МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 43.00.00.000 ПЗ

Група ІПм-23-2

Кіщук Максим

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Кіщук Максим Миколайович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

МАГІСТЕРСЬКА РОБОТА

Інтелектуальні моделі та методи покращення ефективності оцінки кредитних ризиків

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Кіщук М.М.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Бандура Вікторія Валеріївна, к.т.н., доцент

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

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

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

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ 04 ” вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Кіщуку Максиму Миколайовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Інтелектуальні моделі та методи покращення ефективності оцінки кредитних ризиків ”

керівник проекту (роботи) Бандура Вікторія Валеріївна, к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 22 ” листопада 2024 р. № 781/7

2. Строк подання студентом проекту (роботи) 15 грудня 2024 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних технологій інтелектуального аналізу даних

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Дослідження предметної галузі кредитних ризиків в контексті інтелектуальних методів
2. Методи та методології інтелектуального аналізу в контексті роботи з фінансовими даними
3. Інтелектуальні моделі та методологія прийняття рішень для оцінки кредитних ризиків
4. Імплементація інтелектуальних моделей для покращення оцінки кредитних ризиків

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Фінансова модель консалтингової компанії (рис. 1.1)
2. Фреймворк аналітичного ієрархічного процесу Falcon (рис. 1.2)
3. Візуалізація RF алгоритму (рис. 1.3)
4. Пояснення моделі "чорного ящика" на основі SHAP (рис. 2.1)
5. Модель RuleFit (рис. 2.2)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2024 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2024	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2024	виконано
3	Дослідження предметної галузі кредитних ризиків в контексті інтелектуальних методів	15.10.2024	виконано
4	Методи та методології інтелектуального аналізу в контексті роботи з фінансовими даними	08.11.2024	виконано
5	Інтелектуальні моделі та методологія прийняття рішень для оцінки кредитних ризиків	20.11.2024	виконано
6	Імплементація інтелектуальних моделей для покращення оцінки кредитних ризиків	01.12.2024	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2024	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 79 с., 31 рис., 11 табл., 53 джерела.

Тема: Інтелектуальні моделі та методи покращення ефективності оцінки кредитних ризиків

Об'єкт дослідження: процес оцінки кредитних ризиків у фінансових установах за допомогою машинного навчання.

Мета роботи: розробка, імплементація та оцінка ефективності інтелектуальних моделей для покращення оцінки кредитних ризиків, зокрема моделі RuleFit, яка забезпечує високу точність класифікації та дозволяє інтерпретувати правила прийняття рішень.

Предмет дослідження: інтелектуальні моделі та методи для підвищення ефективності та інтерпретованості оцінки кредитних ризиків.

Результати дослідження

В роботі виконано розробку та імплементацію моделі RuleFit, яка поєднує точність класифікації з високою інтерпретованістю рішень, що дозволяє вилучати правила прийняття рішень і робить модель придатною для використання у фінансовій сфері.

Висновок

Реалізація інтелектуальних моделей у поєднанні з експериментальним аналізом підтверджує доцільність використання машинного навчання для оцінки кредитних ризиків. Запропоновані методи підвищують точність, інтерпретованість і стабільність прогнозів, що сприяє надійнішому управлінню кредитними ризиками.

КРЕДИТНИЙ РИЗИК, МАШИННЕ НАВЧАННЯ, ФІНАНСОВИЙ АНАЛІЗ, ІНТЕЛЕКТУАЛЬНІ МОДЕЛІ, ІНТЕРПРЕТОВАНІСТЬ, КЛАСТЕРНИЙ АНАЛІЗ, ОЦІНКА ЕФЕКТИВНОСТІ

ABSTRACT

Master Thesis: 79 pp., 31 fig., 11 tab., 53 sources.

Thesis Subject: Intelligent models and methods of improving the effectiveness of credit risk assessment

The object of the study: the process of credit risk assessment in financial institutions using machine learning.

The purpose of the work: development, implementation and evaluation of the effectiveness of intelligent models to improve the assessment of credit risks, in particular the RuleFit model, which provides high accuracy of classification and allows the interpretation of decision-making rules.

Research subject: intelligent models and methods of improving the efficiency and interpretability of credit risk assessment.

Research results

The work includes the development and implementation of the RuleFit model, which provides classification accuracy with a high interpretation of decisions, which allows the extraction of decision-making rules and makes the model suitable for use in the financial field.

Conclusion

The implementation of intelligent models in combination with experimental analysis confirms the feasibility of using machine learning to assess credit risks. The proposed methods increase the accuracy, interpretability and stability of forecasts, which is more reliable for credit risk management.

CREDIT RISK, MACHINE LEARNING, FINANCIAL ANALYSIS, INTELLECTUAL MODELS, INTERPRETABILITY, CLUSTER ANALYSIS, PERFORMANCE EVALUATION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

LR – Logistic Regression

RF - Random Forest

DT - Decision Tree

MCDA - Multi-Criteria Decision Analysis

AUC – Area Under Curve

BDT – Binary Decision Tree

CV – Cross Validation

ECL – Expected Credit Loss

EDA – Exploratory Data Analysis

FICO – Fair Isaac Corporation (common credit scoring model)

GBDT – Gradient Boosting Decision Tree

GMM – Gaussian Mixture Model

IRB – Internal Ratings-Based approach

KNN – K-Nearest Neighbors

LDA – Linear Discriminant Analysis

LGD – Loss Given Default

LSTM – Long Short-Term Memory

MLP – Multilayer Perceptron

NPL – Non-Performing Loan

RBF – Radial Basis Function

ROC – Receiver Operating Characteristic

SVM – Support Vector Machine

TPR – True Positive Rate

XGB – Extreme Gradient Boosting

RNN – Recurrent Neural Network

ВСТУП

Актуальність теми.

Сучасні фінансові установи стикаються з безперервним зростанням обсягу інформації та посиленням вимог щодо управління ризиками, що обумовлює необхідність у високоточних і надійних моделях для оцінки кредитних ризиків. Кредитні ризики є однією з ключових загроз для фінансової стійкості банків та інших кредитних установ, тому своєчасна і точна оцінка цих ризиків є критично важливою для забезпечення надійності фінансової системи загалом. У зв'язку з цим розробка нових підходів та вдосконалення існуючих методів для оцінки кредитоспроможності позичальників є важливим науковим і практичним завданням.

Незважаючи на активне використання традиційних моделей, таких як логістична регресія, вони мають обмеження в точності прогнозування та складність врахування нелінійних взаємозв'язків між характеристиками позичальників. Інші, більш точні моделі на основі машинного навчання, зокрема Random Forest, часто використовуються як інструменти «чорної скриньки», що обмежує їхню інтерпретованість. Це є серйозною проблемою, оскільки незрозумілість процесу прийняття рішень у таких моделях знижує довіру з боку користувачів та регуляторів. Застосування моделей з високою інтерпретованістю важливе не лише для обґрунтованості та прозорості, але й для відповідності регуляторним вимогам, особливо у фінансових інституціях, де прийняття рішень має бути обґрунтованим та зрозумілим.

