

Рисунок 2.4.3.1 - Логістична регресія.[4]

Логістична регресія в основному використовується для задач бінарної класифікації, де є два можливі результати або класи, які часто позначаються як 0 і 1.

В основі логістичної регресії лежить сигмоїдна функція, позначена σ(z):

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (2.4.3.1)$$

Де z — лінійна комбінація вхідних ознак.

Сигмоїдна функція відображає будь-яке дійсне число в діапазоні [0, 1]. Це має вирішальне значення для логістичної регресії, оскільки вона перетворює результат на ймовірність. Логістична функція має S-подібну криву, що дозволяє вловити нелінійний зв’язок між вхідними характеристиками та прогнозованою ймовірністю.

Лінійна комбінація вхідних характеристик представлена як:

$$z = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \quad (2.4.3.2)$$

Де x_1, x_2,..., x_n є вхідними ознаками, і β_0, β_1,..., β_n це коефіцієнти, які потрібно отримати з даних навчання.

Логістична функція застосовується до лінійної комбінації для моделювання логарифмічних коефіцієнтів:

$$logodds = log(\frac{p}{p-1}) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n \qquad (2.4.3.3)$$

Де p — ймовірність приналежності екземпляра до позитивного класу.

Метою логістичної регресії є знаходження оптимальних значень для коефіцієнтів (β). *Функція вартості* – міра похибки між прогнозованими ймовірностями та справжніми мітками класу. Ця функція виглядає так:

$$J(\beta) = -\frac{1}{m}\sum_{i=1}^{m}[y_i log(\widehat{y}_i) + (1 - y_i) log(1 - \widehat{y}_i)] \qquad (2.4.3.4)$$

Де m — кількість сутностей у навчальному наборі, y_i — фактична мітка класу і-ї сутності (0 або 1), а $\widehat{y}_i$ — прогнозована ймовірність того, що $y_i = 1$.

Коефіцієнти (β) вивчаються шляхом мінімізації функції витрат. Зазвичай це робиться за допомогою алгоритмів оптимізації, таких як градієнтний спуск.

Порівнюючи лінійну регресію та логістичну регресію в машинному навчанні, отримаємо, що лінійна регресія має безперервну залежну змінну, яка виводить значення в спектрі дійсних чисел. Прогнози робляться з використанням найкраще підігнаної лінії з діапазоном від негативної до позитивної нескінченності. Мета полягає в мінімізації суми квадратів різниці між прогнозованими та фактичними значеннями. Коефіцієнти означають зміну залежної змінної при зсуві незалежної змінної на одну одиницю. Цей метод підходить для прогнозування безперервних результатів припускаючи лінійний зв’язок між змінними, хоча він чутливий до відхилень.

З іншого боку, логістична регресія має справу з категоріальною залежною змінною, зазвичай двійковою (0 або 1), що представляє класи.

Результатом є ймовірність у діапазоні від 0 до 1, що вказує на ймовірність належності до певного класу. Прогнози обмежені між 0 і 1 через сигмоїдну функцію. Мета полягає в тому, щоб мінімізувати втрати, вимірюючи невідповідність між прогнозованими ймовірностями та фактичними мітками класу. Коефіцієнти відображають зміну логарифмічних коефіцієнтів залежної змінної для зсуву незалежної змінної на одну одиницю. Логістична регресія зазвичай використовується для завдань бінарної класифікації і передбачає, що логарифмічні коефіцієнти є лінійною комбінацією незалежних змінних. Він виявляється більш стійким до відхилень, особливо з надійними методами оптимізації.

Таким чином, хоча і лінійна, і логістична регресії є методами регресії, вони пристосовані для різних проблем. Лінійна регресія підходить для безперервних результатів, тоді як логістична регресія розроблена для завдань бінарної класифікації з категоричними результатами.

2.4.4 Дерева класифікації та регресії

Дерева класифікації та регресії (CART) – це тип алгоритму дерева рішень, який використовується в машинному навчанні як для завдань класифікації, так і для регресії. Дерева рішень є потужним інструментом, який можна інтерпретувати, для створення прогнозів на основі серії двійкових рішень.

Дерево рішень — це деревоподібна структура, у якій кожен внутрішній вузол представляє рішення на основі функції, кожна гілка — результат цього рішення, а кожен листовий вузол — остаточний прогнозований результат (клас для класифікації або числове значення для регресії).

Побудова дерева рішень передбачає рекурсивний процес поділу набору даних на основі функції, яка забезпечує найкраще зменшення домішок (для класифікації) або зменшення дисперсії (для регресії). Алгоритм продовжує цей процес поділу, доки не буде виконано заздалегідь визначений критерій зупинки, наприклад досягнення максимальної глибини, наявність мінімальної кількості зразків у листі або досягнення мінімального порогу домішок.

У дереві класифікації цільова змінна є категоріальною, а метою є класифікація вхідних екземплярів в один із кількох попередньо визначених класів.

Коефіцієнт Джині коливається від 0 до 1, де 0 означає ідеальну чистоту (всі екземпляри належать до одного класу), а 1 означає максимальну забрудненість.

Мета полягає в тому, щоб мінімізувати коефіцієнт Джині, вибравши розбиття, яке максимально розділяє класи. За набору точок даних із двома класами p і q індекс Джині вимірює домішки або невпорядкованість даних:

$$G = 1 - \sum_{i=1}^{n} P_i^2 \qquad (2.4.4.1)$$

Де P_i - це ймовірність приналежності екземпляра до класу i.

Коефіцієнт приросту інформації вимірює зменшення ентропії після поділу набору даних:

$$IG(D, A) = H(D) - \sum_{v} \frac{|D_v|}{|D|} H(D_V) \qquad (2.4.4.2)$$

Де H(D) - ентропія набору даних D, D_v - підмножина даних, для якої функція A має значення v, а $H(D_v)$ - ентропія підмножини D_v.

Коефіцієнт приросту інформації допомагає визначити функцію, яка надає найбільше інформації про цільову змінну.

Середньоквадратична похибка вимірює середню квадратичну різницю між прогнозованими та справжніми значеннями. За набору точок

даних із справжніми значеннями y_i та прогнозованими значеннями $\hat{y}_i$ середньоквадратична помилка обчислюється як:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (2.4.4.3)$$

Переваги CART включають можливість інтерпретації, оскільки дерева рішень легко зрозуміти та пояснити, забезпечуючи прозорість модельних рішень. Крім того, дерева рішень чудово фіксують нелінійні зв'язки між функціями та цільовими змінними, що дозволяє їм моделювати складні закономірності в даних. Крім того, CART демонструє надійність у обробці відсутніх значень без врахування, що робить його універсальним і практичним алгоритмом для різних завдань машинного навчання.

Хоча CART (дерева класифікації та регресії) є універсальним і широко використовуваним алгоритмом, відомим своєю простотою та ефективністю, він має обмеження, які слід враховувати. Одним із помітних недоліків є сприйнятливість до перенавчання, особливо коли дерева рішень стають глибокими та вловлюють шум у даних. Крім того, CART може бути чутливим до невеликих варіацій у наборі даних, що призводить до різних структур дерева з незначними змінами. Незважаючи на ці обмеження, CART служить основоположним алгоритмом і формує основу для більш просунутих методів ансамблю, таких як випадкові ліси та посилення градієнта, які спрямовані на пом'якшення цих проблем і підвищення ефективності прогнозування.

2.4.5 Метод k-найближчих сусідів

Метод k-найближчих сусідів — це непараметричний класифікатор із керованим навчанням, який використовується для прогнозування або класифікації на основі близькості окремих точок даних. Незважаючи на те,

що він може вирішити як проблеми регресії, так і проблеми класифікації, він в основному використовується як алгоритм класифікації, заснований на концепції, згідно з якою подібні точки даних мають тенденцію групуватися разом.

Рисунок 2.4.5.1 - Діаграма методу k-найближчих сусідів.[5]

У контексті завдань класифікації алгоритм призначає мітку класу точці даних через механізм більшості голосів. Тобто, обирається мітка, яка найчастіше з’являється біля певної точки даних. Варто зазначити, що різниця полягає в тому, що «голосування більшості» суворо вимагає більшості понад 50%, концепція, яка добре піддається бінарній класифікації. У сценаріях з декількома класами, наприклад чотирма категоріями, вимога більшості 50% не є обов’язковою. Натомість мітку класу можна призначити на основі голосування, що перевищує 25%, що дозволяє гнучко визначити приналежність точки даних до класу.

Задачі регресії дотримуються концепції, подібної до класифікації, з ключовою відмінністю в тому, що вони використовують середнє значення k найближчих сусідів, щоб робити прогнози щодо безперервної змінної. На відміну від класифікації, яка має справу з дискретними значеннями, регресія спеціально розроблена для сценаріїв із безперервними

значеннями. Встановлення відстані є вирішальним кроком перед класифікацією, і евклідова відстань виділяється як найбільш часто використовувана метрика.

Метод k-найближчих сусідів відноситься до категорії моделей «ледачого навчання». Ця характеристика означає, що він зберігає лише навчальний набір даних без проходження спеціального етапу навчання. Отже, усі обчислення відбуваються під час класифікації чи прогнозування. Оскільки алгоритм значною мірою покладається на пам’ять для зберігання всього набору навчальних даних, його часто характеризують як метод навчання на основі екземплярів або пам’яті.

Щоб визначити, які точки даних є найближчими до певної точки запиту, необхідно обчислити відстань між точкою запиту та іншими точками даних. Ці показники відстані допомагають сформувати межі рішень, які розділяють запити на різні області.

Алгоритм пошуку сусідів:

- Обчисліть відстані.

- Відсортуйте відстані в порядку зростання.

- Виберіть верхні k точок з найменшими відстанями.

Метрика відстані визначає, як виміряти подібність або відмінність між двома точками даних.

Евклідова метрика:

$$d(X_i, X_{new}) = \sqrt{\sum_{j=1}^{n} (x_{ij} - x_{newj})^2} \qquad (2.4.5.1)$$

Манхэттенская метрика:

$$d(X_i, X_{new}) = \sum_{j=1}^{n} |x_{ij} - x_{newj}| \qquad (2.4.5.2)$$

Метрика Мінковського:

$$d(X_i, X_{new}) = \left(\sum_{j=1}^{n} |x_{ij} - x_{newj}|^p\right)^{\frac{1}{p}} \quad (2.4.5.3)$$

Де X_i - набір даних із навчального набору, X_{new} - нова точка даних для прогнозу, x_{ij} - ознака j точки даних X_i, n - кількість ознак, p – параметр, що контролює розрахунок відстані. Коли p = 2, вона стає евклідовою відстанню.

Виникають проблеми з обчислювальними витратами на обчислення відстані між новою точкою та усіма точками в наборі даних, особливо з більшими наборами даних. Ця складність зростає лінійно з розміром набору даних і кількістю функцій. Крім того, зберігання всього навчального набору даних у пам’яті може призвести до збільшення використання пам’яті та сповільнення прогнозів, особливо з великими наборами даних.

Метод k-найближчих сусідів однаково обробляє всі функції, потенційно впливаючи на продуктивність, якщо деякі функції є нерелевантними. Вибір k має вирішальне значення, причому мале k створює чутливість до шуму, а велике k потенційно згладжує основні шаблони. У задачах класифікації з незбалансованими наборами даних метод k-найближчих сусідів може проявляти зміщення щодо більшості класів. Час передбачення може бути відносно повільним, особливо у просторі з великою розмірністю, де розмірність стає проблемою.

Таким чином, метод k-найближчих сусідів є цінним інструментом, але може бути не універсальним. Такі аспекти, як розмір набору даних, розмірність і обчислювальні ресурси, є ключовими для визначення того, чи є даний метод або альтернативні алгоритми машинного навчання найбільш підходящим вибором для сценарію.

2.4.6 Метод опорних векторів

Основна мета методу опорних векторів – максимізувати розділ. Цей запас відноситься до відстані між розділовою гіперплощиною та навчальними зразками, які є найближчими до цієї гіперплощини, які називаються опорними векторами. Розділ обчислюється як перпендикулярна відстань від лінії лише до найближчих точок (рис. 2.4.6.1).

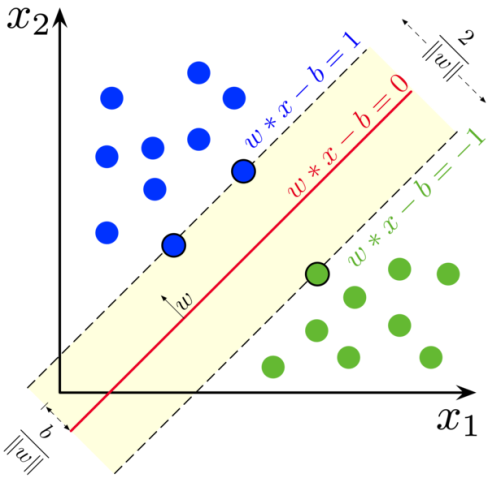

Рисунок 2.4.6.1 - Метод опорних векторів.[6]

ОВМ має на меті встановити межу рішення з максимальним розділом, створюючи чітке розділення всіх точок даних. Ця межа максимального розділу сприяє згуртованому розподілу даних.

Лінійні ОВМ використовуються з лінійно роздільними даними; це означає, що дані не потребують жодних перетворень, щоб розділити дані на різні класи. Математично ця розділова гіперплощина може бути представлена у вигляді:

$$wx + b = 0 \qquad (2.4.6.1)$$

Де w — ваговий вектор, x — вхідний вектор, b — зсув.

Для визначення розділу, тобто максимального поділу між класами, використовуються два методи: класифікація жорсткого розділу та

класифікація м’якого розділу. У випадку ОВМ з жорстким роздільним значенням точки даних точно відокремлюються поза опорними векторами.

Це поняття виражається наступною формулою:

$$(wx_j + b)y_j \geq a \quad (2.4.6.2)$$

Розділ максимізується, що представляється як:

$$max = \frac{a}{||w||} \quad (2.4.6.3)$$

де a — розділ, спроектований на w.

Класифікація з м’яким розділом забезпечує гнучкість завдяки включенню змінних, які мають вільний рівень, що допускає певний ступінь неправильної класифікації. Гіперпараметр C відіграє ключову роль у регулюванні розділу, більш високе значення C зменшує розділ, щоб мінімізувати помилкову класифікацію, тоді як нижче значення C збільшує розділ, вміщуючи більше неправильно класифікованих даних.

У сценаріях реального світу велика частина даних не є лінійно роздільною, що спонукає до використання нелінійних ОВМ. Щоб перетворити такі дані в лінійно розділену форму, до даних для навчання моделі застосовуються методи попередньої обробки, які часто передбачають перехід до простору ознак більшої розмірності. Однак робота в просторі з більшою вимірністю може ускладнювати роботу, створюючи ризики перенавчання та обчислювального навантаження.

2.4.7 Випадковий ліс

Випадковий ліс — це метод ансамблевого навчання, який використовується як для завдань класифікації, так і для регресії. Він будується з кількох дерев рішень під час навчання та виводить метод (для класифікації) або середнє передбачення (для регресії) окремих дерев. Випадковий ліс створює кілька дерев рішень і об’єднує їх, щоб отримати

точніший і стабільніший прогноз, ніж будь-яке окреме дерево. «Випадковість» у методі випадкового лісу походить з двох основних джерел випадковості:

- Для кожного дерева в лісі вибирається випадкова підмножина набору даних із заміною. Це означає, що деякі точки даних можуть повторюватися, а інші можуть бути пропущені.

$$D_i = \{(x_1, y_1), (x_2, y_2),..., (x_n, y_n)\} \quad (2.4.7.1)$$

- Під час побудови кожного дерева рішень при кожному розділенні враховується випадкова підмножина ознак. Це вносить різноманітність серед дерев.

$$M_j = \{f_{j_1}, f_{j_2},..., f_{j_p}\} \quad (2.4.7.2)$$

Алгоритм випадкового лісу базується на трьох основних гіперпараметрах, які необхідно налаштувати перед навчанням: розмір вузла, кількість дерев і кількість відібраних функцій. Після встановлення цих параметрів класифікатор може ефективно вирішувати проблеми регресії та класифікації.

Алгоритм побудовано як ансамбль дерев рішень, кожне дерево будується з вибірки даних, взятої з навчального набору за допомогою техніки, що називається початковою вибіркою, де екземпляри вибираються із заміною. Цей підхід характерний створенням вихідної вибірки, яка складається з однієї третини навчальних даних, не включених до початкової вибірки. Цей зразок вихідної вибірки відіграє вирішальну роль у процесі оцінювання.

Щоб запровадити додаткову варіативність і зменшити кореляцію між деревами рішень, ще один рівень випадковості вводиться через пакетування функцій. Це передбачає вибір випадкової підмножини функцій під час кожного розбиття в процесі побудови дерева.

Процес прогнозування залежить від характеру проблеми. Для задач регресії прогнози окремих дерев рішень усереднюються, що забезпечує остаточне прогнозування регресії. У випадку класифікаційних завдань більшість голосів проводиться серед прогнозів окремих дерев, фактично вибираючи найбільш часту категоріальну змінну як прогнозований клас.

Повертаючись до зразка вихідної вибірки, він відіграє ключову роль у перехресній перевірці. Ефективність моделі оцінюється шляхом оцінки її прогнозів на вихідній вибірці, що забезпечує надійну міру точності. Таким чином, алгоритм забезпечує повну та надійну оцінку своїх прогнозних можливостей.

Незважаючи на сильні сторони, випадковий ліс має недоліки. Однією з помітних проблем є його схильність поводитися як чорний ящик, пропонуючи слабкий контроль над його внутрішньою роботою та ускладнюючи інтерпретацію результатів. Незважаючи на відмінність у класифікаційних завданнях, він може бути не оптимальним вибором для проблем регресії, оскільки він не забезпечує точних безперервних прогнозів. У сценаріях регресії модель обмежується прогнозуванням у межах діапазону навчальних даних, що потенційно може призвести до перенавчання, особливо з масивними наборами даних. Таким чином, хоча випадковий ліс пропонує багато переваг, при застосуванні цього алгоритму повинні бути враховані його обмеження, відповідно до конкретних завдань.

2.5 Некероване навчання

Некероване навчання – це парадигма машинного навчання, у якій алгоритму доручено видобувати шаблони, зв’язки або структури з вхідних даних без явних вказівок чи позначених вихідних даних.

Алгоритми некерованого навчання прагнуть розпізнати шаблони в даних, не маючи жодних знань про очікуваний результат. Цей підхід усуває потребу в даних із мітками, які можуть бути трудомісткими та непрактичними для створення або отримання, дозволяючи ефективно використовувати великі набори даних для аналізу та розробки моделей.

Ключовою технікою некерованого навчання є зменшення розмірності. Цей процес ущільнює дані шляхом ідентифікації меншого, чіткого набору змінних, які відображають суть оригінальних функцій, мінімізуючи втрату інформації. Зменшення розмірності є важливим у вирішенні проблем, пов’язаних із високою розмірністю, сприяючи візуалізації важливих аспектів у високорозмірних даних, які інакше було б складно досліджувати.

У фінансовій сфері, яка характеризується великими наборами даних і численними вимірами, методи зменшення розмірності стають практичними та корисними інструментами. Вони дають нам можливість зменшити шум і надмірність у наборі даних, отримуючи приблизну версію з меншою кількістю функцій. Завдяки зменшеній кількості змінних дослідження та візуалізація набору даних стають простішими. Крім того, методи зменшення розмірності покращують моделі керованого навчання, зменшуючи кількість функцій.