З огляду на це, актуальним є використання пояснюваних моделей, таких як RuleFit, яка дозволяє перетворити Random Forest з інструмента «чорної скриньки» на прозору модель. RuleFit дозволяє вилучати зрозумілі правила прийняття рішень, які можна інтерпретувати і використовувати для покращення управління кредитними ризиками. Використання методів скорочення правил, таких як ієрархічний кластерний аналіз (HCA), рекурсивне усунення ознак (RFE), а також фактор інфляції дисперсії (VIF),

дозволяє усунути мультиколінеарність та зменшити обсяг надмірних характеристик, забезпечуючи точніше і зрозуміліше прогнозування.

Таким чином, розробка та впровадження моделі RuleFit для оцінки кредитних ризиків є важливим кроком до підвищення точності та інтерпретованості моделей, що використовується у фінансових установах для зменшення ризиків, підвищення прозорості та довіри до процесу прийняття рішень.

Метою дослідження є розробка, імплементація та оцінка ефективності інтелектуальних моделей для покращення оцінки кредитних ризиків, зокрема моделі RuleFit, яка забезпечує високу точність класифікації та дозволяє інтерпретувати правила прийняття рішень.

Об'єктом дослідження є процес оцінки кредитних ризиків у фінансових установах за допомогою машинного навчання.

Предметом дослідження є інтелектуальні моделі та методи для підвищення ефективності та інтерпретованості оцінки кредитних ризиків.

Задачі дослідження:

1. Дослідити сучасні інтелектуальні моделі для оцінки кредитних ризиків і визначити їх переваги та недоліки.

2. Розробити методологію прийняття рішень з використанням моделі RuleFit для перетворення моделі «чорної скриньки» на інтерпретовану модель.

3. Провести імітаційне моделювання та оцінку ефективності запропонованої моделі на контрольованих і реальних даних.

4. Здійснити порівняльний аналіз моделей (RuleFit, логістична регресія, Random Forest) за основними метриками, включаючи точність, прецизійність, повноту та AUC ROC.

5. Оцінити практичну значущість запропонованої моделі для вирішення бізнес-задач з оцінки кредитних ризиків.

Методи дослідження

Для досягнення поставленої мети використовувались такі методи: ієрархічний кластерний аналіз (HCA) для обробки та скорочення правил прийняття рішень, рекурсивне усунення ознак (RFE) та фактор інфляції дисперсії (VIF) для відбору значущих ознак, а також статистичний аналіз для визначення остаточної групи ознак. Методи машинного навчання включали Random Forest, логістичну регресію та RuleFit.

Наукова новизна дослідження полягає в розробці та імплементації моделі RuleFit, яка поєднує точність класифікації з високою інтерпретованістю рішень. На відміну від стандартних моделей «чорної скриньки», запропонований підхід дозволяє вилучати правила прийняття рішень, що робить модель придатною для використання у фінансовій сфері, де важлива прозорість та зрозумілість рішень.

Практичне значення результатів

Розроблена модель RuleFit може бути використана для покращення процесу оцінки кредитних ризиків у фінансових установах. Її застосування дозволяє підвищити не лише точність, але й інтерпретованість процесу прийняття рішень, що сприяє підвищенню довіри до системи та дозволяє успішно вирішувати задачі управління ризиками.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 79 сторінок, і містить 31 рисунок, 11 таблиць, список використаних джерел із 53 найменувань.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ КРЕДИТНИХ РИЗИКІВ В КОНТЕКСТІ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ

1.1. Особливості фінансової індустрії та передумови кредитного ризику

У фінансовій індустрії позики надаються окремим особам і компаніям щодня, часто для придбання чогось або здійснення інвестицій, які інакше були б недоступні для них. Наприклад, люди щодня запитують іпотеку, щоб купити будинки. Іншим прикладом є те, що компанії просять кредити для розширення своєї діяльності та збільшення прибутків. Організаціями, які видають позики, зазвичай є банки, уряди чи корпорації. Позика, як правило, складається з грошової суми, але є також випадки, коли позикою є машина, яка передається в лізинг. Загалом ми говоримо, що фінансові установи (кредитори) надають різні форми кредиту. Одержувачі кредиту називаються боржниками.

Основна причина, чому фінансові установи надають кредит, полягає в тому, що це створює вхідний потік доходу у формі процентних платежів. Коли боржник отримує кредит від банку, він або вона має повернути кредит з певним відсотком. Основним джерелом доходу комерційних банків є виплата відсотків. Банки, що надають позики компаніям, заохочуватимуть останніх використовувати банк для інших послуг, таких як ощадні рахунки, фінансові консультації, послуги з підготовки податків тощо. Надання позик також має економічні вигоди, оскільки більше компаній можуть стимулювати місцеву економіку, яка, у свою чергу, має економічне зростання самого банку завдяки збільшенню депозитів на ощадних рахунках і більшій стабільності.

Крім того, заощадження - це короткострокові гроші, які банк залучає і зазвичай позичає на більш тривалий період часу. Для суспільства трансформаційна функція спини є надзвичайно важливою. Однак, якщо багато кредиторів не виконають зобов’язань, на які вони погодилися,

приймаючи позику, банк втратить ліквідність, якщо вкладники вирішать вимагати свої гроші назад. Зрештою це може призвести до банкрутства. Тому оцінка кредитного ризику є ключовим завданням банку. Щоб забезпечити фінансову стабільність і, як наслідок, надати ліквідність тим, хто її потребує, тому необхідна оцінка кредитного ризику.

1.2. Поняття фінансової моделі компанії

Рисунок 1.1 показує процес поточної фінансової моделі, що використовується в Zanders, яка називається моделлю Falcon. Zanders - це міжнародна консалтингова компанія, що спеціалізується на фінансовому менеджменті. Вони надають послуги корпораціям, фінансовим установам, державному сектору та некомерційним організаціям. У цьому проекті я зосереджуся на частині червоного прямокутника, яка полягає в пошуку вдосконаленої методології оцінювання функцій, або іншими словами, ваги ознаки.

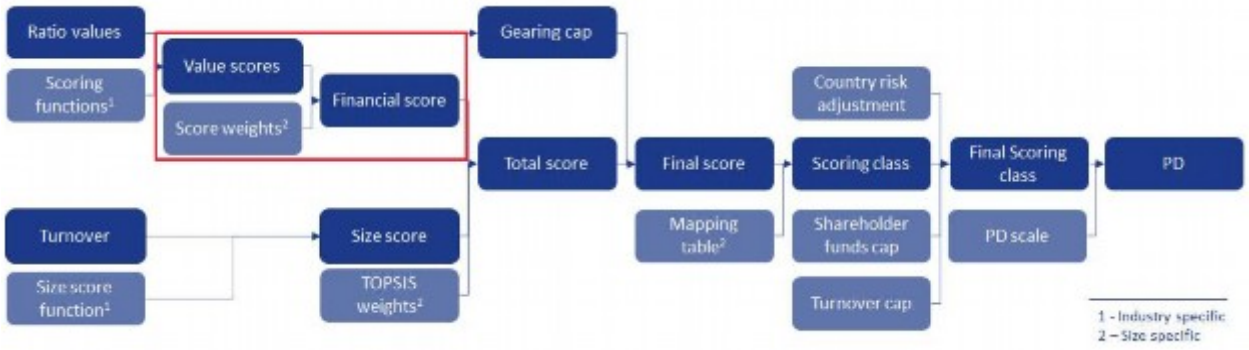


Рис. 1.1. Фінансова модель консалтингової компанії

Наразі вони використовують метод багатокритеріального аналізу рішень (MCDA - Multi-Criteria Decision Analysis), як-от аналітичний ієрархічний процес (AHP - Analytic Hierarchy Process), щоб визначити важливість вибраних функцій від експертів у галузі.

AHP — це метод організації та аналізу складних рішень із використанням математики та психології. Він поділяється на три частини:

1) кінцева мета або проблема,

2) усі можливі рішення, які називаються альтернативами,

3) критерії, за якими оцінюються альтернативи.

Перекладаючи критерії прийняття рішення та потенційні результати в числову форму та пов’язуючи їх з основною метою, AHP пропонує логічну основу для прийняття необхідних рішень. Шляхом попарного порівняння стейкхолдери оцінюють значущість кожного критерію окремо. Наприклад, чи вважаєте ви, що зростання товарообігу чи рентабельність продажів важливіші для дефолту, і наскільки більше? AHP перетворює ці критерії в числа, щоб їх можна було порівняти з усіма відповідними критеріями. AHP виділяється серед інших процесів прийняття рішень завдяки своїм можливостям кількісного визначення. Для кожної з різних можливостей числовий пріоритет визначається на останньому етапі процесу. На основі значень усіх користувачів ці числа вказують на найбільш затребувані рішення.

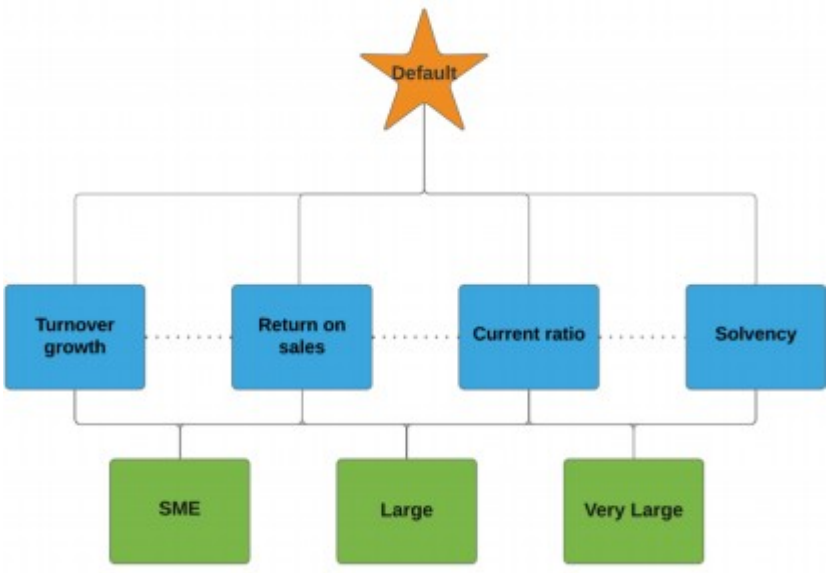


Рис. 1.2. Фреймворк аналітичного ієрархічного процесу Falcon

На рисунку 1.2 показано структуру AHP у моделі Falcon. На третьому рівні, який називається «альтернативи», він представляє розмір підприємств.

«SME» вказує на малі та середні підприємства, «Large» відноситься до великих підприємств, а «Very Large» представляє дуже великі підприємства. На другому рівні він представляє критерії, які були обрані фінансовими експертами. Вони вважають, що ці критерії є найважливішими характеристиками для оцінки ймовірності дефолту. На першому рівні він представляє кінцеву мету — чи відбудеться дефолт.

Як ми бачимо, метод AHP має бути заздалегідь визначений людьми замість самих даних. Крім того, він лише кількісно визначить вагу даних факторів, а не покаже, чи є фактор значущим чи ні. Однак метод машинного навчання, як-от логістична регресія, підхід на основі даних, може ідентифікувати предиктори та оцінити їхні коефіцієнти на основі даних. Більше того, існує кілька тестів, призначених для оцінки значущості окремого предиктора, зокрема тест співвідношення правдоподібності та статистика Вальда. Ось чому я запропонував метод ML замість методології MCDA.

1.3. Існуючі дослідження в області кредитних ризиків з використанням інтелектуальних методів

Оцінка кредитного ризику є однією з перших сфер застосування методів машинного навчання (ML) в економіці. У деяких ранніх дослідженнях оптимізований LR застосовувався до великого набору даних повної історії платежів короткострокових кредитів на виплату [33], методу DT [18], опорних векторних векторів (SVM) [2], нейронних мереж (NN) [40] в оцінці кредитного ризику. Однак, за винятком LR і DT, інші моделі вважаються моделями чорного ящика, і тому їм не вистачає суттєвої характеристики прозорої прогнозної моделі для пояснення причини передбачення. Традиційними моделями білого ящика для завдань класифікації є LR і DT. DT — це інтерпретована модель, яка відображає

судження у формі блок-схеми, що відкрито розуміється та дуже нагадує людське мислення.

LR і DT мають великі зрозумілі таланти, але їхня продуктивність обмежена. В [2] довели, що приріст точності оцінки кредитоспроможності був обмеженим. В результаті застосування методів ансамблю, таких як пакетування та посилення [22], продуктивність моделей оцінки на основі машинного навчання значно покращилася. Ансамблеве навчання [42] є найбільш часто використовуваним методом для покращення ефективності прогнозування моделі. Ансамблеве навчання використовується для покращення ефективності прогнозування однієї прогнозної моделі в задачі прогнозного моделювання.

Випадковий ліс (RF) [28] — це популярна стратегія ансамблевого навчання, яка передбачає навчання багатьох дерев рішень паралельно із початковим завантаженням і агрегацією, відомою як пакетування. RF зазвичай використовується в оцінці кредитного ризику для побудови надійної моделі прогнозування через його високу ефективність.