У практиці ці методи використовують для зменшення розмірності у фінансах для різних цілей, таких як розподіл коштів між класами активів та окремими інвестиціями, визначення торгових стратегій і сигналів, впровадження хеджування портфеля та управління ризиками, а також розробка моделей ціноутворення інструментів.

2.5.1 Кластерний аналіз

І кластеризація, і зменшення розмірності відіграють вирішальну роль у групуванні даних. Зменшення розмірності передбачає ущільнення даних шляхом представлення їх із меншою кількістю функцій, зберігаючи при цьому найбільш релевантну інформацію. Навпаки, кластеризація спрямована на зменшення обсягу даних і розкриття закономірностей шляхом класифікації вихідних даних, а не створення нових змінних. За допомогою алгоритмів кластеризації спостереження розподіляються на підгрупи, що містять подібні точки даних. Основна мета кластеризації полягає в тому, щоб визначити природні групи в даних, де елементи в кластері виявляють більшу схожість один з одним, ніж елементи в різних кластерах. Цей процес покращує розуміння даних, переглядаючи їх через перспективу різних категорій або груп.

У сфері фінансів трейдери та інвестиційні менеджери використовують кластеризацію для визначення однорідних груп активів, класів, секторів і країн на основі спільних характеристик. Аналіз кластеризації збагачує торгові стратегії, пропонуючи зрозуміти різні категорії торгових сигналів. Крім того, ця техніка знайшла застосування в сегментуванні клієнтів або інвесторів на групи, сприяючи глибшому розумінню їхньої поведінки та даючи можливість додаткового аналізу на основі вивчених критеріїв. Автоматична категоризація нових об’єктів на основі цих даних додатково підвищує цінність процесів прийняття рішень у фінансах.

Існують різні методи кластеризації, кожна з яких використовує різні стратегії для ідентифікації групувань у даних. Вибір конкретної методики залежить від природи та структури набору даних. Три поширені техніки кластеризації:

- Ієрархічна кластеризація.
- Кластеризація методом k–середніх.
- Метод поширення близькості.

Ієрархічна кластеризація - тип кластеризації, яка передбачає формування кластерів з чіткою ієрархічною структурою, що демонструє переважний порядок зверху вниз (рис. 2.5.1.1).

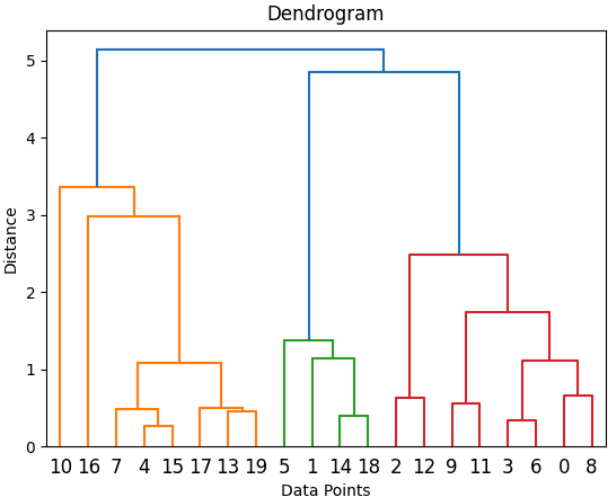

Рисунок 2.5.1.1 - Ієрархічна кластеризаці.[7]

Помітною перевагою цього підходу є усунення необхідності попереднього визначення кількості кластерів, замість цього модель автономно визначає оптимальну кластеризацію. Дві основні кластеризації в ієрархічній кластеризації — це агломеративна ієрархічна кластеризація та роздільна ієрархічна кластеризація.

Агломеративна ієрархічна кластеризація — це висхідний підхід до кластеризації, який починається з окремих точок даних і поступово об’єднує їх у більші кластери. Він утворює ієрархію кластерів, де кожен рівень представляє різні рівні деталізації у групуванні даних.

Процес агломеративної ієрархічної кластеризації:

- Кожна точка даних розглядається як окремий кластер.

- Обчислюється попарна подібність або відмінність між усіма кластерами. Загальні показники відстані включають евклідову відстань, манхеттенську відстань або коефіцієнти кореляції.

- Визначаються два найбільш схожі кластери на основі обраної метрики відстані.

- Обрані два кластери об’єднуються в новий кластер, і розглядаються як єдине ціле.

- Перераховується матриця подібності, щоб відобразити новостворений кластер і його зв'язок з іншими кластерами.

- Попередні кроки повторюються, доки не залишиться лише один кластер, корінь ієрархії.

- Для представлення ієрархії кластерів будується дендрограма. Вертикальні лінії на дендрограмі представляють злиття кластерів, а висота, на якій ці лінії з’єднуються, вказує на різницю, на якій відбулося злиття.

- Визначається потрібна кількість кластерів, розрізавши дендрограму на певній висоті. Горизонтальна лінія, проведена через дендрограму на цій висоті, представляє вибрану кількість кластерів.

- Кожну точка даних призначається відповідному кластеру на основі обраної висоти зрізу.

Агломеративна ієрархічна кластеризація є гнучкою і не вимагає попередньої вказівки кількості кластерів. Однак часова складність цього підходу вища порівняно з іншими алгоритмами кластеризації, що робить його менш придатним для дуже великих наборів даних. Загальні методи зв’язування, які використовуються для обчислення подібності між кластерами, включають одиночний зв’язок, повний зв’язок і середній зв’язок. Кожен метод по-різному визначає подібність між кластерами, впливаючи на структуру результуючої ієрархії.

Роздільна ієрархічна кластеризація — це техніка ієрархічної кластеризації, яка працює протилежно до агломеративної ієрархічної кластеризації. У той час як агломеративна кластеризація починається з окремих точок даних і поступово об’єднує їх у кластери, кластеризація з розділенням починається з одного кластера, який містить усі точки даних і рекурсивно поділяє його на менші кластери.

Процес роздільної ієрархічної кластеризації:

- Процес починається з одного кластера, який охоплює всі точки даних.

- Застосовується стратегія розділення, щоб розділити початковий кластер на менші, більш однорідні підкластери. Цей процес зазвичай керується мірою відмінності, яка ідентифікує точки в кластері, які найменш схожі одна на одну.

- До кожного новоутвореного підкластеру, рекурсивно застосовується процес поділу, розбиваючи його на ще менші кластери. Це рекурсивне розбиття триває, доки кожна точка даних не стане власним кластером, утворюючи деревоподібну структуру, відому як дендрограма.

- Алгоритм може зупинятися на основі певних критеріїв, таких як, попередньо визначена кількість кластерів або порог відмінності.

- Результат роздільної ієрархічної кластеризації часто представляється візуально через дендрограму, деревоподібну діаграму, яка ілюструє ієрархічні зв’язки між кластерами. Точки розгалуження на дендрограмі вказують, коли кластери були розділені, а висота цих гілок відображає різницю, за якої відбулося розділення.

Цей підхід пропонує детальний перегляд ієрархії кластеризації та потенційно може виявити більш складні структури в даних. Але порівняно з агломеративним кластеризуванням ,цей метод обчислювально інтенсивніший.

Ієрархічна кластеризація пропонує кілька переваг, включаючи простоту реалізації, відсутність необхідності попередньо визначати кількість кластерів і створення інформативних дендрограм для покращеного розуміння даних. Однак важливо зазначити, що часова складність ієрархічної кластеризації може призвести до збільшення тривалості обчислень порівняно з альтернативними алгоритмами, такими як кластеризація методом k–середніх. Управління великими наборами

даних стає складним, оскільки розшифровка оптимальної кількості кластерів з дендрограми може виявитися складною. Іншим міркуванням є чутливість ієрархічної кластеризації до викидів, оскільки їх присутність може помітно знизити продуктивність моделі.

Кластеризація методом k-середніх виділяється як одна з найбільш широко визнаних і використовуваних методів кластеризації. Основна мета алгоритму k-середніх — розділити набір даних на окремі класи з акцентом на групуванні точок даних, які демонструють високу схожість (рис. 2.5.1.2).

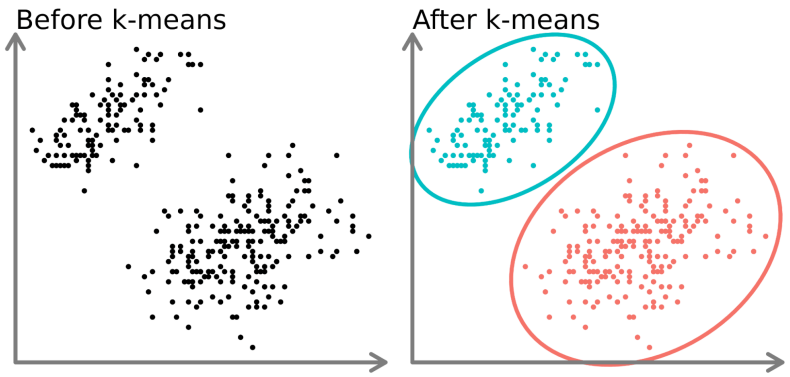

Рисунок 2.5.1.2 - Кластеризація методом k–середніх.[8]

У цьому контексті подібність обернено пропорційна відстані між точками даних. По суті, алгоритм прагне мінімізувати відстані між точками даних у межах одного кластера, виходячи з передумови, що більша близькість свідчить про більшу ймовірність приналежності до однієї згуртованої групи.

Гіперпараметри, що керують поведінкою алгоритму k-середніх, охоплюють:

Кількість кластерів – визначає бажану кількість кластерів і центроїдів, які будуть створені в наборі даних.

Максимальна кількість ітерацій – встановлює верхню межу кількості ітерацій, які алгоритм виконуватиме під час одного виконання, щоб оптимізувати призначення кластерів.

Кількість ініціалізацій - визначає, скільки разів алгоритм виконуватиметься з різними початковими числами центроїда. Кінцевий результат визначається як оптимальний результат серед заданої кількості послідовних прогонів, оцінений з точки зору інерції.

Мета алгоритму полягає в тому, щоб визначити k центроїдів і призначити кожну точку даних одному кластеру, щоб мінімізувати дисперсію всередині кластера, також відому як інерція. Хоча зазвичай використовується евклідова відстань, застосовуються альтернативні метрики відстані. Алгоритм k-середніх, який забезпечує локальний оптимум для заданого k, виконує такі дії:

- Вкажіть потрібну кількість кластерів.
- Довільно виберіть точки даних, які будуть служити початковими центрами кластерів.
- Призначте кожну точку даних найближчому центру кластера.
- Оновіть центри кластерів до середнього значення призначених точок.
- Повторюйте кроки попередны, доки всі центри кластерів не залишаться незмінними.

Універсальність цього алгоритму очевидна в його застосуванні до різноманітних типів даних, будь то числові чи категоричні, за умови, що можна визначити відповідну метрику відстані. Відомий своєю швидкістю та ефективністю алгоритм k-середніх добре підходить для завдань, що вимагають кластеризації в режимі реального часу. Він чудово працює, навіть якщо кластери мають приблизно сферичну форму та подібні розміри, а також демонструє ефективність у ідентифікації кластерів різних форм.

Окрім можливостей кластеризації, алгоритм виявився ефективним як етап попередньої обробки для інших алгоритмів машинного навчання. Ця функція допомагає зменшити розмірність і підвищує загальну

ефективність наступних аналізів. Підводячи підсумок, кластеризація алгоритм k-середніх є потужним і адаптованим інструментом, який поєднує в собі простоту, ефективність і універсальність для вирішення широкого спектру проблем кластеризації даних у різних доменах.

Метод поширення близькості — це алгоритм кластеризації даних, який відноситься до категорії методів на основі зразків. На відміну від традиційних алгоритмів кластеризації, де кожна точка даних призначається кластеру, метод поширення близькості зосереджується на виборі підмножини точок даних як зразків.

Алгоритм працює на основі матриці подібності, яка кількісно визначає подібність або відмінність між парами точок даних. Він використовує поняття «відповідальність» і «доступність» для ітеративного оновлення зразкових кандидатів. Відповідальність відображає накопичені докази того, що одна точка даних повинна вибрати іншу як свій взірець, тоді як доступність представляє накопичені докази того, що точка даних повинна бути обрана в якості зразка іншою.

Метод поширення близькості виконує ітерації передачі повідомлень між точками даних, оновлюючи матриці відповідальності та доступності, доки не буде виконано критерій конвергенції. Алгоритм динамічно визначає кількість кластерів і ідентифікує зразки, які найкраще представляють базову структуру кластера в даних.

Ключові переваги метод поширення близькості включають його здатність виявляти складні кластерні структури, не вимагаючи від користувача заздалегідь вказувати кількість кластерів. Однак важливо зазначити, що алгоритм чутливий до параметрів, і його обчислювальна складність може зробити його менш придатним для дуже великих наборів даних.

2.5.2 Зниження розмірності

Зниження розмірності — важливий крок у машинному навчанні та аналізі даних, який передбачає зменшення кількості функцій або змінних у наборі даних, зберігаючи при цьому важливу інформацію. Це особливо важливо при роботі з даними великої розмірності, де кількість ознак набагато більша, ніж кількість спостережень. Зменшення розмірності може допомогти пом’якшити проблему розмірності, підвищити ефективність обчислень і потенційно підвищити продуктивність моделі.

Робота з високовимірними наборами даних створює кілька практичних проблем для алгоритмів машинного навчання. Ці проблеми включають підвищений час обчислень, підвищені вимоги до пам’яті для керування великими наборами даних тощо. Однак однією з найбільш значних проблем є потенційне зниження точності при побудові прогнозних моделей. Моделі, навчені на високорозмірних наборах даних, часто мають проблеми з узагальненням, тобто вони можуть погано працювати на нових даних за межами навчальних даних.

Методи зниження розмірності спрямовані на пом’якшення негативного впливу, пов’язаного з даними великої розмірності, шляхом зменшення кількості функцій, зберігаючи відповідну інформацію. Ці методи поділяються на дві основні категорії: вибір ознак і вилучення ознак.

Метод головних компонентів — це статистичний метод, який використовується для зменшення розмірності та стиснення даних. Він передбачає перетворення набору корельованих змінних у новий набір некорельованих змінних, званих головними компонентами, які є лінійними комбінаціями вихідних змінних. Мета МГК полягає в тому, щоб охопити якомога більше мінливості даних за допомогою меншої кількості змінних, що полегшує аналіз та інтерпретацію

Ця аналітична техніка включає в себе принципи лінійної алгебри та матричних операцій. Він керує перетворенням оригінального набору даних у нову систему координат, сформовану цими основними компонентами. Основа цього перетворення лежить у власних векторах і власних значеннях, отриманих з коваріаційної матриці, що дозволяє комплексно досліджувати ці лінійні зміни.

Математично, МГК визначається для набору даних із n спостережень і p змінних. Нехай X — це матриця даних n×p, де кожен рядок представляє спостереження, а кожен стовпець — змінну.

Стандартизація даних:

$$Z = \frac{X - \bar{x}}{\sigma} \quad (2.5.2.1)$$

Де σ - стандартне відхилення.

Коваріаційна матриця для стандартизованих даних:

$$C = \frac{1}{n-1} Z^T Z \quad (2.5.2.2)$$

Розкладання власних значень:

$$Cv_i = \lambda_i v_i, \; i = 1, 2, \ldots, p \quad (2.5.2.3)$$

У МГК розраховуються два основні компоненти: перший головний компонент (ГК1) і другий головний компонент (ГК2).

Основний головний компонент (ГК1) позначає напрямок у просторі, який максимізує дисперсію точок даних. Це лінія, яка оптимально фіксує форму спроектованих точок. Величина мінливості, зафіксована першим компонентом, безпосередньо відповідає кількості інформації, збереженої з вихідного набору даних. Важливо, що жоден інший головний компонент не може демонструвати більшу дисперсію. Обчислення другого головного компонента (ГК2) відбувається за подібною процедурою, що й для ГК1. ГК2 представляє наступну найвищу дисперсію в наборі даних і не має бути корельованим з ГК1, тобто, ГК2 має бути ортогональним або

перпендикулярним ГК1. Ця ортогональність виражається математично як кореляція між ГК1 і ГК2, яка дорівнює нулю.

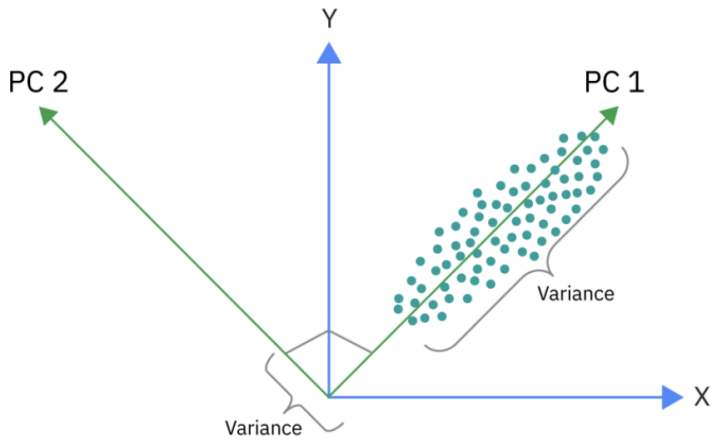

Рисунок 2.5.2.1 - Метод головних компонентів.[9]

Коли МГК використовується для набору даних, діаграма розсіювання часто використовується для ілюстрації зв’язку між ГК1 і ГК2. Осі ГК1 і ГК2 розташовані перпендикулярно одна одній, візуально підкреслюючи їх відсутність кореляції в перетвореній системі координат.

Ключовим обмеженням МГК є виняткове застосування лінійних перетворень. Аналіз основних компонентів ядра долає це обмеження, розширюючи МГК для врахування нелінійності. Аналіз основних компонентів ядра ініціюється шляхом відображення вихідних даних у нелінійний простір ознак, як правило, більшої розмірності. Згодом МГК застосовується для вилучення основних компонентів у цьому трансформованому просторі. Використання аналізу основних компонентів ядра полегшує розділення компонентів у просторі з більшою вимірністю, оскільки відображення в цьому просторі часто покращує можливості класифікації.

2.6 Висновки до розділу 2

У цій главі було розглянуто широкий спектр методів і алгоритмів машинного навчання, особливо зосередившись на їх застосуванні в

управлінні фінансовими ризиками. Ми почали з вичерпного огляду (2.1), щоб підготувати основу, наголошуючи на важливості розуміння основних принципів. Згодом були досліджені різноманітні завдання машинного навчання у фінансовому контексті (2.2), визнаючи його ключову роль у зниженні ризиків.

В основі розділу (2.3) представлені різні моделі та алгоритми, які забезпечують повне розуміння цієї області. Методи керованого навчання (2.4) були розібрані, охоплюючи лінійну регресію, регуляризовану регресію, логістичну регресію, класифікацію та дерева регресії, метод k-найближчих сусідів, метод опорних векторів і підхід випадкового лісу. Кожен метод був детально писаний, пропонуючи уявлення про сильні сторони та потенційні застосування кожного з них.

Переходячи до некерованого навчання (2.5), були досліджені кластерний аналіз і зниження розмірності. Ці методи, хоча й не керуються чітко позначеними даними, відкривають шляхи для розпізнавання шаблонів і виявлення аномалій, найважливіших аспектів у сфері управління фінансовими ризиками.

Цей розділ заклав міцну основу для подальшого дослідження конкретних застосувань у фінансовому контексті. Набір розглянутих методів і алгоритмів надає повний інструментарій, сприяючи глибшому розумінню нюансів стратегій, що застосовуються в області аналізу фінансового ризику.

РОЗДІЛ 3. АНАЛІЗ РОБОТИ ТА ОЦІНЮВАННЯ ЯКОСТІ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ФІНАНСОВИХ РИЗИКІВ

3.1 Фактори вибору моделі

Під час оцінювання моделей машинного навчання не можна застосувати жодне єдине рішення чи універсальний підхід. На вибір відповідної моделі впливають різні фактори, але продуктивність моделі є ключовим критерієм. Однак, поряд із продуктивністю, на процес вибору моделі можуть вплинути численні інші критерії.

Процес вибору моделі включає в себе такі фактори:

- *Простота* - ступень простоти моделі, яка призводить до швидших, більш масштабованих і зрозумілих моделей і результатів.

- *Час навчання* - швидкість, продуктивність, використання пам’яті та загальний час, необхідний для навчання моделі.

- *Обробка нелінійності даних* - оцінка здатності моделі керувати нелінійними зв’язками між змінними.

- *Стійкість до перенавчання* - оцінка здатності моделі справлятися з перенавчання.

- *Розмір набору даних* - вимір спроможності моделі до обробки великої кількості навчальних прикладів у наборі даних.

- *Кількість функцій* - здатність моделі працювати з великою розмірністю простору функцій.

- *Інтерпретація моделі* - інтерпретованість моделі є вирішальною для вжиття конкретних дій для вирішення основної проблеми.

- *Масштабування функцій* - розглядає модель масштабування змінних або дотримання нормального розподілу.

Такі моделі, як лінійна та логістична регресія, є простішими, тоді як складність зростає з методами ансамблевого навчання та штучними

нейронними мережами. Що стосується часу навчання, лінійні моделі та дерева класифікації та регресії демонструють відносно швидке навчання порівняно з методами ансамблевого навчання .

Моделі лінійної та логістичної регресії обмежені в обробці нелінійних зв'язків. Навпаки, усі інші моделі, можуть розглядати нелінійні зв'язки, використовуючи нелінійні ядра для взаємодії між залежними та незалежними змінними.

Метод випадковий ліс демонструє меншу чутливість до перенавчання порівняно з лінійною регресією, логістичною регресією. На ступінь перенавчання впливають такі параметри, як розмір даних і налаштування моделі, і його можна оцінити, вивчивши результати тестового набору для кожної моделі. Методи підсилення, такі як градієнтне підсилення, створюють вищий ризик перенавчання порівняно з методами пакетування, такими як випадковий ліс.

При визначенні пріоритету прогностичної продуктивності без необхідності розуміння внутрішньої роботи моделі краще використовувати моделі з меншою інтерпретованістю. Тим не менш, є випадки, коли можливість інтерпретації є вирішальною, наприклад, у фінансовій галузі. Наприклад, алгоритм машинного навчання використовується для схвалення або відхилення заявки фізичної особи на кредитування. Якщо заявник стикається з відмовою та оскаржує рішення, фінансова установа повинна мати можливість пояснити причину цього. Хоча це завдання може бути майже неможливим для штучних нейронних мереж, моделі на основі дерева рішень пропонують більш просте пояснення.

3.2 Способи оцінки якості моделі

Ключовими компонентами оцінки ефективності моделі, є перенавчання, перехресна перевірка та метрики оцінювання.

Важливість метрик, які використовуються для оцінки алгоритмів машинного навчання, важко переоцінити. Вибір метрики відіграє ключову роль у формуванні вимірювання та порівняння продуктивності застосованих алгоритмів. Ці метрики не лише впливають на те, як призначається важливість різним характеристикам у результатах, але й відіграють вирішальну роль у визначенні алгоритму, який найкраще відповідає цілям поставленої задачі.

Основні метрики оцінювання для регресії:

- Середня абсолютна похибка.
- Середньоквадратична похибка.
- Коефіцієнт детермінації.
- Скоригований коефіцієнт детермінації.

Основні метрики оцінювання для класифікації:

- Точність.
- Правильність.
- Чутливість.
- Площа під кривою.
- Матриця невідповідностей.

Середня абсолютна похибка - це показник, який вимірює середню абсолютну різницю між фактичними та прогнозованими значеннями. Він забезпечує простий і легкий для розуміння показник точності моделі.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y}_i| \qquad (3.2.1)$$

Де n – кількість спостережень, y_i – фактичне значення, $\widehat{y}_i$ – прогнозоване значення.

Нижча середня абсолютна похибка вказує на кращу продуктивність, при цьому 0 є ідеальним показником. Він менш чутливий до викидів порівняно з середньоквадратичною похибкою.

Середньоквадратична *похибка* - це показник, який вимірює середнє значення квадратів різниць між фактичними та прогнозованими значеннями. Він карає більші помилки суворіше, ніж менші.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3.2.2)$$

Подібно до середньої абсолютної похибки, нижча середньоквадратична похибка вказує на кращу продуктивність. Однак середньоквадратична похибка, як правило, більш чутлива до викидів через квадрат помилок.

R^2, також відомий як коефіцієнт детермінації, вимірює частку дисперсії залежної змінної, яку можна передбачити на основі незалежних змінних. Він коливається від 0 до 1, де 0 вказує на те, що модель не пояснює жодної дисперсії, а 1 вказує на ідеальний прогноз.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (3.2.3)$$

Де $\bar{y}$ - є середнім фактичних значень.

Більше значення R^2 вказує на кращу відповідність моделі даним. Він представляє частку дисперсії в залежній змінній, яку пояснює модель.

Скоригований R^2 - це модифікована версія R^2, яка враховує кількість предикторів у моделі. Цей метод карає за додавання нерелевантних предикторів, які не покращують продуктивність моделі.

$$R^2 = 1 - \frac{(1-R^2)(n-1)}{n-k-1} \quad (3.2.4)$$

Де n – кількість спостережень, k – кількість предикторів.

Подібно до R^2, більш високе скориговане значення R^2 вказує на кращу відповідність моделі. Однак це більш консервативний захід, коли мова йде про використання моделей з великою кількістю предикторів, допомагаючи запобігти перенавчанню.

Середня абсолютна похибка та середньоквадратична похибка зазвичай використовуються для завдань регресії, тоді як R^2 і скоригований R^2 широко використовуються для оцінки відповідності в моделях лінійної регресії. Вибір метрики залежить від конкретних характеристик проблеми та бажаного балансу між інтерпретабельністю та чутливістю до різних типів помилок.

Точність є одним із найпростіших показників і представляє відношення правильно передбачених випадків до загальної кількості випадків.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (3.2.5)$$

Точність легко зрозуміти та інтерпретувати. Однак це може бути не найкращий показник у ситуаціях, коли класи незбалансовані.

Правильність – це відношення правильно передбачених позитивних спостережень до загальної кількості прогнозованих позитивних результатів. Цей показник зосереджений на точності позитивних прогнозів.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (3.2.6)$$

Правильність цінна, коли вартість помилкових спрацьовувань висока. Показник чутливий до кількості помилкових спрацьовувань і використовується для оцінки здатності моделі уникати неправильної класифікації негативів як позитивних.

Чутливість - це відношення правильно передбачених позитивних спостережень до загальної кількості фактичних позитивних результатів. Він вимірює здатність моделі охоплювати всі відповідні випадки.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (3.2.7)$$

Показник чутливості корисний, коли вартість хибно негативних результатів висока. Він чутливий до кількості помилкових негативів і

використовується для оцінки здатності моделі ідентифікувати всі позитивні випадки.

Площа під кривою — це графічне представлення компромісу між істинно позитивною швидкістю та помилковою позитивною частотою за різних порогів. Площа під кривою надає одне скалярне значення, яке підсумовує продуктивність класифікатора. Підхід особливо корисний при роботі з незбалансованими наборами даних. Вища площа під кривою вказує на кращу дискримінацію між позитивними та негативними класами.

Матриця невідповідностей — це таблиця, яка підсумовує продуктивність алгоритму класифікації. Він показує кількість істинно позитивних, істинно негативних, хибнопозитивних і хибнонегативних результатів.

Компоненти:

- Істинно позитивні - випадки, коли модель правильно передбачає позитивний клас.
- Істинно негативні - випадки, коли модель правильно передбачає негативний клас.
- Хибнопозитивні - випадки, коли модель неправильно передбачає позитивний клас.
- Хибнонегативні - випадки, коли модель неправильно передбачає негативний клас.

Матриця невідповідностей надає детальну розбивку продуктивності моделі, дозволяючи глибше зрозуміти типи помилок, які вона робить.

Баланс між надмірним і недостатнім навчанням має вирішальне значення для створення моделей, які добре узагальнюють невидимі дані. Такі методи, як регуляризація, перехресна перевірка та ретельний вибір складності моделі, можуть допомогти досягти цього балансу в моделях машинного навчання.

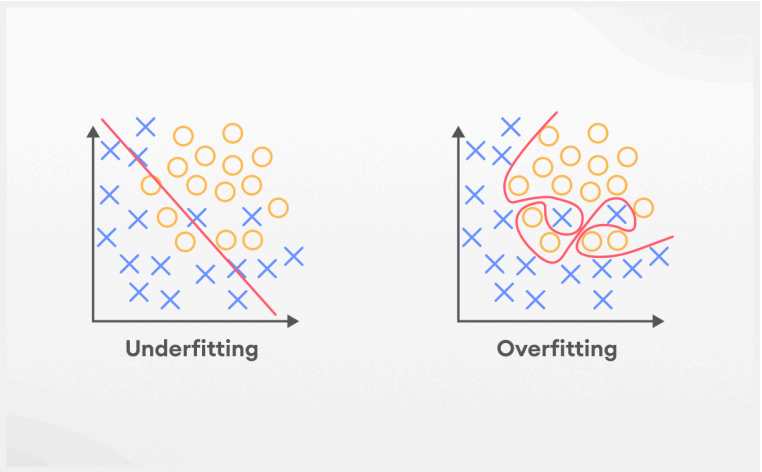

Рисунок 3.2.1 - Недостатнє навчання та перенавчання.[10]

Перенавчання відбувається, коли модель машинного навчання надто добре вивчає навчальні дані, вловлюючи шум або випадкові коливання, які можуть не відображати базові моделі даних. Як наслідок, модель добре працює на навчальному наборі, але погано на нових, невидимих даних.

Причини:

- Використання складної моделі, яка є занадто гнучкою та вловлює шум.

- Недостатньо даних для складності моделі.

- Відсутність регуляризації, що дозволяє моделі адаптувати шум у даних.

Ознаки переобладнання:

- Висока точність на навчальному наборі, але низька точність на перевірочному або тестовому наборі.

- Модель добре працює на конкретних прикладах, але провалюється на нових, різноманітних прикладах.

Для того щоб уникнути перенавчання ,треба використовувати простіші моделі або зменшувати складність існуючих моделей. Також може допомогти збільшення обсягу навчальних даних та використання перехресної перевірки для оцінки ефективності моделі.

Недостатнє навчання виникає, коли модель машинного навчання занадто проста, щоб охопити базові закономірності в навчальних даних. Як наслідок, модель погано працює як на навчальному наборі, так і на нових, невидимих даних.

Причини:

- Використання моделі, яка є занадто простою для складності даних.

- Недостатньо навчальних даних для навчання більш складної моделі.

- Неправильний вибір або розробка функцій.

Ознаки недостатнього навчання:

- Низька точність як для навчання, так і для перевірки/тесту.

- Модель не може вловити базові шаблони в даних.

Для профілактики треба використовувати більш складні моделі з більшою потужністю та збільшити обсяги навчальних даних.

Вирішення проблеми тренування моделей машинного навчання для ефективного узагальнення невидимих даних передбачає баланс між перенавчанням і недостатнім навчанням, інкапсульованим у компромісі зміщення та дисперсії. Перехресна перевірка стає цінним методом вирішення цієї проблеми.

Фундаментальною концепцією перехресної перевірки є систематичне розділення набору даних, використовуючи кожне розділення як набір перевірки принаймні один раз, а решту використовують для навчання. Це передбачає навчання алгоритму на частині даних (навчальна вибірка), а потім оцінку його продуктивності на решті частини (вибірка перевірки), щоб оцінити ризик алгоритму. Перехресна перевірка служить засобом отримання надійних оцінок помилки узагальнення моделі.

Наприклад, у k-кратній перехресній перевірці навчальні дані випадковим чином поділяються на k-кратні згортки. Модель навчається за допомогою k-1 згорток, і її продуктивність оцінюється на k-му згортанні.

Цей процес повторюється k разів, а отримані бали усереднюються, щоб забезпечити повну оцінку ефективності.

Хоча перехресна перевірка дає цінну інформацію про продуктивність моделі, вона може супроводжуватися обчислювальними витратами, особливо в поєднанні з налаштуванням гіперпараметрів за допомогою пошуку в сітці.

3.3 Постановка задачі для моделювання

Кредитування є важливою частиною фінансової індустрії, і кредитори надають позики позичальникам в очікуванні погашення з відсотками. Прибутковість для кредиторів залежить від успішного погашення кредиту позичальником, що робить обов’язковим вирішення двох основних питань у секторі кредитування:

- Який рівень ризику пов'язаний з позичальником?
- Враховуючи профіль ризику позичальника, чи слід надавати кредит?

Машинне навчання виявляється особливо корисним у прогнозуванні, представляючи ідеальне рішення завдяки його здатності навчатися на величезній кількості даних. Автоматизовані завдання, такі як зіставлення записів даних, ідентифікація винятків і оцінка прийнятності позики, можуть ефективно оброблятися за допомогою алгоритмів. Ці алгоритми можуть заглиблюватися в базові тенденції, забезпечуючи безперервний аналіз для виявлення факторів, які можуть вплинути на майбутні ризики кредитування.

Реальні сценарії, включно з моделюванням дефолту за кредитом, часто передбачають роботу з недосконалими даними. Такі проблеми, як відсутні значення, неповні категоричні дані та нерелевантні функції, є звичним явищем. Хоча важливість очищення даних не завжди

підкреслюється, вона має вирішальне значення для результату роботи моделі машинного навчання. Незважаючи на те, що алгоритми мають значну потужність, їхня ефективність залежить від відповідних даних. Таким чином, для моделювання необхідна комплексна підготовка та очищення даних. Щоб забезпечити очищення та організацію простору ознак, будуть використані різні методи, що охоплюють обробку даних, вибір ознак і пошуковий аналіз.

Основна мета цього моделювання - це побудова моделі машинного навчання, здатної передбачити ймовірність непогашення кредиту.

Увага буде зосереджена на таких єтапах:

- Підготовка даних, очищення даних і робота з великою кількістю ознак.
- Дискретизація даних і обробка категоріальних даних.
- Вибір ознак і перетворення даних.

3.4 Аналіз та підготовка вхідних даних для моделювання

У класифікаційній системі для цього випадку прогнозованою змінною є списання, борг, який кредитор відмовився від спроб стягнути після того, як позичальник пропустив платежі протягом кількох місяців.

Прогнозована змінна приймає значення 1 у разі списання та значення 0 в іншому випадку.

Будуть проаналізовані дані про кредити з 2007 по 4 квартал 2018 року від американської компанії, яка надає кредити. Набір даних містить 100 тисяч спостережень із 151 ознакою, що містять повні дані про позики для усіх позик, наданих за вказаний період часу. Ознаки включають дохід, вік, кредитні рейтинги, володіння будинком, місцезнаходження позичальника, колекції та багато інших. Буде досліджено 151 ознаку прогнозів для вибору необхідних ознак.

3.4.1 Аналіз даних для моделювання

Початковий етап аналітичного прогнозування включає підготовку та аналіз даних, і цей етап зазвичай є найбільш трудомістким. У зібраних даних можуть бути відсутні значення, помилки або невідповідності в типах або форматах. Крім того, він може бути шумним, неповним або містити зайву інформацію, що може бути складним для аналізу. Деякі дані можуть бути нерелевантними та можуть ускладнювати процес прогнозування, не додаючи цінності. Вкрай важливо вирішити ці проблеми, очистивши та впорядкувавши дані, щоб переконатися, що вони точні та послідовні, перш ніж переходити до етапу моделювання. Ця превентивна дія допомагає запобігти впливу будь-яких недоліків на продуктивність моделі.

Для роботи за даними будуть використані такі бібліотеки ,як :pandas, seaborn.

Датасет має 151 ознаку та 100 тисяч записів. Кількість ознак велика і буде зменшена в подальшому дослідженні. Загальний вигляд датасету (3.4.1.1):

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...	hardsh
0	68407277	NaN	3600.0	3600.0	3600.0	36 months	13.99	123.03	C	C4	...	
1	68355089	NaN	24700.0	24700.0	24700.0	36 months	11.99	820.28	C	C1	...	
2	68341763	NaN	20000.0	20000.0	20000.0	60 months	10.78	432.66	B	B4	...	
3	66310712	NaN	35000.0	35000.0	35000.0	60 months	14.85	829.90	C	C5	...	
4	68476807	NaN	10400.0	10400.0	10400.0	60 months	22.45	289.91	F	F1	...	

5 rows × 151 columns

Рисунок 3.4.1.1 - Загальний вигляд датасету.

Розглянемо окремі ознаки які можуть мати значення для результатів прогнозування.

Ознака «term» означає кількість платежів за кредитом, виражену в місяцях, з варіантами 36 або 60 місяців. Спостерігається, що кредити терміном на 60 місяців мають вищу ймовірність списання. Щоб полегшити

подальший аналіз, перетворимо значення термінів на цілі числа та згрупуємо їх відповідно (рис. 3.4.1.2).

```
term
36    0.171664
60    0.355100
Name: proportion, dtype: float64
```

Рисунок 3.4.1.2 - Розподіл термінів відносно відсотку списання у датасеті.

Позики, що охоплюють п'ятирічні періоди, демонструють ймовірність списання, яка більш ніж вдвічі вища, ніж позики з трирічними періодами. Ця особливість може мати значення для цілей прогнозування.

Розглянемо ознаку «emp_length» ,показник який відповідає за водображення терміну працевлаштування ,відносно списання(рис. 3.4.1.3):

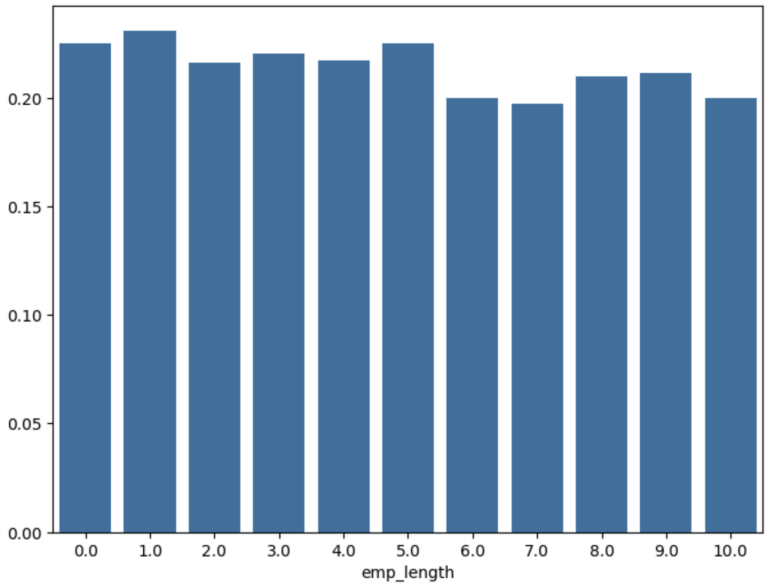

Рисунок 3.4.1.3 - Діаграма терміну працевлаштування відносно відсотку списання.