Ґрунтуючись на порівняльних дослідженнях [3, 34] вони довели, що RF значно перевершує LR та інші моделі, такі як DT і SVM. В індустрії кредитного ризику RF поступово стала однією зі стандартних моделей [26]. Однак, оскільки RF є моделлю чорного ящика, це перешкоджає прозорості та зрозумілості прогнозних моделей. Таким чином, радіочастота не використовується в ситуаціях, коли необхідна прозорість прогнозної моделі. Оцінка кредитного ризику, наприклад, вимагає чіткої системи прийняття рішень для обґрунтування причин схвалення або відхилення заявок. Як наслідок, щоб зрозуміти рішення, необхідна насичена чітка та зрозуміла система з меншою кількістю функцій та обмежень, що робить систему ефективною, переконливою для користувача та значною мірою керованою.

Таким чином, одна з головних проблем методів ML в оцінці кредитного ризику полягає в тому, що вони не піддаються поясненню та інтерпретації. Багато з цих алгоритмів, зокрема методи ансамблю, такі як Random Forest

(RF), розглядаються як «чорна скринька», що означає, що процес затвердження кредиту не є прозорим для клієнтів і регуляторів.

Відповідно, регуляторне співтовариство стурбоване управлінням штучним інтелектом через необхідність інтерпретації, особливо у сфері оцінки кредитного ризику.

Щоб подолати недолік природи чорної скриньки та отримати прозору прогнозну модель, яку можна інтерпретувати, у цій роботі пропонується альтернативна модель, яку можна інтерпретувати під назвою RuleFit, яка побудована на основі LR з розробкою функцій на основі правил. Продуктивність запропонованої моделі порівнюватиметься з LR та RF.

1.4. Методи та методології інтелектуального аналізу в контексті роботи з фінансовими даними

1.4.1. Логістична регресія

Найбільш поширеною моделлю оцінки кредитного ризику є модель LR. Є кілька причин, чому я вибираю LR замість лінійної регресії. Однією з причин є те, що лінійна модель не може вивести ймовірності, тому їх не можна інтерпретувати як ймовірності. Хоча я можу позначити кожен клас цифрами, як-от 1, 2, 3 і так далі, класи можуть не мати жодного значущого порядку, що означає, що лінійна модель накладає дивну структуру на зв’язок між функціями та прогнозами класу. Таким чином, моделі лінійної регресії не підходять для завдання класифікації.

Порівняно з моделлю лінійної регресії, модель LR може впоратися з цим, ввівши логістичну функцію:

$$\text{logistic}(x) = \frac{1}{1 + \exp(-x)}$$

де x — дійсне число. Логістична функція стисне результат лінійного рівняння між 0 і 1.

Однак інтерпретація вагових коефіцієнтів у LR відрізняється від інтерпретації вагових коефіцієнтів у лінійній регресії, оскільки результат у LR є ймовірністю від 0 до 1. Ваги не більше впливати на ймовірність лінійним чином. Тому потрібно переформулювати рівняння, щоб зробити його лінійним:

$$\ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

Щоб знайти кращий метод інтерпретації, можна знайти зв'язок між ймовірністю позитивного класу та ймовірністю негативного класу.

$$\frac{P(y=1)}{P(y=0)} = odds = \exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p\right)$$

У цьому випадку, якщо ми знаємо значення шансів, наприклад 2, тоді ми можемо інтерпретувати, що ймовірність y = 1 вдвічі вища, ніж y = 0 .

Ми також можемо зрозуміти, як змінюється прогноз, якщо одна з числових характеристик x і змінюється на одну одиницю, а всі інші характеристики залишаються незмінними:

$$\frac{\text{odds}_{x_i+1}}{odds_{x_i}} = \frac{\exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_i\left(x_i+1\right) + \ldots + \beta_p x_p\right)}{\exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_i x_i + \ldots + \beta_p x_p\right)} = \exp\left(\beta_i\left(x_i+1\right) - \beta_i x_i\right) = \exp\left(\beta_i\right)$$

Отже, ми можемо інтерпретувати, що зміна x і на одну одиницю збільшує логарифм відношення шансів на значення відповідної ваги в і. Наприклад, якщо вага функції x і дорівнює 0,7, збільшення x і на одну одиницю множить шанси, які в останньому прикладі дорівнюють 2, на exp(0,7) (приблизно 2), а нові шанси дорівнюють 4, тобто що ймовірність y=1 у чотири рази більша за ймовірність y = 0 .

Найбільша перевага LR полягає в тому, що він простий і зрозумілий, особливо в порівнянні з моделями чорних ящиків. Ось чому більшість дослідників хочуть знайти спосіб підвищити точність LR, як, наприклад, останні дослідження [15, 20].

Однак у LR також є багато недоліків. По-перше, точність LR обмежена. В основі LR лежить припущення, що залежна змінна повинна мати лінійну кореляцію з незалежними змінними. Якщо припущення порушується, то LR не може передбачити цілі з хорошою точністю. Що, якщо ми введемо нелінійні функції в LR? Точніше, чи має значення нелінійність у моделюванні кредитного ризику? Автор у [29] перевірив, чи нелінійні кореляції між кредитним ризиком і пояснювальними змінними мають істотний вплив на продуктивність моделі, аналізуючи словенські банківські дані. Згідно з їхніми дослідженнями, моделі нелінійного прогнозування працюють краще, ніж класична модель LR. Їхні висновки демонструють значне збільшення рівня класифікації на 8 % завдяки вдосконаленій обробці нелінійних взаємодій і особливостей категоріальних змінних. Тому має сенс запровадити конструкцію функцій для підвищення продуктивності традиційного LR. По-друге, ідеальне розділення може статися в LR. Модель LR більше не можна навчити, якщо є функція, яка може ідеально розділяти два класи. Тому що оптимальна вага або коефіцієнт для цієї функції буде нескінченним, що означає, що коефіцієнт для цієї функції більше не збігатиметься.

1.4.2. Дерево рішень

Коли функції та результати мають нелінійний зв’язок або коли функції взаємодіють одна з одною, лінійні моделі, такі як LR, обмежені. Однак моделі на основі дерева поділяють дані на різні частини відповідно до граничних значень певних ознак. Набір даних розділено на багато підмножин, причому кожен екземпляр належить до окремої підмножини. Проміжні підмножини відомі як внутрішні вузли або розділені вузли, тоді як

кінцеві підмножини називають кінцевими або кінцевими вузлами. Середній результат даних навчання цього вузла використовується для прогнозування результату кожного листкового вузла.

Для вирощування дерева можна використовувати різні алгоритми. Вони різняться за потенційною структурою дерева, такою як кількість поділів на вузол, методи, що використовуються для ідентифікації поділів, точка, в якій поділ має припинитися, і методи, які використовуються для оцінки основних моделей у листових вузлах. Алгоритм дерев класифікації та регресії (CART) є, ймовірно, найпопулярнішим алгоритмом для індукції дерев.