Статус кредиту не сильно залежить від тривалості роботи. Хоча тривалість роботи може дати певне уявлення про стабільність позичальника, вона не є остаточним показником його кредитоспроможності чи здатності погасити позику. Замість цього кредитори зосереджуються на ширшому наборі критеріїв для оцінки ризику, включаючи кредитну історію, рівень доходу, співвідношення боргу

до доходу та історію платежів. Цей багатогранний підхід дозволяє кредиторам більш комплексно оцінювати позичальників, забезпечуючи точнішу оцінку їх профілю ризику.

Розглянемо наступну ознаку кредитного рівня позичальника - «sub_grade» відносно списання(рис. 3.4.1.4):

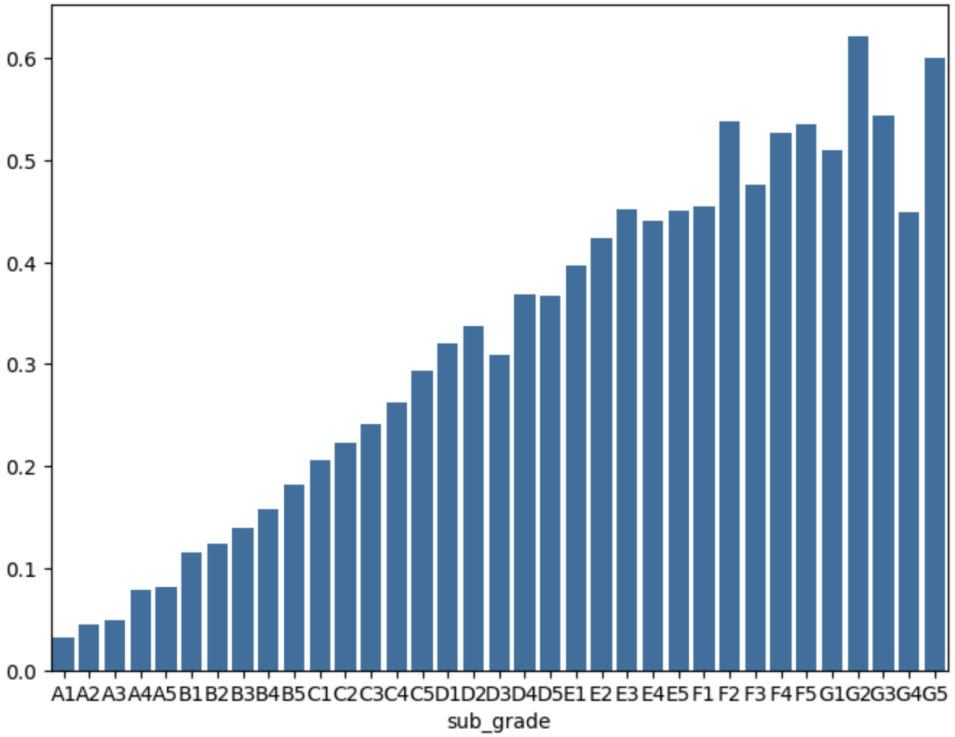

Рисунок 3.4.1.4 - Діаграма «sub_grade».

Діаграма ілюструє чітку тенденцію, яка вказує на те, що ймовірність списання зростає в міру погіршення «sub_grade». Цей зв’язок свідчить про те, що кредити з нижчими субкласами, які часто відображають вищий профіль ризику, мають більшу ймовірність зіткнутися з фінансовими проблемами, які призведуть до списання. Зі зниженням субкласу кредитна якість позик зазвичай знижується, що вказує на вищий ризик дефолту, прострочення або інших несприятливих подій, які вимагають списання. Отже, ця закономірність визнається ключовою ознакою в аналізі.

Розглянемо річний дохід позичальників ,ознаку «year_inc» (рис. 3.4.1.5):

	annual_inc
count	5.744300e+04
mean	7.666708e+04
std	5.751163e+04
min	0.000000e+00
25%	4.600000e+04
50%	6.500000e+04
75%	9.100000e+04
max	4.462056e+06

Рисунок 3.4.1.5 - Данні річного доходу.

Річний дохід коливається від 0 до 9 000 000 доларів із середнім показником 65 000 доларів.

Розглянемо функцію оцінки FICO («fico_range_low», «fico_range_high») (рис. 3.4.1.6). *Оцінка FICO* – це кредитна оцінка. Оцінки FICO враховують дані в п’яти сферах для визначення кредитоспроможності позичальника: історія платежів, поточний рівень заборгованості, типи використаного кредиту, тривалість кредитної історії та нові кредитні рахунки.

	fico_range_low	fico_range_high
fico_range_low	1.0	1.0
fico_range_high	1.0	1.0

Рисунок 3.4.1.6 - Кореляція ознак FICO.

Оскільки кореляція між низьким і високим показниками FICO дорівнює 1, доцільно зберегти лише одну ознаку.

Розглянемо специфіку очікуваної ознаки. Прогнозована ознака має бути взята зі стовпця «loan_status».

З документації визначення даних(рис. 3.4.1.7):

- *"Повністю сплачено"* - позики, які були повністю погашені.

- *"За замовчуванням*" - позики, які не були поточними протягом 121 дня або більше.

- *"Списані*" - позики, за якими більше немає обґрунтованого очікування подальших платежів.

```
loan_status
Fully Paid            45023
Current               40913
Charged Off           12420
Late (31-120 days)     1047
In Grace Period         383
Late (16-30 days)       212
Default                   2
Name: count, dtype: int64
```

Рисунок 3.4.1.7 - Розподілення за статусом позики.

Значна частина спостережень наразі має статус «Поточний», і майбутні результати цих справ — чи будуть вони «Стягнені», «Повністю сплачені» чи «За замовчуванням» — залишаються невизначеними. Випадки, позначені як «За замовчуванням», є мінімальними порівняно з «Повністю оплачено» або «Стягнено», і тому виключаються з розгляду.

Приблизно 78% непогашених кредитів було успішно повністю виплачено, тоді як решта 21% були списані.

```
loan_status
Fully Paid     0.783786
Charged Off    0.216214
Name: proportion, dtype: float64
```

Рисунок 3.4.1.8 - Відсоткове співвідношення «Повністю оплачено» та «Стягнено».

На наступному кроці додаємо кілька ознак, які допоможуть у подальшому розгляді та аналізі набору даних. Перші дві важливі функції, які ми додамо, будуть текстовим і цифровим відображенням розподілу поганих і хороших кредитів. Погані ознаки включатимуть всі ознаки зі статусом «Стягнено», «За замовчуванням», «У пільговому періоді», «Прострочено (16-30 днів)» і «Прострочено (31-120 днів)». У числовому варіанті погані ознаки позначатимуться як 1, а хороші – 0 відповідно.

Крім того, будуть додані інші корисні ознаки в наш набір даних, щоб покращити наш аналіз. Тривалість зайнятості (термін зайнятості) буде перетворено в числове значення, що забезпечить вимірний показник стабільності роботи. Ця трансформація може допомогти оцінити зв’язок між стабільністю зайнятості та статусом кредиту.

Крім того, регіон позичальника буде класифіковано на основі його штату адреси. Відображення кожного штату в ширшому регіоні дозволяє нам визначити географічні тенденції в ефективності позики, що може мати вирішальне значення для аналізу демографічних моделей і розуміння того, як регіональні економічні фактори впливають на результати позики.

Використовуючи ці нові ознаки, ми створюємо багатший набір даних із ширшим контекстом для аналізу позик. Це дозволить нам виявити уявлення про те, які ознаки найбільше пов’язані з поганими чи хорошими кредитами, допомагаючи керувати майбутнім моделюванням і прогнозною аналітикою. Ця додаткова інформація також може покращити управління кредитним портфелем, дозволяючи кредиторам приймати більш обґрунтовані рішення щодо оцінки ризиків і видачі позики.

	65925	144128	170806	212009	686119
id	65057695	60316036	58051403	55481904	79084266
member_id	NaN	NaN	NaN	NaN	NaN
loan_amnt	10000.0	21575.0	28000.0	19200.0	9000.0
funded_amnt	10000.0	21575.0	28000.0	19200.0	9000.0
funded_amnt_inv	10000.0	21575.0	28000.0	19200.0	9000.0
...	...	...	...	...	...
settlement_term	NaN	NaN	NaN	NaN	NaN
loan_condition_int	0	1	1	1	0
loan_condition	Good Loan	Bad Loan	Bad Loan	Bad Loan	Good Loan
emp_length_int	1.0	10.0	10.0	6.0	10.0
region	West	West	SouthWest	SouthEast	SouthWest

155 rows × 5 columns

Рисунок 3.4.1.9 - Результуючий вигляд датасету.

Розглянемо розподіл за новою ознакою «loan_condition» (рис. 3.4.1.10):

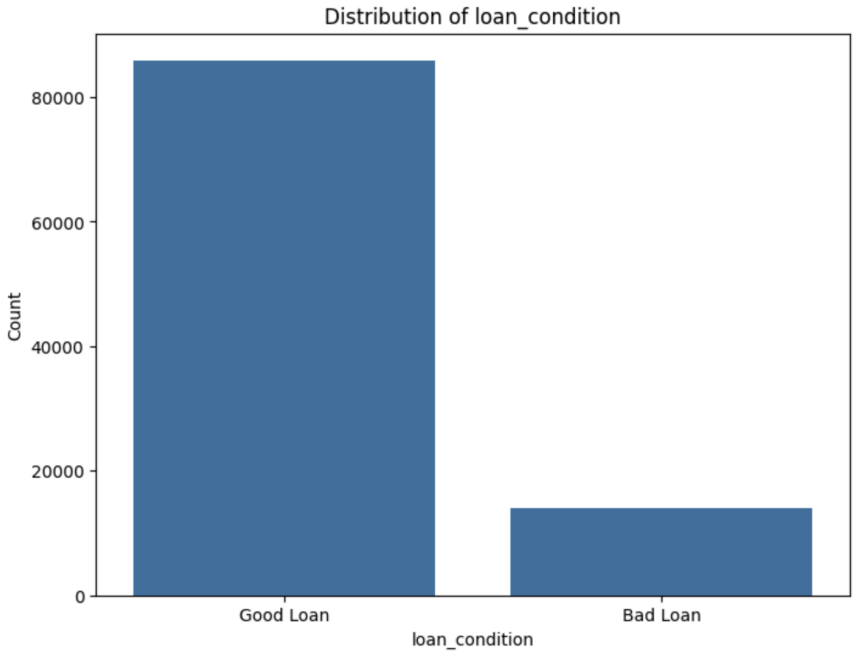

Рисунок 3.4.1.10 - Розподіл «loan_condition».

Як ми бачимо, кількість хороших кредитів значно перевищує кількість поганих приблизно у співвідношенні 8 до 1 відповідно. Ця значна розбіжність свідчить про те, що в межах набору даних більшість позик працюють добре, що вказує на загалом здоровий кредитний портфель. Високе співвідношення хороших і поганих кредитів відображає сприятливу тенденцію та підкреслює, що більшість позичальників виконують свої зобов’язання відповідно до очікувань.

Однак ця статистика також має значення для аналізу ризиків і прогнозного моделювання. З огляду на те, що безнадійні кредити становлять меншу частку набору даних, під час створення прогнозних моделей можуть виникнути проблеми, пов’язані з дисбалансом класів. Цей дисбаланс може призвести до перенавчання основного класу, потенційно вплинувши на здатність моделі точно ідентифікувати безнадійні кредити.

Давайте розглянемо зв’язок між ознаками «total_rec_prncp» і «loan_condition», які ми додали до нашого набору даних. «total_rec_prncp»

(Загальна отримана основна сума) стосується основної суми, яка була погашена на даний момент. Цей показник дає уявлення про те, яку частину початкової позики було повернуто позичальниками з часом.

Враховуючи те, що «loan_condition» є категоричним, показуючи, чи позика вважається «хорошою позикою» чи «поганою позикою», а «total_rec_prncp» є числовою ознакою, скрипки діаграма, може бути використана для ілюстрації розподілу основної суми, сплаченої за різними умовами позики.

Скрипковий графік корисний, оскільки він показує розподіл безперервної змінної між різними категоріями, пропонуючи розуміння діапазону даних, медіани та потенційних шаблонів або аномалій. На графіку відображатиметься дзвоноподібна форма для кожної категорії, а ширина вказуватиме на щільність точок даних у певному значенні. Ця візуалізація може допомогти нам зрозуміти, чим відрізняються схеми погашення хороших і поганих кредитів.

Рисунок 3.4.1.11 - Розподіл між ознаками «total_rec_prncp» і «loan_condition».

Візуалізувавши цей зв’язок, ми можемо отримати глибше розуміння поведінки погашення кредиту. Використовуючи скрипковий графік, ми

побачили, що хороші позики, як правило, мають вищі виплати основної суми порівняно з проблемними позиками.

В останньому прикладі ми розглянемо розподіл трьох ознак, пов’язаних із позикою: «grade», що представляє рівень ризику позики, «term», що вказує на тривалість позики, і «loan_amnt», що представляє суму позики. Враховуючи, що «grade» і «term» є категоріальними змінними, тоді як «loan_amnt» є числовою змінною, ми знову можемо використати скрипковий графік, щоб проілюструвати розподіл сум позики між різними комбінаціями оцінок і умов .

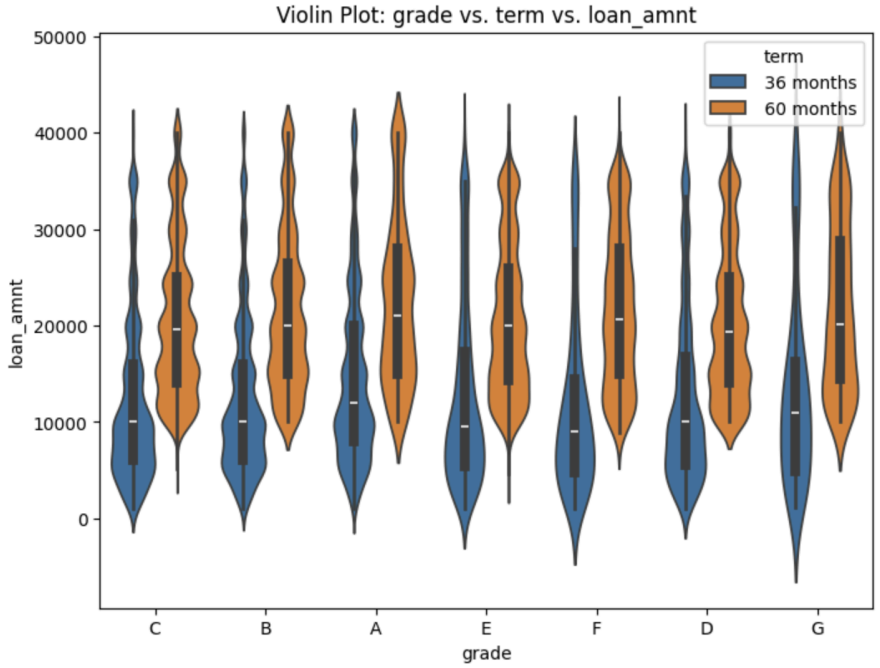

Рисунок 3.4.1.11 - Розподіл між ознаками «grade», «term», «loan_amnt».

Можемо помітити, що вищі рейтинги позики (вказують на нижчий ризик) пов’язані з вищими сумами позики або що триваліші терміни позики мають іншу модель розподілу порівняно з коротшими.

3.4.2 Підготовка даних для моделювання

В цій частині буде розглянуто: видалення виключень, врахування відсутнього значення, видалення викидів.

Видалення виключень - цей крок передбачає ідентифікацію та видалення будь-яких точок даних або записів, які не відповідають конкретним критеріям або вважаються невідповідними для аналізу. Винятки можуть ґрунтуватися на різних факторах, таких як невідповідні дані, неправильні чи нерелевантні записи або значення, які виходять за межі визначеного діапазону. Усунувши ці винятки, ми можемо забезпечити більш чистий і послідовний набір даних для аналізу.

Врахування відсутнього значення - цей процес усуває будь-які відсутні або нульові значення в наборі даних. Імпутація може включати різні методи, такі як заміна відсутніх значень середнім, медіаною чи модою, або використання більш просунутих методів, таких як інтерполяція чи прогнозне моделювання. Мета полягає в тому, щоб підтримувати цілісність набору даних і уникати потенційних упереджень або неточностей, які можуть внести відсутні значення під час аналізу.

Викиди – це екстремальні значення, які значно відрізняються від інших спостережень у наборі даних. Вони можуть бути результатом помилок у зборі даних, незвичайних подій або внутрішньої мінливості. Цей крок спрямований на виявлення та видалення цих викидів, щоб запобігти спотворенню статистичних аналізів або прогнозних моделей.

Спочатку розглянемо які типи даних присутні в нашому датасеті (рис. 3.4.2.1):

```
id                    object
member_id            float64
loan_amnt            float64
funded_amnt          float64
funded_amnt_inv      float64
                      ...
settlement_term      float64
loan_condition_int     int64
loan_condition        object
emp_length_int       float64
region                object
Length: 155, dtype: object
```

Рисунок 3.4.2.1 - Типи даних датасету.

Знання типів даних у наборі даних має вирішальне значення, оскільки воно керує процесами аналізу та попередньої обробки даних. Різні типи даних вимагають різних підходів до очищення, перетворення та аналізу інформації. Загалом знання типів даних у наборі даних забезпечує міцну основу для надійного й точного процесу аналізу даних.

Початковий крок у процесі обробки даних передбачає перевизначення класифікації хороших і поганих кредитів. Щоб покращити ясність, нам потрібно виключити позики, позначені як «Поточні», оскільки ця класифікація не вказує чітко, чи вважаються позики хорошими чи поганими. Це видалення допоможе усунути неоднозначність у наборі даних, дозволяючи нам працювати з більш визначеним набором категорій кредитів. Без цього виключення наш аналіз міг би ввести в оману через незрозумілу природу «Поточні».

	loan_status	count
0	Fully Paid	45023
1	Current	40913
2	Charged Off	12420
3	Late (31-120 days)	1047
4	In Grace Period	383
5	Late (16-30 days)	212
6	Default	2

Рисунок 3.4.2.2 - Частка кредитів зі статусом «Поточні.

Наступний крок: деякі ознаки у наборі даних можуть містити значну кількість відсутніх даних. Ці ознаки не є важливими для нашого аналізу та можуть призвести до неточностей або неефективності. Цей крок видалить усі стовпці, у яких відсутні 20% або більше даних. Іншими словами, буде збережено лише стовпці з принаймні 80% ненульових значень відносно загальної кількості рядків. Цей підхід допомагає оптимізувати набір даних,

гарантуючи, що ми працюємо з надійними даними, які відповідають мінімальному порогу повноти.