Наступна формула описує зв'язок між результатом y і ознаками x.

$$\hat{y} = \hat{f}(x) = \sum_{m=1}^{M} c_m I\{x \in R_m\}$$

де M означає наявність M листкових вузлів R, а Rm означає m-й листовий вузол. I {x Є Rm } — функція ідентифікації, яка повертає 1, якщо вибірка x є в підмножині Rm, і 0 в іншому випадку. Якщо екземпляр потрапляє в листовий вузол Rm , прогнозований результат y = cm , де cm — середнє значення всіх навчальних екземплярів у листовому вузлі Rm.

Однак наступне питання полягає в тому, звідки беруться підмножини? Щоб вирішити, яка гранична точка мінімізує індекс Джині розподілу класів у для завдань класифікації, CART бере до уваги одну особливість. Якщо всі класи мають однакову частоту, вузол вважається «нечистим» відповідно до індексу Джіні; однак, якщо присутній лише один клас, вузол вважається максимально чистим. Індекс Джині мінімізується, коли точки даних у вузлах мають дуже схожі значення для y . Як наслідок, найкраща гранична точка робить два отримані підмножини настільки різними, наскільки це можливо, щодо цільового результату. Для категоріальних ознак алгоритм намагається створити підмножини, пробуючи різні групи категорій. Після визначення найкращого відсікання для кожної ознаки алгоритм вибирає функцію для

розбиття, яке призведе до найкращого розбиття з точки зору індексу Джіні, і додає це розбиття до дерева. Алгоритм продовжує цей пошук і розбиття рекурсивно в обох нових вузлах, доки не буде досягнуто критерій зупинки. Прикладами можливих критеріїв є мінімальна кількість екземплярів, які мають бути у вузлі перед поділом, або мінімальна кількість екземплярів, які мають бути у термінальному вузлі.

Деревоподібна структура добре фіксує взаємодію між функціями в даних. Як інтерпретабельну модель інтерпретація досить проста. По-перше, структуру дерева легко візуалізувати з її вузлами та ребрами. Деревоподібна форма природним чином заохочує людей розглядати очікувані значення для конкретних випадків як контрфактичні: «Якби функція була більшою/меншою за точку розділення, тоді прогноз був би уі замість у 2 ». Оскільки ви завжди можете порівняти передбачення екземпляра з відповідними ситуаціями «що, якщо» (як визначено деревом), які є лише іншими листковими вузлами дерева, пояснення дерева є контрастними. Правдивість прогнозу залежить від прогнозної продуктивності дерева. Оскільки кожне розбиття призводить до того, що екземпляр потрапляє або в один, або в інший аркуш, і оскільки двійкові рішення прості для розуміння, пояснення для коротких дерев є простими та універсальними. Крім того, немає необхідності трансформувати функції. У лінійних моделях іноді необхідно логарифмувати ознаку.

Однак дерева не справляються з лінійними зв’язками. Будь-який лінійний зв’язок між вхідною функцією та результатом має бути апроксимований розбиттям, створюючи ступінчасту функцію. Це неефективно. Незначні зміни у функції введення можуть мати великий вплив на прогнозований результат, що зазвичай небажано. Уявіть собі дерево, яке передбачає ймовірність дефолту, і дерево використовує totaLassets як одну з функцій розділення. Якщо розділення відбувається на 100,5 євро. Розглянемо компанію, яка використовує модель дерева рішень: загальна сума активів становить 99 євро для цієї компанії, а ймовірність дефолту становить 60%.

Проте. компанія помітила, що вони забули додати один актив до totalAssets , і вони не впевнені у вартості цього активу. Тож вони намагаються змінити загальні активи до 100 і 101 євро. Прогноз ймовірності дефолту становить 60% і 40%, що є досить неінтуїтивним, оскільки ймовірність дефолту не змінилася з 99 євро до 100 євро. Припущення полягає в тому, що збільшення totalLassets зменшить, а не залишить стандартну ймовірність незмінною.

Крім того, моделі дерев також досить нестабільні. Кілька змін у навчальних даних можуть створити зовсім іншу структуру дерева. Це пояснюється тим, що кожен розділ залежить від батьківського розділення. І якщо іншу функцію вибрано як першу розділену, вся структура дерева змінюється. Це не створює довіри до моделі, якщо структура змінюється так легко.

1.4.3. Випадковий ліс

Random Forest (RF) — це набір класифікаторів дерев. Це метод ансамблевого навчання для класифікації, регресії та інших завдань, який працює шляхом побудови безлічі дерев рішень під час навчання. Деревний ансамбль можна описати такою загальною формулою:

$$\hat{f}(x) = a_0 + \sum_{m=1}^{M} a_m \hat{f}_m(X)$$

Крім того, RF має додаткові характеристики випадкового вибору функцій на кожному вузлі та відсутність правил скорочення або зупинки [1]. Результат RF визначається простою більшістю голосів або простим усередненням результату одного дерева. На рисунку 1.3 показано, як працює RF-алгоритм. У RF кожне дерево росте з використанням зразків, вибраних із оригінального навчального зразка за допомогою технології початкового завантаження. Випадковий вибір ознак у кожному вузлі зменшує кореляцію між деревами в лісі, таким чином зменшуючи рівень помилок лісу. У RF є

два важливі параметри: кількість змінних, попередньо вибраних для вузлів дерев, і кількість дерев у лісі. Перший параметр визначає єдине дерево рішень, а другий параметр визначає загальний розмір RF.

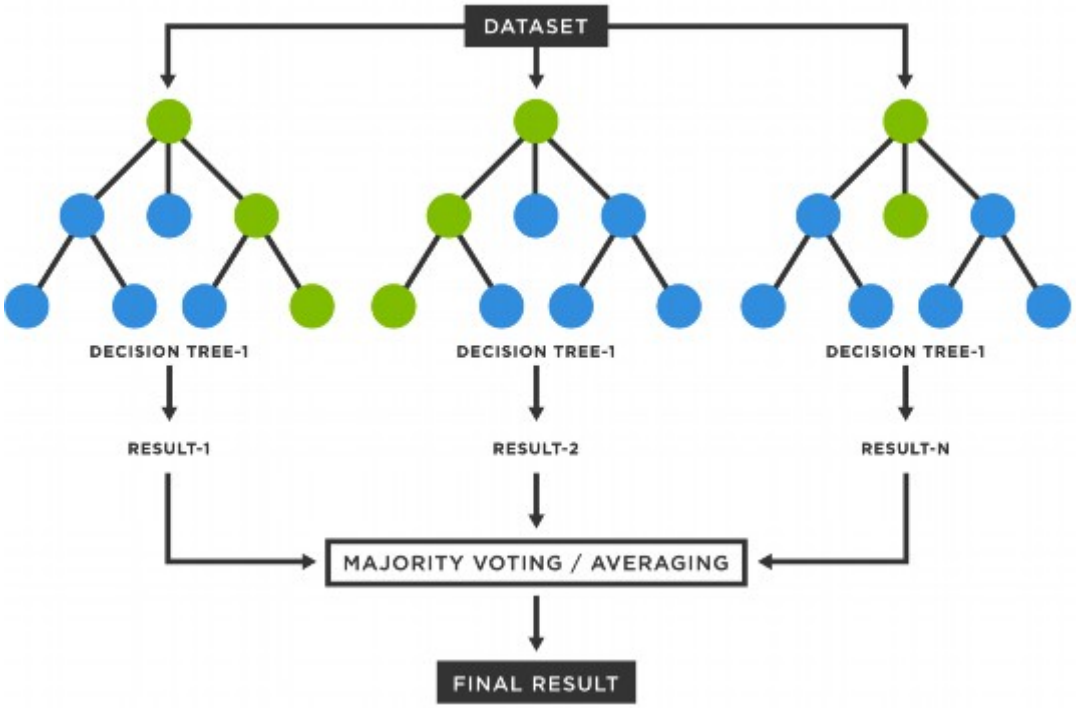


Рис. 1.3. Візуалізація RF алгоритму

Помилка поза мішком [10] використовується для дискримінації продуктивності RF замість виконання перехресної перевірки. Оскільки кожен класифікатор вибирає частину зразків випадковим чином із вихідного набору даних, невідібрані зразки називаються «зразками поза пакетом». Після цього ці відібрані зразки використовуються для навчання окремої моделі, а зразки з мішка використовуються для тестування окремої моделі. Швидкість, з якою зразки неправильно розподіляються між усіма тестовими зразками, називається частотою помилок поза мішком. Крім того, ще однією важливою особливістю в RF є важливість функції. RF-алгоритм оцінює важливість кожної функції, дивлячись на те, наскільки збільшується помилка передбачення, коли вихідні дані для цієї змінної переставляються, а всі інші залишаються незмінними [35].

Важливість функції є ключовою характеристикою для радіочастотних моделей. У ньому описано, які функції є релевантними¬. Це може допомогти з кращим розумінням розв'язаної проблеми та іноді призведе до покращення моделі за допомогою вибору функцій. Існує два способи обчислення важливості функції:

• Важливість Джіні (або середнє зменшення домішок (MDI))

• Важливість функції на основі перестановки

Gini Importance, який є методом за замовчуванням для обчислення важливості функції в RF. Найбільшою перевагою цього методу є швидкість обчислень - усі необхідні значення обчислюються під час навчання. Однак недоліками методу є:

1) Результати зміщені в бік ознак високої потужності. Функція високої потужності означає, що для однієї функції може бути багато можливих значень. Наприклад, якщо ідентифікаційний номер є однією ознакою. ця змінна матиме найбільшу функцію важливість. Оскільки в листових вузлах кожен зразок має свій ідентифікатор і вихід. Але здатність узагальнення моделі дорівнює нулю.

2) Результати обчислюються на основі статистики навчального набору і, отже, не відображають здатність функції бути корисною для створення прогнозів, які узагальнюються на тестовий набір (коли модель має достатню ємність).

Важливість Джіні не може бути обчислена в тестовому наборі, оскільки вона буде обчислена лише під час фази навчання. На етапі прогнозування RF не буде будувати дерева рішень. Таким чином, він не обчислить важливість Джині повторно.

Альтернативним методом є перестановка важливості ознаки [23], яка є технікою перевірки моделі, яку можна використовувати для табличних даних. Важливість ознаки перестановки визначається як зменшення оцінки моделі, коли одне значення функції випадково перемішується [10], що

означає, що воно перемішує певний стовпець, тому значення цього стовпця більше не є важливим під час обчислення важливості функції цього колонка.

Іншими словами, я вимірюю важливість функції, обчислюючи збільшення помилки прогнозування моделі після перестановки функції. Функція є «важливою», якщо перетасування її значень збільшує помилку моделі, тому що в цьому випадку модель покладалася на функцію для прогнозування. Функція є «неважливою», якщо перетасування її значень залишає помилку моделі незмінною, оскільки в цьому випадку модель проігнорувала функцію для прогнозу.

Algorithm 1 Permutation Importance Algorithm

Input the fitted model $\hat{f}$, feature matrix X, target y and loss function $L(y, \hat{f})$.
Output the sorted importance value.

1: Compute the original model error $\epsilon = L(y, \hat{f}(X))$.
2: **for each column** j **do**
3: Randomly shuffle column j to generate a new feature matrix X_j.
4: Estimate the error $\epsilon_j = L(y, \hat{f}(X_j))$ based on the predictions of the permuted data.
5: Calculate permutation feature importance: ϵ_j/ϵ
6: **end for**
7: Sort features by descending feature importance.

Важливість функції на основі перестановки може уникнути проблеми Gini Importance, яка надає високу важливість функціям, які можуть бути непередбачуваними на невидимих даних, коли модель переобладнана. Оскільки важливість перестановки може бути обчислена за даними, які раніше не були використані, використання тестового набору дає змогу виділити, які функції найбільше сприяють узагальнюючій здатності досліджуваної моделі.

Однак одним із недоліків є те, що якщо я додам пов’язану функцію, це може зменшити важливість пов’язаної функції, що означає, що важливість між обома функціями буде розділена. Наприклад, якщо я хочу передбачити ймовірність дефолту та використовувати loans_0 як одну функцію разом з іншими некорельованими функціями. Я навчаю RF-модель, і виявляється, що loans_0 є найважливішою функцією. Якщо я додатково включаю займи_1 як

нову функцію, яка тісно пов’язана з займами_0 . Позики_1 можуть не надати мені багато додаткової інформації, якщо я вже знаю позики_0 . Але ми знаємо, що мати більше функцій – це завжди добре. Отже, я знову треную RF-модель із цими двома функціями позик та іншими некорельованими функціями. Нарешті, деякі дерева рішень у РФ виберуть loans_0 , інші – loans_1 . Дві функції позики разом мають дещо більше значення, ніж функція окремої позики раніше, але замість того, щоб бути у верхній частині списку важливих функцій, кожна функція позики тепер знаходиться десь посередині. Я пропустив найважливішу функцію з вершини драбини важливості, представивши корельовану функцію. Це розумно, тому що воно просто відображає поведінку моделі РФ. Loans_0 просто стала менш важливою функцією , оскільки модель тепер може покладатися на позики_1 також.

Однак це значно ускладнює інтерпретацію важливості ознаки. Наприклад, у першому випадку я можу перевірити, що найголовніші важливі функції містять loans_0 , але в другому випадку я не можу знайти жодну функцію позик лише тому, що loans_0 і loans_1 зараз мають спільну важливість.