	Missing Count	Missing Percent	Type
member_id	59087	100.000000	float64
desc	59079	99.986461	object
orig_projected_additional_accrued_interest	58781	99.482120	float64
hardship_dpd	58654	99.267182	float64
payment_plan_start_date	58654	99.267182	object
...	...	...	...
title	605	1.023914	object
mths_since_recent_bc	592	1.001912	float64
last_pymnt_d	105	0.177704	object
revol_util	37	0.062620	float64
dti	30	0.050773	float64

72 rows × 3 columns

Рисунок 3.4.2.3 - Кількість і відсоток значень, які відсутні.

Як можно побачити з таблиці (рис. 3.4.2.3) деякі ознаки повністю, або майже повністю відсутні. Видалення цих ознак полегшить подальшу розробку моделей та знизить кількість необхідних обчислювальних потужностей. Це гарантує, що ви залишитеся з більш чистим, керованим набором даних, який підтримує ефективний аналіз і моделювання.

	Missing Count	Missing Percent	Type
mths_since_recent_bc_dlq	44260	74.906494	float64
mths_since_last_major_derog	41989	71.063009	float64
il_util	38988	65.984057	float64
mths_since_recent_revol_delinq	38020	64.345795	float64
mths_since_rcnt_il	36512	61.793626	float64
...	...	...	...
title	605	1.023914	object
mths_since_recent_bc	592	1.001912	float64
last_pymnt_d	105	0.177704	object
revol_util	37	0.062620	float64
dti	30	0.050773	float64

32 rows × 3 columns

Рисунок 3.4.2.4 - Датасет після видалення малозаповнених стовпців.

Деякі ознаки в наборі даних безпосередньо вказують на результат умови позики. Наприклад, «total_rec_prncp» (Загальна сума основної суми,

отримана на сьогоднішній день) — це змінна, яка показує, чи було повністю погашено кредит. Якщо це значення збігається з «loan_amnt» (загальна сума позики), це вказує на те, що позика є «гарною позикою». Ця залежність візуально представлена на діаграмі (рис. 3.4.2.5):

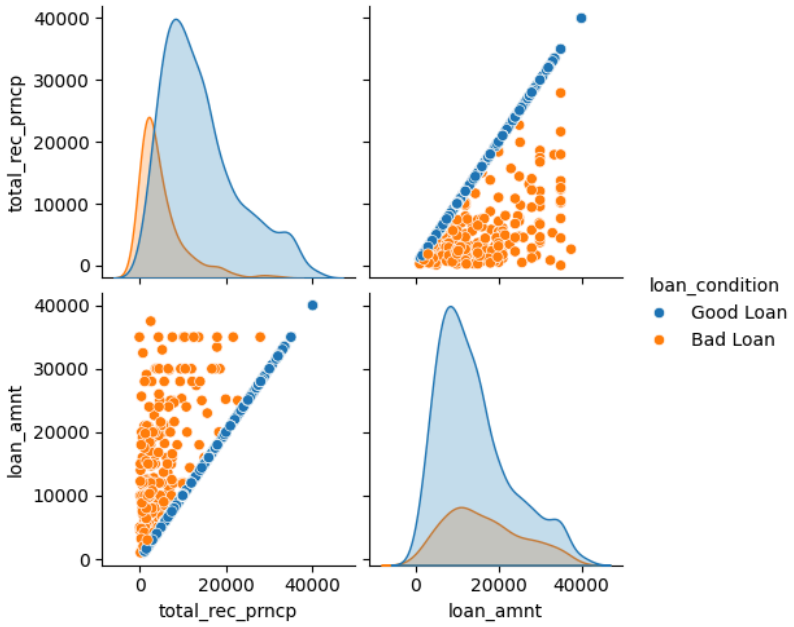

Рисунок 3.4.2.5 - Залежність «total_rec_prncp» та «loan_amnt».

Враховуючи те, що метою є прогнозування результатів позики протягом періоду погашення, важливо виключити змінні, які змінюються з часом або залежать від подій після закінчення терміну позики. Цей крок гарантує, що ми працюємо зі стабільним набором прогнозів, які точно відображають фактори, що впливають на умови позики.

Зосереджуючись лише на змінних, які безпосередньо впливають на результати позики, ми уникаємо упереджень або залежності від майбутніх подій. Цей підхід допомагає створити більш надійну прогностичну модель і гарантує, що змінні, які використовуються в аналізі, справді представляють чинники, що сприяють успіху або невдачі позики.

На останньому етапі цього етапу ми маємо справу з ідентифікаторами «id», назва роботи «title» та посадами «emp_title» — атрибутами, які, як правило, є унікальними для кожного запису та

непридатні для цілей моделювання. Назви роботи ознака «title» та посади ознака «emp_title» часто мають високу мінливість із занадто великою кількістю унікальних значень, щоб бути практичними для моделювання. Незважаючи на те, що професія та посада потенційно можуть надати інформацію про неплатежі по кредитах, більша частина цієї інформації, ймовірно, уже міститься в підтверджених доходах позичальника.

Крім того, будь-яка корисна інформація з цих ознак вимагала б значної додаткової обробки, такої як стандартизація або групування подібних назв для зменшення варіативності. Ця додаткова робота виходить за рамки цього прикладу, але може бути досліджена в майбутніх ітераціях моделі.

Географічна інформація, як-от поштові індекси «zip_code», може сприяти розумінню кредитного ризику, надаючи детальну інформацію на основі місцезнаходження. Однак підготовка цих даних для моделювання потребує додаткових кроків, таких як категоризація або анонімізація, щоб уникнути проблем із конфіденційністю. Зважаючи на необхідний рівень зусиль, цей аспект також розглядався поза межами поточного аналізу.

Вищезгадані ознаки «id», «title», «emp_title» та «zip_code» будуть видалені.

На наступному кроці буде виконане імпутування відсутніх значень. Спочатку обробляємо відсутні значення у датафреймі, заповнюючи відсутні дані відповідними замінами, залежно від того, чи є змінні категоріальними (типу об’єкт) чи числовими.

Для таких категоріальних змінних, як «last_pymnt_d» (дата останнього платежу) і «last_credit_pull_d» (дата останнього отримання кредиту), заповнюємо відсутні значення на основі найчастішого значення у кожному регіоні.

Для таких числових змінних, як «pub_rec» (кількість публічних записів) і «total_acc» (загальна кількість кредитних ліній), відсутні значення будуть заповнені середнім значенням у кожному регіоні.

Подібним чином "emp_length_int" (тривалість роботи) також заповнюється медіаною. Для «annual_inc» (річний дохід) і «delinq_2yrs» (кількість прострочень за останні два роки) код використовує середнє значення в кожному регіоні, щоб заповнити пропущені значення. Використання середнього або медіани є звичайним для числових даних для підтримки узгодженості розподілів.

Після заповнення певних відсутніх значень, будуть виконані операції замінюючі будь-які порожні значення, що залишилися, нульовими. Цей крок гарантує, що в датафреймі не залишиться пропущених значень.

Після імпутації відсутніх значень обчислимо розподіл класів умов кредиту у відсотках, що дає змогу зрозуміти співвідношення гарних і поганих кредитів (рис. 3.4.2.6):

```
loan_condition_int
0    76.19781
1    23.80219
Name: count, dtype: float64
```

Рисунок 3.4.2.6 - Співвідношення гарних і поганих кредитів.

Як можно побачити відсоток гарних кредитів складає трохи більше 76 відсотків ,поганих кредитів налічується майже 24 відсотки.

Наступним кроком у підготовці даних для моделювання є видалення викидів із набору даних. Викиди – це екстремальні значення, які можуть спотворити результати аналізу та моделювання, що призведе до неточних прогнозів. Видалення цих викидів ґрунтуватиметься на таких умовах:

- «annual_inc» (річний дохід) має бути меншим або дорівнювати 250 000.

- «dti» (співвідношення боргу до доходу) має бути менше або дорівнювати 50.

- «open_acc» (кількість відкритих кредитних ліній) має бути менше або дорівнювати 40.

- «total_acc» (загальна кількість кредитних рахунків) має бути менше або дорівнювати 80.

- «revol_util» (рівень використання поновлюваного кредиту) має бути меншим або дорівнювати 120%.

- «revol_bal» (поновлюваний кредитний баланс) має бути меншим або дорівнювати 250 000.

Ці порогові значення встановлено для видалення крайніх значень, які можуть спотворити набір даних і вплинути на якість моделювання. Застосовуючи ці умови, ми усуваємо викиди, які не відображають типові характеристики позики чи позичальника.

У результаті цього процесу видалення викидів набір даних зменшився майже на 1000 записів (рис. 3.4.2.7).

```
Dataset before removing outlier: (59087, 97)
Dataset after removing outlier: (58106, 97)
```

Рисунок 3.4.2.7 - Датасет до та після видалення викидів.

	0	1	2	3	4
loan_amnt	10000.0	21575.0	28000.0	19200.0	9000.0
funded_amnt	10000.0	21575.0	28000.0	19200.0	9000.0
funded_amnt_inv	10000.0	21575.0	28000.0	19200.0	9000.0
term	36 months	36 months	36 months	36 months	36 months
int_rate	13.18	10.99	9.99	12.29	11.99
...	...	...	...	...	...
disbursement_method	Cash	Cash	Cash	Cash	Cash
debt_settlement_flag	N	N	N	N	N
loan_condition_int	0	1	1	1	0
emp_length_int	1.0	10.0	10.0	6.0	10.0
region	West	West	SouthWest	SouthEast	SouthWest

97 rows × 5 columns

Рисунок 3.4.2.8 - Вигляд датасету після видалення викидів.

Це зменшення вказує на те, що значну кількість викидів було видалено, що призвело до чистішого та надійнішого набору даних для

моделювання. Цей крок має вирішальне значення для забезпечення того, щоб на дані, які використовуються в прогнозному моделюванні, не впливали екстремальні або аномальні значення, що веде до більш надійних і точних моделей.

Розглянемо кореляційний аналіз. Цей крок передбачає вивчення зв’язків між різними змінними в наборі даних, щоб зрозуміти, як вони можуть впливати одна на одну або на конкретну цільову змінну. Кореляційний аналіз дає зрозуміти, які змінні тісно пов’язані, допомагаючи визначити ключові предиктори та потенційні надмірності.

У цьому аналізі ми обчислюємо коефіцієнти кореляції, щоб виміряти силу та напрямок зв’язку між змінними. Позитивна кореляція вказує на те, що зі збільшенням однієї змінної інша має тенденцію до зростання. Негативна кореляція свідчить про протилежне: коли одна змінна збільшується, інша має тенденцію до зменшення. Значення кореляції, близькі до 0, вказують на незначний лінійний зв’язок між змінними або його відсутність.

Розглянемо кореляцію ознак з «loan_condition_int» (рис. 3.4.2.9):

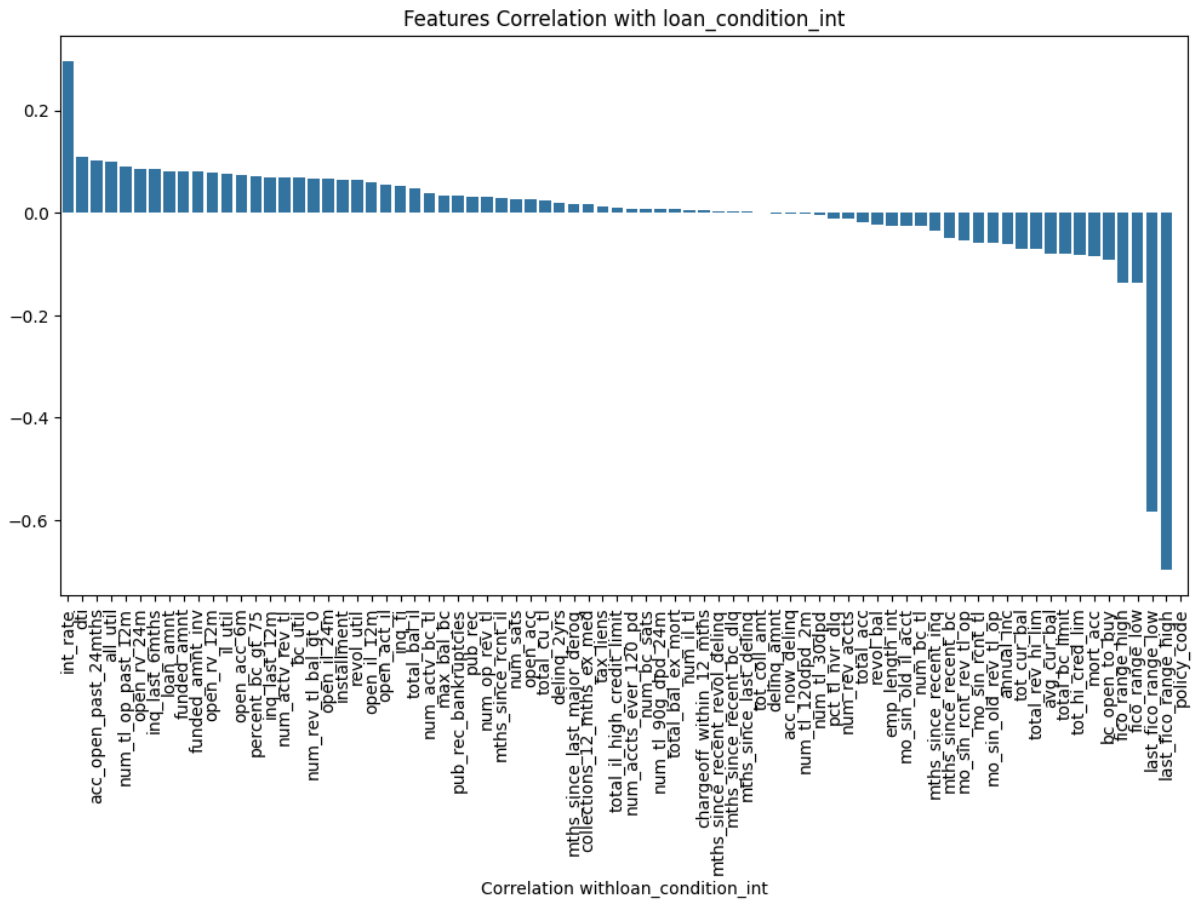

Рисунок 3.4.2.9 - Кореляція ознак з «loan_condition_int».

int_rate	0.296175
dti	0.109689
acc_open_past_24mths	0.103422
all_util	0.099417
num_tl_op_past_12m	0.089561
...	...
fico_range_high	-0.136155
fico_range_low	-0.136157
last_fico_range_low	-0.582552
last_fico_range_high	-0.697276
policy_code	NaN

Рисунок 3.4.2.10 - Таблиця кореляції ознак з «loan_condition_int».

На діаграмі можна побачити, що змінні з найвищою кореляцією з «loan_condition_int» включають «int_rate»(відсоткова ставка), «dti»(співвідношення боргу до доходу) і «acc_open_past_24mths»(рахунки, відкриті за останні 24 місяці). Ці змінні мають тісний зв’язок з умовами позики, що свідчить про те, що вони є впливовими факторами при

визначенні того, чи позику можна буде класифікувати як хорошу чи погану.

З іншого боку, змінними з найменшою кореляцією з «loan_condition_int» є «last_fico_range_low», «last_fico_range_high» і «policy_code»(код політики компанії). Ці змінні демонструють мінімальний або слабкий зв’язок з умовами позики, що свідчить про те, що вони менш значущі для прогнозування того, чи буде позика хорошою чи поганою.

Виберемо змінні з найвищою кореляцією із залежною змінною та дослідимо кореляцію між ними, щоб зрозуміти потенційну мультиколінеарність або надмірність серед найбільш значущих предикторів. Цей крок допомагає переконатися, що вибрані змінні пропонують унікальну інформацію та не вносять нестабільність у процес моделювання.

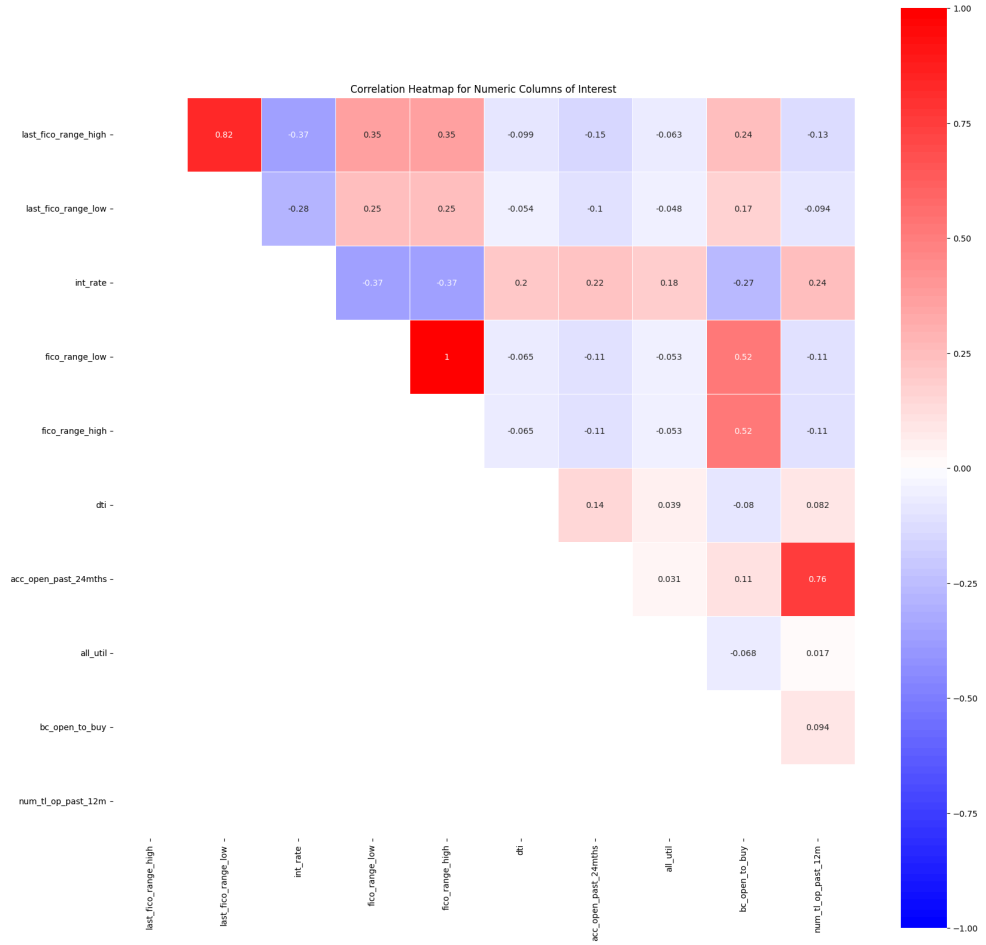

Рисунок 3.4.2.11 - Теплова карта кореляції для числових ознак.

Для початку визначимо змінні, які виявляють сильні кореляції із залежною змінною. Найбільші показники мають «fico_range_low» та «fico_range_high», «last_fico_range_high» та «last_fico_range_low», «acc_open_past_24mths» та «num_tl_op_past_12m». Вони, ймовірно, матимуть найбільший вплив на результат і повинні бути включені в аналіз або прогнозне моделювання. Зосереджуючись на цих змінних, можно підвищити точність моделі та зменшити включення нерелевантних або слабо корельованих функцій.