Інший недолік полягає в тому, що коли дві функції корельовані, а одна з них переставлена, модель все одно матиме доступ до функції через її корельовану функцію. Це призведе до нижчого значення важливості для обох функцій, де вони насправді можуть бути важливими.

Один із способів впоратися з цим — кластеризувати корельовані функції та зберігати лише одну функцію з кожного кластера. Ця стратегія представлена далі в роботі.

Висновки до розділу

В даному розділі проведено комплексний аналіз теоретичних аспектів та практичних підходів до оцінки кредитних ризиків із застосуванням інтелектуальних методів.

Дослідження показали, що інтелектуальні методи, зокрема машинне навчання та методи аналізу великих даних, є ефективними для оцінки кредитного ризику. Використання цих методів дає можливість враховувати велику кількість змінних та розширює можливості для автоматизації процесів оцінки ризиків. Застосування різних методів машинного навчання дозволяє підвищити точність оцінки кредитних ризиків:

Логістична регресія є одним із базових методів для прогнозування ймовірності дефолту. Її застосування дозволяє з високою точністю передбачити ризик неплатежів, що зумовлено її здатністю обробляти числові дані і визначати зв'язки між змінними.

Дерево рішень є інтуїтивно зрозумілим методом, який забезпечує високу інтерпретованість результатів і дозволяє ідентифікувати ключові фактори ризику. Завдяки цьому його часто використовують для побудови базових моделей оцінки ризику.

Випадковий ліс є одним із найбільш точних методів, що базується на побудові ансамблів дерев рішень і забезпечує високу стійкість до аномалій. Цей метод дозволяє суттєво підвищити точність прогнозування ризиків порівняно з традиційними методами.

Загалом, результати дослідження предметної галузі кредитних ризиків свідчать про доцільність впровадження інтелектуальних методів у процесі оцінки ризиків для покращення точності та зниження суб’єктивності оцінки. Інтелектуальні методи забезпечують адаптивність моделей та можливість врахування численних факторів, що робить їх важливим інструментом для фінансових установ у прийнятті рішень щодо кредитування

РОЗДІЛ 2. ІНТЕЛЕКТУАЛЬНІ МОДЕЛІ ТА МЕТОДОЛОГІЯ ПРИЙНЯТТЯ РІШЕНЬ ПОКРАЩЕННЯ ЕФЕКТИВНОСТІ ОЦІНКИ КРЕДИТНИХ РИЗИКІВ

2.1. Концепція пояснюваного штучного інтелекту

На відміну від інтерпретованих моделей, які також називають "моделями білого ящика", моделі "чорного ящика" надзвичайно важко пояснити, і їх важко зрозуміти фахівцям у предметній області. Однак, пояснюваний штучний інтелект (XAI) може допомогти користувачам зрозуміти, чому і як ШІ приймає рішення.

Мета XAI полягає в тому, щоб пояснити, що було зроблено, що робиться зараз, що буде зроблено далі, і на якій інформації ґрунтуються дії.

Інтерпретованість методів машинного навчання можна класифікувати як внутрішню або постфактумну. Внутрішня інтерпретованість спрямована на інтерпретовані моделі, такі як логістична регресія (LR) та дерева рішень (DT). Постфактумна інтерпретованість спрямована на моделі "чорного ящика", такі як ансамблеві моделі та нейронні мережі.

Ці моделі можна пояснити лише після навчання моделі. У цьому розділі я обговорюватиму лише постфактумну інтерпретованість. Метод постфактум, також відомий як модель-агностичний метод, може бути використаний на будь-якій моделі машинного навчання та застосований після того, як модель була навчена. Зокрема, я використовуватиму Shapley Additive Explanations (SHAP) для пояснення всіх моделей.

2.1.1 Підхід теорії ігор SHAP

SHAP - це підхід теорії ігор для пояснення виходу будь-якої моделі машинного навчання. Він пов'язує оптимальний розподіл внесків з локальними поясненнями, використовуючи класичні значення Шеплі з теорії ігор та їхні пов'язані розширення. Мета SHAP - пояснити прогнозування

зразка x шляхом обчислення внеску кожної ознаки в прогнозування y. Рисунок 2.1 показує, як SHAP пояснює модель "чорного ящика".

Рис. 2.1. Пояснення моделі "чорного ящика" на основі SHAP

Формулу SHAP можна визначити так:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

де g – модель пояснення, z' Є { 0,1 } M – спрощені ознаки, 1 означає, що відповідне значення ознаки «присутнє», а 0, що воно «відсутнє». M – кількість виділених ознак. ϕ_j - значення Шеплі ознаки j.

2.1.2. Переваги та недоліки

SHAP - це гібридний метод, який поєднує локальні інтерпретовані модель-агностичні пояснення (LIME) та значення Шеплі. Він може пояснити модель з глобальної та локальної точки зору, що означає, що він може допомогти нам зрозуміти деякі питання, такі як "Як навчена модель робить прогнози?" та "Чому модель зробила певне прогнозування для конкретного зразка?".

Більше того, SHAP має швидку реалізацію для моделей на основі дерев (TreeSHAP). Це вирішує найбільший бар'єр, який полягає в обчисленні

значень Шеплі. Завдяки швидким обчисленням, він може обчислити багато значень Шеплі, необхідних для глобальних інтерпретацій моделі.

Однак недоліком TreeSHAP є те, що він генерує неінтуїтивні атрибуції ознак. TreeSHAP вирішує проблему екстраполяції на малоймовірні точки даних, але робить це шляхом зміни функції значення, що дещо змінює гру. Якщо ознаки не впливають на прогнозування, то їх значення Шеплі має бути 0. Але тепер ці ознаки, що не впливають, отримають значення TreeSHAP замість 0.

2.2. Пропонована методологія прийняття рішень з використанням моделі RuleFit

У цьому розділі буде досліджено методологію запропонованого методу. Моделі RuleFit бувають двох різних типів. Перша модель є базовою моделлю, яка є LR + DT, а запропонована модель – LR + RF. Правила прийняття рішень для базової моделі походять від DT, тоді як правила прийняття рішень для запропонованої моделі походять від RF.

Моделі RuleFit - це метод машинного навчання, який поєднує в собі прогнозну силу дерев рішень та інтерпретованість лінійних моделей.

Алгоритм роботи:

- Створення правил: Спочатку будується ансамбль дерев рішень (наприклад, випадковий ліс). Кожне дерево розбиває дані на підмножини на основі певних правил (наприклад, "якщо вік > 30 і дохід < 50 000, то..."). Ці правила витягуються з дерев.

- Лінійна комбінація: Правила, витягнуті з дерев, перетворюються на нові ознаки. Потім будується лінійна модель (логістична регресія або лінійна регресія), яка використовує як оригінальні ознаки, так і нові ознаки, отримані з правил. Коефіцієнти в лінійній моделі показують важливість кожної ознаки та правила.