Далі перевіримо кореляції між цими висококореляційними змінними, щоб виявити будь-які зв’язки, які можуть свідчити про мультиколінеарність — коли дві або більше змінних сильно корельовані одна з одною. «fico_range_low» та «fico_range_high» мають кореляцію 1 ,що свідчить про необхідність видалити одну з цих ознак. Мультиколінеарність може призвести до переобладнання, ускладнюючи інтерпретацію коефіцієнтів моделі та впливаючи на узагальнення моделі на нові дані.

Далі досліджуємо специфічний розподіл зв’язку між змінними під дією залежної змінної «loan_condition_int» (рис. 3.4.2.12). Цей аналіз має на меті зрозуміти, як залежна змінна впливає на розподіл і взаємодію між іншими змінними в наборі даних.

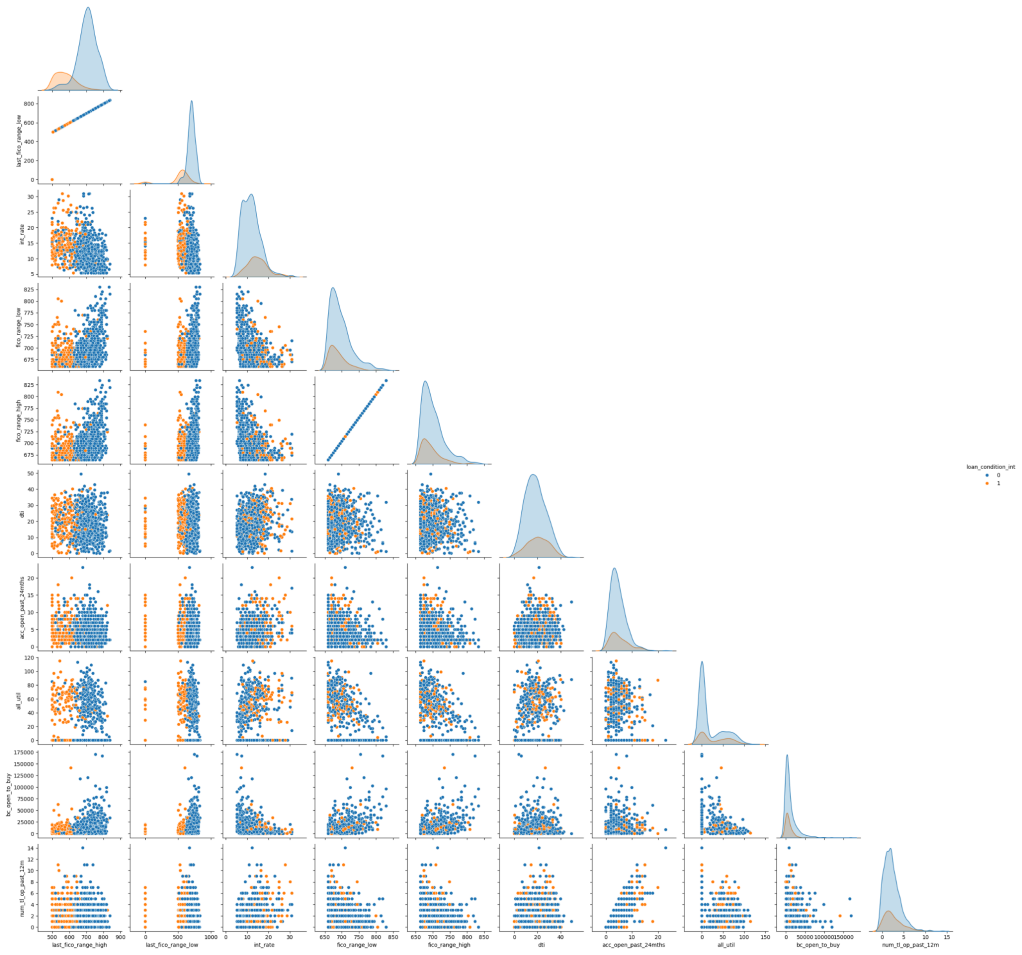

Рисунок 3.4.2.12 - Розподіл зв’язку між змінними під дією залежної змінної «loan_condition_int».

Досліджуючи розподіл зв’язків між ключовими змінними, ви можете отримати більш детальне розуміння структури даних і факторів, що впливають на залежну змінну. Ця інформація є важливою для розробки функцій.

Був проведений кореляційний аналіз різних змінних, щоб оцінити їх значимість і зв’язок з цільовою змінною, «y». Цей аналіз допоміг визначити, які характеристики були найбільш релевантними для прогнозування гарних чи поганих кредитів.

Для проведення кореляційного аналізу категоріальні змінні були закодовані мітками, перетворюючи їх у числові значення для розрахунку коефіцієнтів кореляції. Хоча кодування міток може створювати штучні

зв’язки через довільне призначення номерів категоріям, воно було корисним для отримання швидкого огляду змінної важливості в цьому контексті.

Для побудови фактичної моделі для категоріальних змінних використовувалися більш складні методи кодування, такі як цільове кодування. Ці методи допомагають підтримувати значущі зв’язки між категоріями та цільовою змінною, забезпечуючи точніші результати моделювання.

Переходимо до розробки ознак. Розробка ознак є важливим кроком у підготовці даних для аналізу та моделювання. Вона передбачає перетворення необроблених даних у формат, який підходить для алгоритмів машинного навчання.

Першим кроком розділимо змінні на числові змінні та категоріальні змінні, категоріальні змінні будуть поділені на бінарні змінні та багатовимірні змінні.

```
['grade',
 'sub_grade',
 'home_ownership',
 'verification_status',
 'issue_d',
 'purpose',
 'addr_state',
 'earliest_cr_line',
 'last_pymnt_d',
 'last_credit_pull_d',
 'region']
```

Рисунок 3.4.2.13 - Список багатовимірних змінних.

Виконаємо двійкове кодування для категоріальних змінних і поєднаємо категоріальні ознаки для створення нових ознак взаємодії. Цей процес перетворює категоричні дані в числовий формат, що полегшує обробку й аналіз. Він також створює нові ознаки шляхом об’єднання існуючих категоріальних стовпців, потенційно надаючи додаткову інформацію про взаємодію між категоріальними змінними. Це може бути корисним у прогнозному моделюванні та розробці ознак, оскільки фіксує складніші зв’язки всередині даних.

	0	1	2	3	4
loan_amnt	10000.0	21575.0	28000.0	19200.0	9000.0
funded_amnt	10000.0	21575.0	28000.0	19200.0	9000.0
funded_amnt_inv	10000.0	21575.0	28000.0	19200.0	9000.0
int_rate	13.18	10.99	9.99	12.29	11.99
installment	337.81	706.24	903.35	640.38	298.89
...	...	...	...	...	...
initial_list_status_w	True	True	True	False	True
application_type_Joint App	False	False	False	False	False
hardship_flag_Y	False	False	False	False	False
disbursement_method_DirectPay	False	False	False	False	False
debt_settlement_flag_Y	False	False	False	False	False

Рисунок 3.4.2.14 - Результуюча таблиця двійкового кодування.

Мета наступного кроку полягає в тому, щоб мати однакове співвідношення в навчальних і тестових наборах. Ця дія гарантує, що обидва набори зберігають подібний розподіл цільової змінної, сприяючи послідовній продуктивності моделі та зменшуючи ймовірність спотворення результатів через незбалансовані дані.

Цей процес балансування має бути завершено перед цільовим кодуванням і нормалізацією, щоб уникнути витоку даних. Витік даних відбувається, коли інформація з тестового набору ненавмисно впливає на навчальний набір, що призводить до надто оптимістичної продуктивності моделі під час розробки, але потенційно поганого узагальнення в сценаріях реального світу. Спочатку розділяючи дані, а потім застосовуючи перетворення, можно мінімізувати ризик витоку даних і гарантувати, що тестовий набір залишається справжнім представленням даних.

Розділимо дані для тренування та тестування у співвідношенні 80 до 20.

```
loan_condition_int
0    0.609180
1    0.190806
Name: count, dtype: float64
loan_condition_int
0    0.152308
1    0.047706
Name: count, dtype: float64
(46484, 97)
```

Рисунок 3.4.2.15 - Розподілення даних у датасеті для тренування та тестування.

Як можно побачити розподілення гарних та поганих кредитів приблизно 3 до 1 відповідно. Загальна кількість записів дорівнює майже 46.5 тисячам, кількість ознак дорівнює 97.

Далі виконаємо процес цільового кодування для багатокатегорійних змінних. Цей процес створює закодовані версії навчальних і тестових наборів, де багатокатегорійні змінні замінюються числовими значеннями, отриманими із середнього значення цільової змінної. Закодовані набори даних тепер готові для подальшого аналізу або моделювання зі зниженим ризиком перенавчання та узгодженим кодуванням як для навчальних, так і для тестових наборів.

	0	1	2	3	4
loan_amnt	30000.0	15000.0	40000.0	31500.0	26500.0
funded_amnt	30000.0	15000.0	40000.0	31500.0	26500.0
funded_amnt_inv	30000.0	15000.0	40000.0	31500.0	26500.0
int_rate	7.35	9.99	13.59	12.69	14.65
installment	931.13	483.94	922.25	1056.67	914.1
...	...	...	...	...	...
addr_state	0.159049	0.253237	0.259389	0.260982	0.231731
earliest_cr_line	0.245192	0.256303	0.257962	0.157895	0.2
last_pymnt_d	0.181602	0.321918	0.181875	0.181875	0.321918
last_credit_pull_d	0.212993	0.178764	0.249047	0.212993	0.041237
region	0.223661	0.246777	0.246777	0.248746	0.229776

96 rows × 5 columns

Рисунок 3.4.2.16 - Результуюча таблиця цільового кодування.

Далі треба створити нормалізовані навчальні та тестові набори, де числові змінні стандартизовано, щоб мати середнє значення 0 і стандартне

відхилення 1. Ця нормалізація має вирішальне значення для багатьох алгоритмів машинного навчання, які працюють краще, коли числові дані мають аналогічний масштаб. Послідовне застосування одних і тих самих правил трансформації в навчальних і тестових наборах забезпечує надійне та справедливе оцінювання під час створення моделі.

	0	1	2	3	4
loan_amnt	1.76814	0.058236	2.908076	1.93913	1.369162
funded_amnt	1.76814	0.058236	2.908076	1.93913	1.369162
funded_amnt_inv	1.769272	0.058843	2.909557	1.940314	1.370172
int_rate	-1.144923	-0.597682	0.148557	-0.038003	0.368283
installment	1.889787	0.181158	1.855859	2.369452	1.824719
...	...	...	...	...	...
addr_state	0.159049	0.253237	0.259389	0.260982	0.231731
earliest_cr_line	0.245192	0.256303	0.257962	0.157895	0.2
last_pymnt_d	0.181602	0.321918	0.181875	0.181875	0.321918
last_credit_pull_d	0.212993	0.178764	0.249047	0.212993	0.041237
region	0.223661	0.246777	0.246777	0.248746	0.229776

96 rows × 5 columns

Рисунок 3.4.2.17 - Результуюча таблиця після нормалізації.

Набір даних незбалансований, приблизно 80% даних представляють гарні кредити. Враховуючи цей дисбаланс, ми особливо повинні приділити увагу точним прогнозуванням проблемних кредитів, які становлять меншу частину набору даних. Цей дисбаланс може призвести до моделей, упереджених у бік більшості класів, у цьому випадку гарних позик, що призведе до високої загальної точності, але поганої продуктивності у виявленні проблемних позик.

Зменшимо вибірку навчального набору, щоб створити більш збалансований розподіл між гарними та поганими кредитами.

	count
loan_condition_int	
0	11087
1	11087

Рисунок 3.4.2.18 - Розподіл після балансування.

Після процесу балансування даних в нашому датасеті ,в кожній категорії порівну записів - 11087. Таким чином, це допомагає запобігти упередженню моделі до класу більшості та покращує її здатність вчитися у класу меншості.

Останній крок аналізу даних - вибір ознак. Вибір ознак – це процес визначення та збереження найбільш релевантних ознак із набору даних, одночасно відкидаючи ті, які є надлишковими, нерелевантними або з низьким вмістом інформації. Це важливий крок у попередній обробці даних, який часто призводить до покращення продуктивності моделі, зменшення перенавчання та швидшого навчання.

Спершу визначимо та видалимо ознаки з низькою дисперсією з навчального набору, забезпечуючи використання лише функцій із достатньою варіабельністю в подальшому аналізі чи моделюванні.

Після видалення в датасеті залишилось трохи більше 22 тисяч записів з ознаками в кількості 50.

Усуваючи ознаки з невеликим вмістом інформації, код оптимізує набір даних і зменшує ризик переобладнання, створюючи надійніші моделі.

Далі виконаємо обгортку поетапного виділення з використанням послідовного вибору ознак (SFS) для визначення найкращої підмножини ознак для певного класифікатора, у цьому випадку випадкового лісу.

Послідовний вибір починається з порожнього набору ознак і послідовно додаються ознаки, які покращують продуктивність моделі (рис. 3.4.2.19). Цей процес триває, доки не буде досягнуто бажаної кількості ознак(в даному випадку 50) або поки не буде спостерігатися подальше покращення.

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
1	(11,)	[0.883106340759448, 0.8854514296022369]	0.884279	(last_fico_range_high,)	0.005045	0.001173	0.001173
2	(11, 12)	[0.883106340759448, 0.8854514296022369]	0.884279	(last_fico_range_high, last_fico_range_low)	0.005045	0.001173	0.001173
3	(11, 12, 30)	[0.8828357535852801, 0.8852710381527915]	0.884053	(last_fico_range_high, last_fico_range_low, ch...	0.005239	0.001218	0.001218
4	(11, 12, 30, 44)	[0.8826553621358347, 0.8851808424280689]	0.883918	(last_fico_range_high, last_fico_range_low, ch...	0.005433	0.001263	0.001263
5	(11, 12, 15, 30, 44)	[0.8823847749616668, 0.8848200595291783]	0.883602	(last_fico_range_high, last_fico_range_low, ac...	0.005239	0.001218	0.001218
...	...	...	...	...	...	...	...
46	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	[0.8740867682871832, 0.8754397041580229]	0.874763	(funded_amnt, funded_amnt_inv, int_rate, insta...	0.002911	0.000676	0.000676
47	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	[0.8720122666185622, 0.8757102913321908]	0.873861	(funded_amnt, funded_amnt_inv, int_rate, insta...	0.007956	0.001849	0.001849
48	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	[0.8753495084333003, 0.8726436366916208]	0.873997	(funded_amnt, funded_amnt_inv, int_rate, insta...	0.005821	0.001353	0.001353
49	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[0.8765220528546946, 0.8750789212591323]	0.8758	(loan_amnt, funded_amnt, funded_amnt_inv, int_...	0.003105	0.000722	0.000722
50	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[0.8729142238657888, 0.8703887435735547]	0.871651	(loan_amnt, funded_amnt, funded_amnt_inv, int_...	0.005433	0.001263	0.001263

50 rows × 7 columns

Рисунок 3.4.2.19 - Результуюча таблиця роботи SFS.

Послідовний вибір ознак в результаті дасть нам такий набір ознак (рис. 3.4.2.20):

```
['pub_rec',
 'last_fico_range_high',
 'last_fico_range_low',
 'collections_12_mths_ex_med',
 'acc_now_delinq',
 'open_acc_6m',
 'open_il_12m',
 'open_il_24m',
 'total_bal_il',
 'total_cu_tl',
 'inq_last_12m',
 'chargeoff_within_12_mths',
 'num_tl_30dpd',
 'pub_rec_bankruptcies',
 'tax_liens']
```

Рисунок 3.4.2.20 - Ознаки вибрані SFS.

Створимо графік, що демонструє продуктивність моделі, після додавання більшої кількості ознак. Діаграма забезпечує візуальне представлення поетапного процесу вибору ознак, включаючи стандартне відхилення продуктивності моделі (рис. 3.4.2.21):

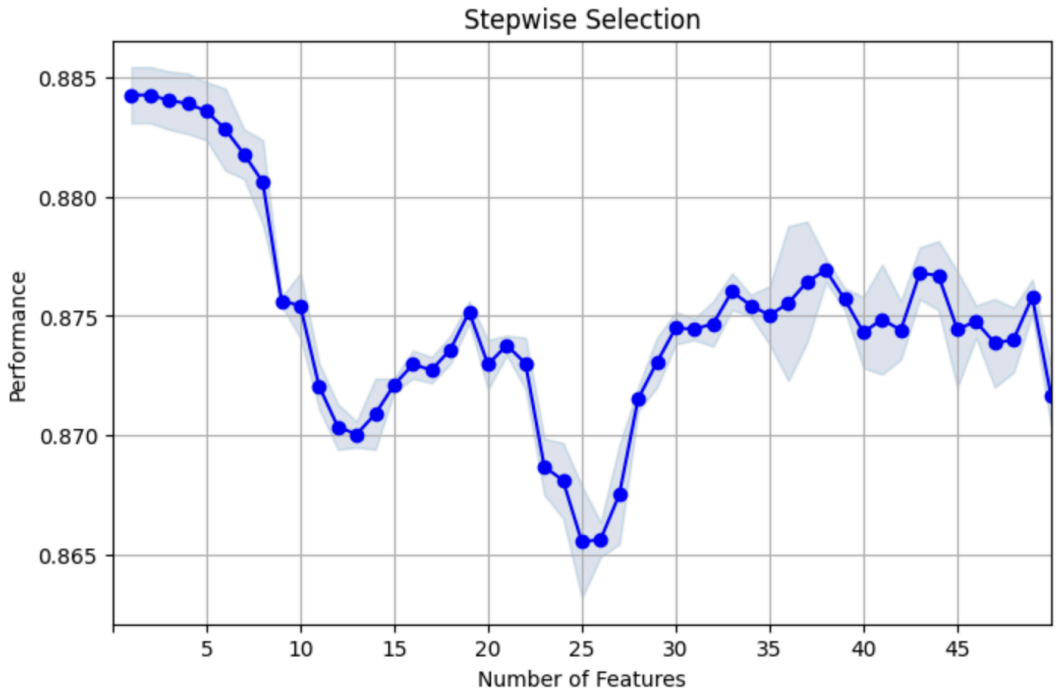

Рисунок 3.4.2.21 - Діаграма SFS.

Модифікуємо навчальний та тестовий датасети ,які будуть містити тільки вибрані ознаки.

```
final train/test X shape (target encoded): (22174, 15) (11622, 15)
final train/test y shape: (22174, 1) (11622, 1)
```

Рисунок 3.4.2.22 - Модифікований датасет.

Фіанальний вигляд датасету буде таким (рис. 3.4.2.23):

	pub_rec	last_fico_range_high	last_fico_range_low	collections_12_mths_ex_med	acc_now_delinq	open_acc_6m	open_il_12
0	-0.386145	0.803025	0.588709	-0.132866	-0.0698	-0.447815	-0.42369
1	1.246672	0.742010	0.551890	-0.132866	-0.0698	-0.447815	-0.42369
2	-0.386145	0.375920	0.330978	-0.132866	-0.0698	-0.447815	-0.42369
3	-0.386145	1.230130	0.846440	-0.132866	-0.0698	-0.447815	-0.42369
4	-0.386145	-0.112201	0.036429	-0.132866	-0.0698	-0.447815	-0.42369

Рисунок 3.4.2.23 - Фіанальний вигляд датасету.

У фінальній версії підготовлених даних залишилось 15 ознак. Це означає, що після всіх етапів попередньої обробки та відбору функцій найбільш значущі ознаки були відібрані для подальшого аналізу та моделювання. Ці 15 ознак вважаються найбільш інформативними та суттєвими для передбачення цільової змінної та вирішення поставленої задачі.

3.5 Створення моделей

Моделі будуть навчені на навчальному наборі, і буде використана 5-кратна перехресна перевірка для оцінки точності моделей під час навчання. Перехресна перевірка допомагає переконатися, що модель добре узагальнюється, багаторазово розбиваючи дані на набори для навчання та перевірки, забезпечуючи більш надійну оцінку продуктивності моделі. На цьому етапі для створення прогнозних моделей використовувалися різні алгоритми машинного навчання.

Кожна з цих моделей пропонує унікальний підхід до вирішення проблеми прогнозування результатів позики: логістична регресія, дерево рішень, к найближчих сусідів, випадковий ліс, Гауссівський наївний байєс, LightGBM, XGBoost, посилення градієнта, нейронна мережа.

Досліджуючи різноманітний набір моделей, мета полягає в тому, щоб знайти ту, яка пропонує найкращу ефективність і узагальнення для прогнозування умов позики. Вибір моделі може залежати, зокрема, від таких факторів, як точність, час навчання та легкість інтерпретації.

3.6 Аналіз результатів моделювання

Проведемо аналіз тестування базових моделей на наборі даних із недостатньою вибіркою передбачає оцінку різних моделей машинного навчання на даних, де більшість класів було зменшено для створення більш збалансованого набору даних.

Логістична регресія - лінійна модель, яка використовується для бінарної класифікації. Модель розраховує ймовірність того, що вхідні дані належать до того чи іншого класу, що робить його придатним для прогнозування хороших чи поганих кредитів.

Ця матриця невідповідностей (рис. 3.6.1) для логістичної регресії представляє продуктивність моделі логістичної регресії для задачі двійкової класифікації.

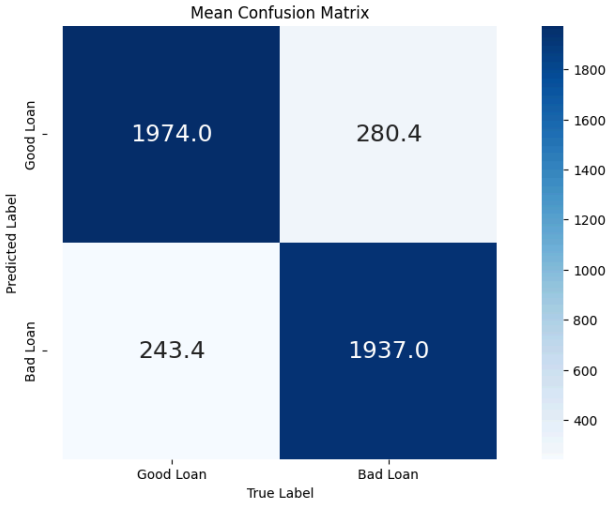

Рисунок 3.6.1 - Матриця невідповідностей для логістичної регресії.

Вона показує кількість прогнозованих і фактичних класів, що дозволяє нам зрозуміти, наскільки добре модель розрізняє їх.

Матриця невідповідностей має чотири значення, організовані таким чином:

Верхній ліворуч (справжні позитивні результати): 1974 – це кількість зразків, правильно передбачених як позитивний клас.

Верхній правий (хибнопозитивні результати): 243,4 – це зразки, неправильно визначені як позитивні, хоча насправді вони були негативними.

Внизу зліва (помилкові негативні результати): 280,4 – це значення показує зразки, неправильно визначені як негативні, хоча насправді вони були позитивними.

Внизу праворуч (Справжні негативи): 1937 – це зразки, правильно передбачені як негативний клас.

Більшість прогнозів правильні (88 %), про що свідчить велика кількість істинно позитивних і істинно негативних. Наявність

хибно-позитивних і хибно-негативних результатів (12 %) вказує на те, що модель допускає помилки, при цьому хибно-негативні результати можуть вказувати на недооцінку позитивного класу, а хибно-позитивні результати вказують на переоцінку того самого класу. Ця матриця невідповідностей пропонує відносно збалансовану модель із розумним співвідношенням між істинними та хибними прогнозами.

Дерево рішень - деревоподібна модель, яка розділяє дані на основі конкретних умов, створюючи ієрархію рішень. Ця модель є інтуїтивно зрозумілою та легкою для розуміння.

Для цієї моделі результати матриці подібності, майже співпадають з матрицею логістичної регресії. 87 % правильних і 13 % неправильних.

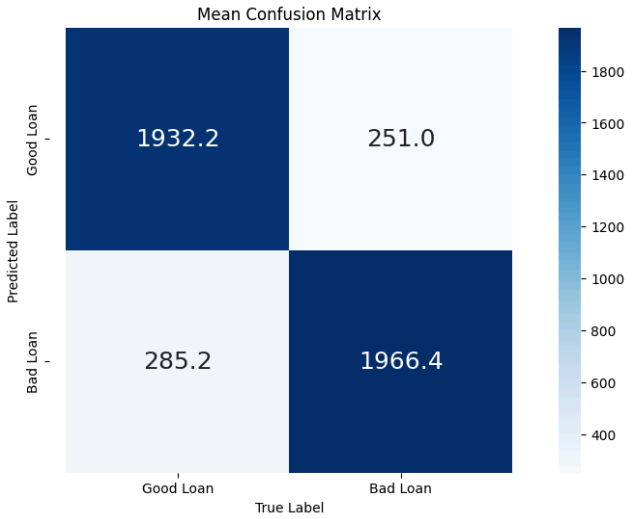

Рисунок 3.6.2 - Матриця невідповідностей для дерева рішень.

Наступна модель - K Найближчих сусідів. Це непараметрична модель, яка передбачає клас вибірки на основі класів її найближчих сусідів. Це просто й ефективно для невеликих наборів даних, але може бути дорогим у плані обчислень для великих наборів даних.

Згідно з матрицею невідповідностей для даної моделі (рис. 3.6.3):

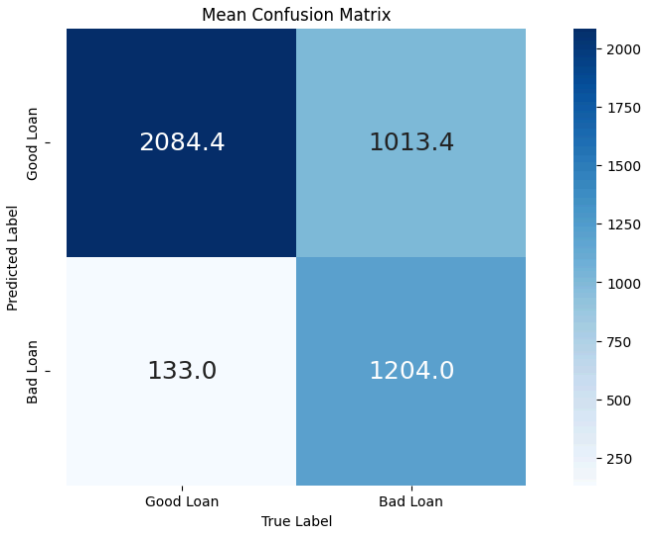

Рисунок 3.6.3 - Матриця невідповідностей для K Найближчих сусідів.

Ця модель показує набагато гірший результат порівнюючи з попередніми моделями. Справжні позитиви та справжні негативи становлять 2084.4 і 1204 відповідно (74 %). Частка хибних прогнозів збільшилась аж на 14 %, що показує значно меншу ефективність даної моделі. Це найгірший результат серед усіх моделей.

Випадковий ліс - модель ансамблю, яка створює кілька дерев рішень і поєднує їхні прогнози. Цей підхід часто забезпечує більшу точність і надійність порівняно з окремими деревами, як ми можемо побачити з матриці невідповідностей (рис. 3.6.4).

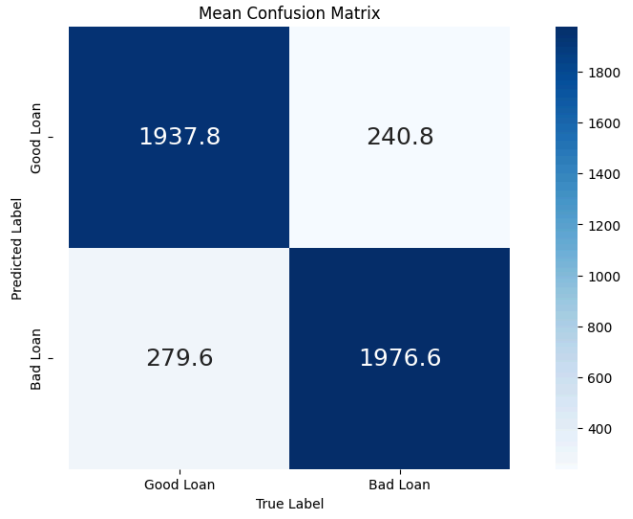

Рисунок 3.6.4 - Матриця невідповідностей для випадкового лісу.

Показники правильних прогнозів незначною мірою зросли порівнюючи з результатами для моделі дерев рішень.

Гауссівський наївний байєс - імовірнісна модель, заснована на теоремі Байєса, припускаючи, що ознаки умовно незалежні від класу. Ця модель проста і добре працює з даними великого розміру, але показала один з найгірших результатів порівняно з іншими моделями ,всього 77 % правильних прогнозів, що лише на 3 % більше за показники моделі K Найближчих сусідів.

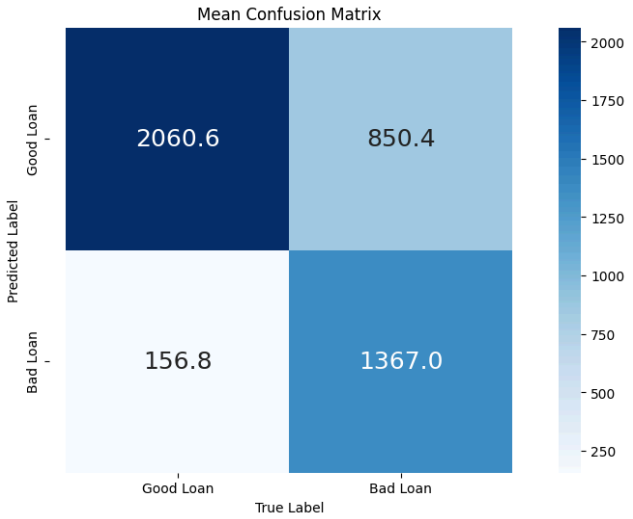

Рисунок 3.6.5 - Матриця невідповідностей для Гауссівського наївного байєсу.

Інші моделі, такі як LightGBM, XGBoost, Gradient Boosting і Neural Network, продемонстрували подібні результати, досягнувши приблизно 88% правильних прогнозів і 12% неправильних прогнозів. Ці результати свідчать про незмінний рівень точності в багатьох розширених алгоритмах машинного навчання, що вказує на те, що моделі загалом працюють добре.

LightGBM - платформа підвищення градієнта, розроблена для високої продуктивності та масштабованості. Для підвищення точності та ефективності він використовує стратегію росту дерева по листях.

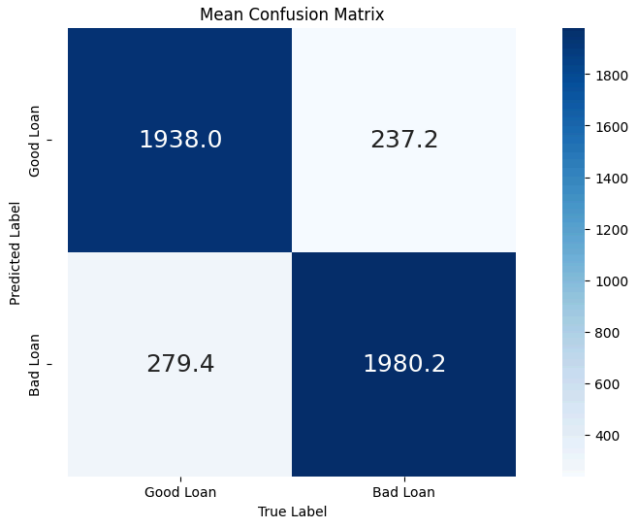

Рисунок 3.6.6 - Матриця невідповідностей для LightGBM.

XGBoost - ще один популярний алгоритм посилення градієнта, відомий своєю швидкістю та продуктивністю. Він включає регулярізацію для запобігання переобладнанню та підтримує різні функції оптимізації.

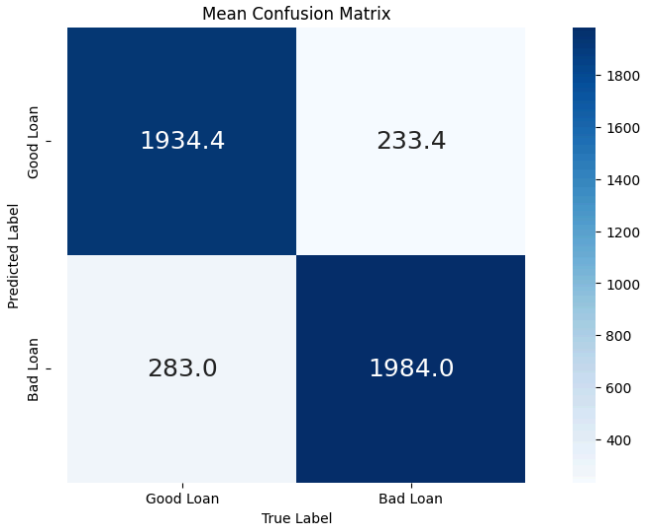

Рисунок 3.6.7 - Матриця невідповідностей для XGBoost.

Посилення градієнта - метод ансамблю, який будує дерева послідовно, при цьому кожне нове дерево має на меті виправити помилки попередніх.

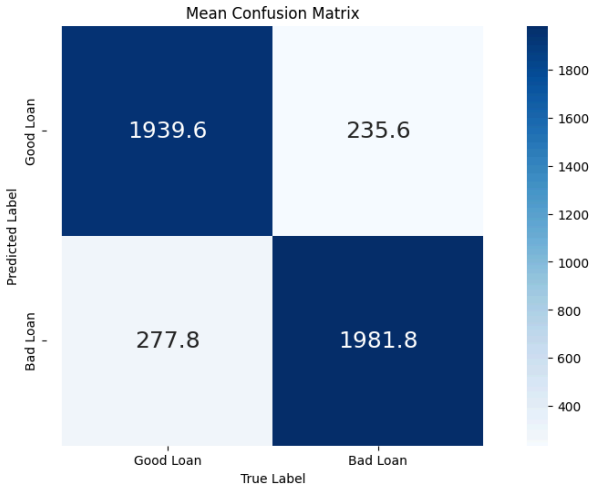

Рисунок 3.6.8 - Матриця невідповідностей для посилення градієнта.

Нейронна мережа - гнучка модель, що імітує структуру людського мозку. Нейронні мережі можуть фіксувати складні зв'язки та шаблони в даних, що робить їх ідеальними для завдань глибокого навчання.

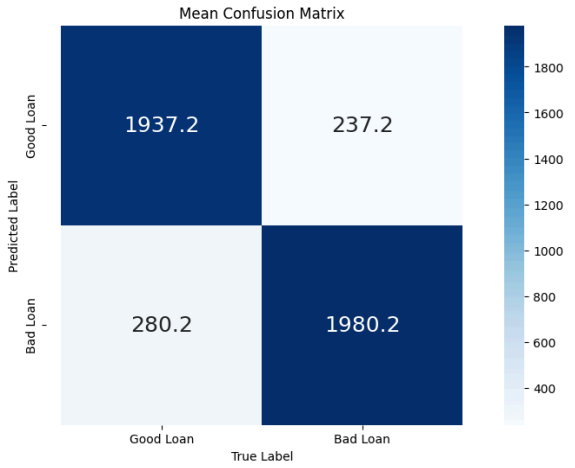

Рисунок 3.6.9 - Матриця невідповідностей для нейронна мережа.

Загальні результати для навчених моделей буде представлено в таблиці (рис. 3.6.10), яка підсумовує різні показники, що використовуються для оцінки ефективності моделей.

Таблиця містить такі ключові показники:

«accuracy» - цей показник вимірює частку правильних прогнозів, зроблених моделлю. Він забезпечує загальну індикацію продуктивності моделі, але не враховує дисбаланс класів.

«auc» (Площа під кривою ROC) - цей показник представляє площу під кривою робочих характеристик приймача, яка відображає істинну позитивну швидкість проти помилкової позитивної частоти. Вища «auc» вказує на кращу дискримінацію моделі між класами.

«ks» (статистика Колмогорова-Смирнова) - ця метрика кількісно визначає відстань між функціями емпіричного розподілу двох вибірок. Він зазвичай використовується для вимірювання поділу між прогнозованими ймовірностями для різних класів, вказуючи, наскільки добре модель розрізняє їх.

«precision» - цей показник обчислює частку справжніх позитивних результатів серед усіх прогнозованих позитивних результатів. Це корисно для оцінки ефективності моделі в контекстах, де помилкові спрацьовування мають високу вартість.

«recall» - цей показник вимірює частку справжніх позитивних результатів серед усіх фактичних позитивних результатів. Це корисно в ситуаціях, коли відсутність справжніх позитивних результатів має значні наслідки.

«f1» - цей показник поєднує в собі точність і запам’ятовування в одне значення, що представляє їх гармонійне середнє. Це особливо корисно, коли існує нерівномірний розподіл класів, що забезпечує збалансоване уявлення про продуктивність моделі.

	accuracy	auc	ks	precision	recall	f1
Logistic Regression (undersample)	0.881889	0.938157	0.769145	0.888341	0.873552	0.880881
Decision Tree (undersample)	0.879093	0.935257	0.764359	0.873399	0.886838	0.880021
K Nearest Neighbors (undersample)	0.741499	0.800495	0.500764	0.900465	0.542931	0.677390
Random Forest (undersample)	0.882655	0.937735	0.768146	0.876020	0.891411	0.883604
Gaussian Naive Bayes (undersample)	0.772887	0.905621	0.697065	0.896857	0.616358	0.729153
Light GBM (undersample)	0.883512	0.939992	0.770456	0.876321	0.893005	0.884577
XGBoost (undersample)	0.883557	0.939743	0.770303	0.875147	0.894787	0.884841
Gradient Boosting (undersample)	0.884234	0.940041	0.770349	0.877059	0.893747	0.885285
Neural Network (undersample)	0.883332	0.939134	0.768646	0.876140	0.893036	0.884483

Рисунок 3.6.10 - Загальні результати для навчених моделей.

З таблиці модель з найвищою точністю – це модель Підсилення градієнту, яка досягла точності 0,884234. Цей результат свідчить про те, що модель правильно передбачила приблизно 88,4% результатів у наборі даних, що робить її найточнішою серед оцінюваних моделей.

Оцінка точності 0,884234 свідчить про те, що модель Підсилення градієнту є надійною для прогнозування правильних результатів, перевершуючи інші моделі з точки зору точності.
Враховуючи високу точність, модель Підсилення градієнту може бути сильним кандидатом для використання у виробництві або в реальних сценаріях, де точні прогнози є критичними.

Висока точність моделі Підсилення градієнту означає, що вона ефективно фіксує ключові показники в даних, що дає змогу створювати надійні прогнози. Ця точність може бути корисною в оцінці кредитного ризику.

Розглянемо вплив гіперпараметрів на ефективність моделей (рис. 3.6.11).

```
'LogisticRegression': {'C': [0.1, 1, 10], 'penalty': ['l2'], 'solver': ['lbfgs']},
'DecisionTreeClassifier': {'max_depth': [3, 5, 10], 'criterion': ['gini', 'entropy']},
'KNeighborsClassifier': {'n_neighbors': [3, 5, 7], 'weights': ['uniform', 'distance']},
'RandomForestClassifier': {'n_estimators': [100, 200], 'max_depth': [5, 10], 'criterion': ['gini', 'entropy']},
'GaussianNB': {},  # No significant hyperparameters
'LGBMClassifier': {'num_leaves': [31, 50], 'learning_rate': [0.01, 0.1], 'n_estimators': [100, 200]},
'XGBClassifier': {'max_depth': [3, 5, 7], 'learning_rate': [0.01, 0.1], 'n_estimators': [100, 200]},
'GradientBoostingClassifier': {'learning_rate': [0.01, 0.1], 'n_estimators': [100, 200], 'max_depth': [3, 5]},
'MLPClassifier': {'hidden_layer_sizes': [(100,), (50, 50)], 'learning_rate': ['constant', 'adaptive']},
```

Рисунок 3.6.11 - Класифікатори з гіперпараметрами.

Гіперпараметри представляють параметри налаштування для різних класифікаторів машинного навчання. Ці гіперпараметри контролюють поведінку та конфігурацію моделей, дозволяючи оптимізувати їх для кращої продуктивності.

Розглянемо гіперпараметри для кожного класифікатора:

Логістична регресія:

«C» - обернена сила регуляризації. Нижчі значення вказують на сильнішу регулярізацію. Загальний параметр налаштування, щоб збалансувати компроміс між надмірним і недостатнім оснащенням.

«penalty» - тип регуляризації, який використовується. «l2» - це значення за замовчуванням із застосуванням квадратичного штрафу до коефіцієнтів.

«solver» - алгоритм, що використовується для оптимізації. «lbfgs» є поширеним вибором, який підходить для менших наборів даних або багатокласових проблем.

Класифікатор дерева рішень:

«max_depth» - максимальна глибина дерева. Глибші дерева можуть вловлювати складніші візерунки, але вони більш схильні до переобладнання.

«criterion» - функція, яка використовується для вимірювання якості розбиття. «gini» та «entropy» є звичайними варіантами, де «gini» є типовим.

Класифікатор K Найближчих сусідів:

«n_neighbors» - кількість найближчих сусідів для класифікації. Менші значення можуть бути більш чутливими до локальних моделей, тоді як більші значення краще узагальнюють.

«weights» - як ваги призначаються сусідам. «uniform» надає однакову вагу всім, тоді як «distance» призначає ваги обернено пропорційні відстані.

Класифікатор випадкового лісу:

«n_estimators» - кількість дерев у лісі. Більше дерев, як правило, призводить до кращої продуктивності, але вимагає більше обчислень.

«max_depth» - максимальна глибина для кожного дерева. Контроль глибини допомагає збалансувати складність моделі.

«criterion» - те саме, що і для дерев рішень, що визначає якість розбиття.

Гауссівський наївний байєс - цей класифікатор немає значущих гіперпараметрів які можно налащтувати.

Класифікатор LightGBM:

«num_leaves» - максимальна кількість листів у кожному дереві. Вищі значення можуть призвести до більш складних дерев.

«learning_rate» - контролює розмір кроку під час навчання. Нижчі показники можуть призвести до більш стабільних моделей, тоді як вищі показники можуть тренуватися швидше, але є ризик переобладнання.

«n_estimators» - кількість раундів підвищення. Більше оцінювачів зазвичай покращують продуктивність, але вимагають більше обчислень.

XGBClassifier:

«max_depth» - максимальна глибина для кожного дерева в ансамблі.

«learning_rate» - те саме, що LightGBM, впливає на процес навчання.

«n_estimators» - кількість раундів посилення, що впливає на складність моделі.

Класифікатор посилення градієнта:

«learning_rate» - Подібно до LightGBM і XGBClassifier, впливає на розмір кроку під час посилення.

«n_estimators» - загальна кількість раундів посилення, що впливає на складність моделі.

«max_depth» - максимальна глибина для кожного дерева, подібно до дерев рішень і випадкового лісу.

MLPClassifier (нейронна мережа):

«hidden_layer_sizes» - визначає архітектуру прихованих шарів. Різні конфігурації можуть впливати на здатність моделі вивчати складні шаблони.

«learning_rate» - контролює швидкість навчання. «constant» підтримує фіксовану швидкість, тоді як «adaptive» регулює залежно від прогресу навчання.

Ці гіперпараметри керують поведінкою кожної моделі, дозволяючи точно налаштувати їх для конкретних наборів даних або цілей. Вивчаючи різні комбінації цих гіперпараметрів, можно оптимізувати продуктивність моделі, забезпечуючи хороший баланс між точністю, складністю та узагальненням.

Після підбору найкращих параметрів з якими моделі показали найкращу ефективність ,отримаємо діаграму з показниками (рис. 3.6.12):

Рисунок 3.6.12 - Результати для моделей з підібраними гіперпараметрами.

Невелике збільшення спостерігається у всіх моделях. Незважаючи на це покращення, К Найближчих сусідів і Гауссівський наївний байєс все ще демонструють найгірші результати серед оцінюваних моделей. Їх нижча продуктивність може бути пов’язана з їх чутливістю до шумових даних, недостатньою точністю налаштування або обмеженнями моделі.

На відміну від цих моделей, логістична регресія показує найкращий результат, її точність покращилася з 0,8818 до 0,8869. Це збільшення свідчить про те, що логістична регресія отримує користь від коригувань або налаштування гіперпараметрів, застосованих під час навчання та

тестування моделі. Відносна простота алгоритму та стійкість до перенавчання можуть сприяти його високій продуктивності, особливо в порівнянні зі складнішими моделями, такими як нейронні мережі або класифікатори на основі дерева.

Результати показують, що навіть невеликі зміни можуть призвести до вимірних покращень точності моделі, підкреслюючи важливість продовження експериментів та оптимізації.

Спробуємо використати Беггінг, методику ансамблевого навчання, на наборі моделей машинного навчання. Беггінг, скорочення від Бутстрепова агрегація, - це методика ансамблевого навчання, яка використовується для підвищення точності та надійності моделей машинного навчання, зокрема в задачах класифікації та регресії. Він призначений для зменшення дисперсії та перенавчання шляхом навчання декількох моделей на різних підмножинах даних навчання та агрегування їхніх прогнозів.

Загалом пакетування покращує продуктивність моделі шляхом зменшення дисперсії за допомогою усереднення по ансамблю, але є випадки, коли воно менш ефективно або навіть контрпродуктивно, особливо з певними класифікаторами, такими як К Найближчих сусідів і Гауссівський наївний байєс. Тому ми не будемо використовувати ці класифікатори в даному методі.

Створисо ансамблі Бутстрепової агрегації за допомогою різних базових класифікаторів, навчимо ці моделі на наборі наших даних і оцінимо їх продуктивність за допомогою кількох показників.

	accuracy	auc	ks	precision	recall	f1
Logistic Regression (bagging)	0.887369	0.938098	0.768136	0.716100	0.874459	0.787396
Decision Tree (bagging)	0.879539	0.939044	0.768062	0.692265	0.891053	0.779180
Random Forest (bagging)	0.877904	0.940306	0.770666	0.687241	0.895743	0.777760
Light GBM (bagging)	0.880916	0.940275	0.770553	0.695713	0.889971	0.780943
XGBoost (bagging)	0.880571	0.940375	0.770349	0.694930	0.889971	0.780449
Gradient Boosting (bagging)	0.881346	0.940282	0.771183	0.696252	0.891414	0.781838
Neural Network (bagging)	0.879797	0.939276	0.768723	0.693607	0.888528	0.779061

Рисунок 3.6.13 - Результати Бутстрепової агрегації.

З результуючої таблиці розуміємо, що маємо покращення результатів. Логістична регресія все ще зберігає лідерство ,і навіть покращення результату з 0,8869 до 0,8873, шо може свідчити про ефективність даного підходу.

Нарешті проведемо оцінку моделей на тестовому наборі даних. Моделі оцінювали на тестовому наборі, щоб отримати справжню точність тесту. Цей процес забезпечує більш реалістичну оцінку продуктивності моделі, оскільки тестовий набір представляє невидимі дані, яким моделі не піддавалися під час навчання або перехресної перевірки.

	accuracy	auc	ks	precision	recall	f1
Logistic Regression	0.886939	0.938122	0.767637	0.715044	0.874459	0.786758
Decision Tree	0.881260	0.936481	0.767511	0.696944	0.888528	0.781161
Random Forest	0.877990	0.938875	0.768351	0.687847	0.894300	0.777604
Light GBM	0.877990	0.940472	0.770088	0.687431	0.895743	0.777882
XGBoost	0.880141	0.940519	0.768189	0.693843	0.890332	0.779902
Gradient Boosting	0.881260	0.940291	0.770697	0.696167	0.891053	0.781646
Neural Network	0.877990	0.938431	0.767524	0.689106	0.889971	0.776763
Logistic Regression_bagging	0.887369	0.938098	0.768136	0.716100	0.874459	0.787396
Decision Tree_bagging	0.879539	0.939044	0.768062	0.692265	0.891053	0.779180
Random Forest_bagging	0.877904	0.940306	0.770666	0.687241	0.895743	0.777760
Light GBM_bagging	0.880916	0.940275	0.770553	0.695713	0.889971	0.780943
XGBoost_bagging	0.880571	0.940375	0.770349	0.694930	0.889971	0.780449
Gradient Boosting_bagging	0.881346	0.940282	0.771183	0.696252	0.891414	0.781838
Neural Network_bagging	0.879797	0.939276	0.768723	0.693607	0.888528	0.779061

Рисунок 3.6.14 - Результати тестування.

Оцінки ефективності на наборі для тестування майже такі ж, як результати перехресної перевірки на навчальному наборі. Ця узгодженість між навчальними й тестовими результатами є позитивним свідченням того, що моделі не переобладнуються, що свідчить про те, що вони добре узагальнюють невидимі дані.

Логістична регресія після Беггінгу має найвищу точність серед моделей, що вказує на те, що вона правильно передбачає більшу частку результатів порівняно з іншими моделями. Ця висока точність свідчить про те, що Беггінг ефективний у зменшенні помилок і покращенні ефективності прогнозування.

Поєднання високої точності(«accuracy»), акуратність(«precision») та оцінки «f1» вказує на те, що логістична регресія після Беггінгу ефективна в більшості сценаріїв.

Низька «auc» і «recall» вказують на області для покращення, такі як коригування ваг класу або налаштування гіперпараметрів для посилення розрізнення та фіксації більшої кількості позитивних випадків.

Враховуючи високу точності(«accuracy») і акуратність(«precision»), логістична регресія може бути придатною для застосувань, де помилкові спрацьовування є дорогими. Однак низький показник «recall» може викликати занепокоєння в контекстах, де ризиковано втратити позитивні випадки.

Розглянемо важливість ознак у моделі логістичної регресії, щоб зрозуміти, які змінні найбільше впливають на прогнози моделі. Важливість ознаки в логістичній регресії визначається коефіцієнтами моделі, які вказують на вплив кожної ознаки на цільову змінну.

Logistic Regression
y=1 top features

Weight[?]	Feature
+0.338	last_fico_range_low
+0.149	open_il_24m
+0.119	total_bal_il
+0.089	open_acc_6m
+0.055	pub_rec_bankruptcies
+0.046	total_cu_tl
+0.031	inq_last_12m
+0.012	acc_now_delinq
+0.007	tax_liens
-0.000	chargeoff_within_12_mths
-0.016	collections_12_mths_ex_med
-0.032	num_tl_30dpd
-0.057	pub_rec
-0.081	open_il_12m
-1.204	<BIAS>
-2.924	last_fico_range_high

Рисунок 3.6.15 - важливість ознак у моделі логістичної регресії.

Найважливішими ознаками в моделі логістичної регресії є «last_fico_range_low» (ознака представляє нижню межу останнього діапазону кредитних балів FICO), яка має найбільше значення коефіцієнта, «open_il_24m» (ознака позначає кількість кредитів, відкритих за останні 24 місяці) і «total_bal_il» (ознака представляє загальний баланс за всіма кредитами з виплатою). Ці ознаки відіграють важливу роль у визначенні прогнозів моделі, вказуючи на те, що вони мають значний вплив на цільову змінну.

Загалом, подібність показників продуктивності між наборами тестів для тренувань і тестів які не використовувались в тренуванні є сильним свідченням успішного навчання та перевірки моделі, що дає впевненість у надійності та точності моделей.

3.7 Висновки до розділу 3

У цій главі були розглянуті застосування алгоритмів логістична регресія, дерево рішень, к найближчих сусідів, випадковий ліс, Гауссівський наївний байєс, LightGBM, XGBoost, посилення градієнта, нейронна мережа для прогнозування дефолту за кредитами. Були підкреслена важливість підготовки даних як основного кроку в процесі

створення моделі. Це передбачало використання різних методів, таких як, кореляційний аналіз, візуалізація та перевірки якості даних щодо ознак, щоб полегшити ефективне видалення ознак.

Дослідження проілюструвало різноманітність методів, доступних для обробки та аналізу категоріальних даних, наголошуючи на перетворенні таких даних у формат, сумісний із моделлю.

Була підкреслена ключова роль обробки даних і встановлення розуміння змінної важливості в процесі розробки моделі. Зосередившись на цих критичних кроках, були успішно створені моделі, які дали надійні результати для прогнозування успішності кредитування.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Дослідження та аналіз, проведені в цій дипломній роботі в розділах 1, 2 і 3, дають цінну інформацію про інтеграцію методів машинного навчання в управління та запобігання фінансовим ризикам. У цьому заключному розділі містяться основні висновки та окреслюються перспективи майбутніх досліджень і практичних застосувань.

У розділі 1 ми ознайомилися з основними концепціями управління фінансовими ризиками та поточним станом використання машинного навчання в цій галузі. Наше дослідження підкреслило важливість розуміння фінансового ризику, використання статистичних моделей і визнання потенціалу програм машинного навчання. Розділ завершувався формулюванням дослідницької проблеми, готуючи основу для подальшого поглибленого аналізу.

У розділі 2 представлений комплексний аналіз різних методів машинного навчання та алгоритмів, які можно застосувати до управління фінансовими ризиками. Від методів керованого навчання, таких як лінійна регресія, логістична регресія та дерева рішень, до методів некерованого навчання, таких як кластерний аналіз і зменшення розмірності, ми дослідили спектр інструментів.

Розділ 3 заглибився в практичні аспекти вибору моделі, оцінки якості та загального процесу прогнозування фінансового ризику. Виявлення факторів, що впливають на вибір моделі, і методів оцінки якості моделі забезпечили міцну основу для подальшої емпіричної роботи. У розділі також розглядалися важливі етапи підготовки даних і створення моделі, наголошуючи на важливості ретельного аналізу та оцінки.

Інтеграція машинного навчання в управління фінансовими ризиками відкриває численні шляхи для дослідження:

- Майбутні дослідження можуть вивчити інтеграцію гібридних моделей, поєднуючи сильні сторони різних алгоритмів машинного навчання для підвищення точності прогнозування та стійкості моделі.

- Застосування машинного навчання в процесі прийняття рішень у режимі реального часу для управління фінансовими ризиками залишається сферою, що розвивається. Майбутні дослідження можуть бути зосереджені на розробці моделей, здатних швидко адаптуватися до динамічних ринкових умов.

- Оскільки машинне навчання продовжує формувати процес прийняття фінансових рішень, етичні міркування та прозорість стають першочерговими. Майбутні дослідження мають заглибитися в етичні наслідки машинного навчання в управлінні фінансовими ризиками та запропонувати основи для відповідального використання.

- Пристосування моделей машинного навчання до конкретних галузей у фінансовому секторі, таких як банківська справа, страхування чи інвестиції, може запропонувати більш цілеспрямовані та ефективні рішення для управління ризиками.

Ця робота внесла свій внесок у розвиток ландшафту управління фінансовими ризиками, подолавши розрив між традиційними методологіями та інноваційними програмами машинного навчання. Викладені тут перспективи служать посібником для майбутніх дослідників і практиків, які прагнуть розвивати цю сферу, зрештою підвищуючи стійкість фінансових систем в економічному середовищі, що постійно змінюється.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The 5% Value at Risk of a hypothetical profit-and-loss probability density function URL: https://en.wikipedia.org/wiki/Value_at_risk#/media/File:VaR_diagram.JPG.

2. The main types of machine learning. URL: https://www.researchgate.net/figure/The-main-types-of-machine-learning-Main-approaches-include-classification-and_fig1_354960266.

3. Linear regression. URL: https://en.wikipedia.org/wiki/Linear_regression#/media/File:Linear_least_squares_example2.svg.

4. Logistic regression. URL: https://ai.plainenglish.io/why-is-logistic-regression-called-regression-if-it-is-a-classification-algorithm-9c2a166e7b74.

5. KNN Diagram. URL: https://www.ibm.com/topics/knn.

6. SVM Margin. URL: https://uk.m.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:SVM_margin.png.

7. Hierarchical Clustering. URL: https://spotintelligence.com/2023/09/12/hierarchical-clustering-comprehensive-practical-how-to-guide-in-python/.

8. k-Means Clustering. URL: https://www.datacamp.com/tutorial/k-means-clustering-python.

9. PCA. URL: https://www.ibm.com/topics/principal-component-analysis/

10. Overfitting and Underfitting. URL: https://www.superannotate.com/blog/overfitting-and-underfitting-in-machine-learning.

11. Machine Learning for Financial Risk Management with Python by Abdullah Karasan / Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. 2022. / 194 p.

12. Probabilistic Machine Learning for Finance and Investing: A Primer to Generative AI with Python 1st Edition by Deepak K. Kanungo / O'Reilly Media; 1st edition (September 19, 2023) / 264 pages.

13. Fundamentals of Machine Learning in Finance. Igor Halperin. URL: https://www.coursera.org/learn/fundamentals-machine-learning-in-finance.

14. Machine Learning in Finance: From Theory to Practice by Matthew F. Dixon, Igor Halperin, Paul Bilokon. Publisher : Springer; 1st ed. 2020 edition (July 2, 2020).

15. Using Machine Learning in Trading and Finance, Jack Farmer. URL: https://www.coursera.org/learn/machine-learning-trading-finance.

16. Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python, Stefan Jansen. Packt Publishing; 2nd ed. edition. 2020.

17. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition. O'Reilly Media; 1st edition (May 9, 2017). 572 p.