Як обговорювалося в першому розділі, хоча логістична регресія (LR) є хорошою інтерпретованою моделлю, вона не може обробляти взаємодії функцій. Дерево рішень (DT) є порівнянною інтерпретованою моделлю, але на відміну від лінійної регресії (LR), воно не в змозі врахувати лінійні зв'язки між x і y. Таким чином, інтуїція полягає в пошуку альтернативної моделі, яка поєднує в собі переваги LR і DT. Отже, чи існує такий же простий і придатний для інтерпретації алгоритм, як моделі LR, але який також інтегрує взаємодію функцій, як DT? Так, є. Це модель RuleFit, запропонована [24].

Модель RuleFit вивчає розріджену лінійну модель з оригінальними функціями, а також низкою нових функцій, які є правилами прийняття рішень. Найбільша різниця між моделлю RuleFit і DT полягає в тому, як вони генерують правила прийняття рішень. Для DT він генеруватиме правила від кореневого вузла до кінцевого вузла. Однак для моделі RuleFit воно генеруватиме правило не лише від кореневого вузла до кінцевого вузла, але й від кореневого вузла до некінцевого вузла.

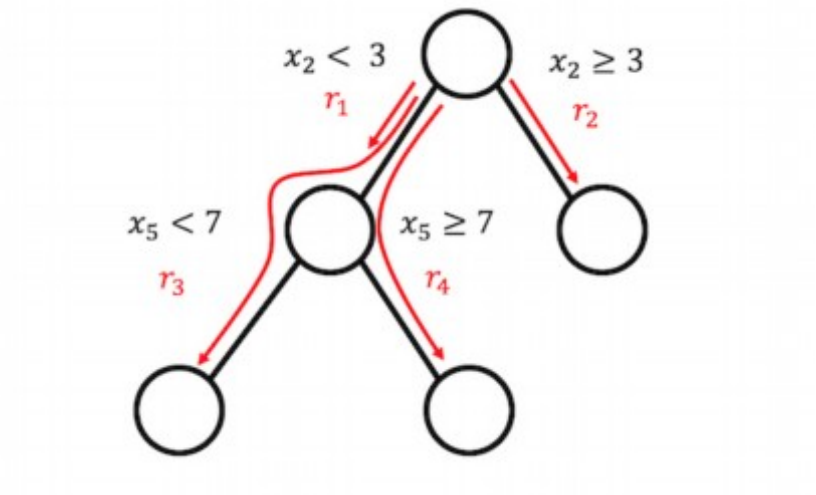


Рис. 2.2. Модель RuleFit

Як показано на рисунку 2.2, у цьому бінарному дереві всього чотири правила. Якщо метод DT, то є тільки 3 правила: r 3 , r 4 і r 2 . Якщо метод RuleFit, то існує 4 правила: ri, r 2 , r 3 і r 4 . У цьому випадку кожне правило можна розглядати як нову функцію взаємодії на основі оригінальних

функцій. Наприклад, якщо початкова функція – це x 2 і x 5 , нова функція взаємодії буде «x 2 < 3 і x 5 < 7» або «x 2 < 3 і x 5 > 7». Значення кожної нової функції дорівнює 0 або 1. Отже, усі ці нові функції є двійковими.

Якщо бути точніше, RuleFit складається з двох кроків: перший крок полягає у створенні «правил прийняття рішень» з DT або інших алгоритмів на основі дерева, таких як Random Forest (RF). Другим кроком є навчання лінійної моделі з оригінальними функціями та скороченими правилами. У цій роботі я побудую дві моделі RuleFit: базову модель, яка складається з LR і DT, і іншу, яка складається з LR і RF.

Найбільшою перевагою є те, що RuleFit автоматично додає взаємодії функцій до лінійних моделей, що вирішує проблему лінійних моделей, де нам доводиться додавати умови взаємодії вручну, і це трохи допомагає з проблемою моделювання нелінійних зв'язків.

Крім того, створені правила легко інтерпретувати, що подібно до DT. Але хороша інтерпретація гарантується, лише якщо кількість умов у межах правила не надто велика. Нам здається розумним правило з умовами від 1 до 3, що означає максимальну глибину 3 для дерев у ансамблі дерев. Навіть якщо в моделі є багато правил, вони не застосовуються до кожної вибірки, лише коли вони мають ненульові ваги. Це покращує локальну інтерпретацію та відфільтровує непотрібні правила.

Однак недоліком є те, що його важко інтерпретувати, коли у вас є правила, що збігаються. Наприклад, якщо є правило:

вік > 40

і інше правило може бути

вік > 30 і зарплата = 5000

Якщо зарплата 5000 і вік більше 30 років, то виконується і правило про вік більше 40 років. У випадку, коли застосовується друге правило, також застосовується перше правило. Інтерпретація розрахункової ваги для другого

правила така: «Якщо припустити, що всі інші характеристики залишаються фіксованими, прогнозована ймовірність дефолту збільшується на β_i , коли зарплата становить 5000 і вік вище 30». Якщо перше правило також застосовується, тлумачення є нелогічним.

2.3. Метрики та моделі оцінювання машинного навчання

Показники оцінювання використовуються для вимірювання якості статистичної моделі або моделі ML. Оцінка моделей або алгоритмів ML є важливою для будь-якого проекту. Існує багато типів оціночних показників, доступних для тестування моделі. У моделюванні кредитного ризику дефолту більшість досліджень використовує матрицю невідповідностей, середню геометричну оцінку, точність, відкликання, оцінку F1, оцінку ROC AUC і оцінку Брієра як метрики для оцінки ефективності моделей дефолту кредитного ризику.

Дуже важливо аналізувати моделі за допомогою різноманітних оціночних показників. Це пов’язано з тим, що модель може працювати добре, коли використовується вимірювання з одного показника оцінки, але погано, якщо використовується інше вимірювання з тієї самої метрики оцінювання. Щоб переконатися, що модель працює належним чином та ідеально, метрики оцінки є важливими.

2.3.1. Матриця невідповідності

Матриця невідповідності дає нам матрицю як результат і описує повну продуктивність моделі. Він може представляти всі прогнози (правильні та неправильні) у матриці. Кожна комірка підраховує, скільки зразків належать до класу або будь-якого іншого класу.

У цій роботі позитивний клас відноситься до класу за замовчуванням, а негативний — до класу, що не є за замовчуванням. Тому True Positive (TP) вказує на те, що модель правильно передбачає позитивний клас (за

замовчуванням), True Negative (TN) вказує на те, що модель правильно передбачає негативний клас (не за замовчуванням), False Negative (FN) вказує на те, що модель неправильно прогнозує не дані за замовчуванням, а помилковий результат (FP) вказує на те, що модель неправильно передбачає дані за замовчуванням.

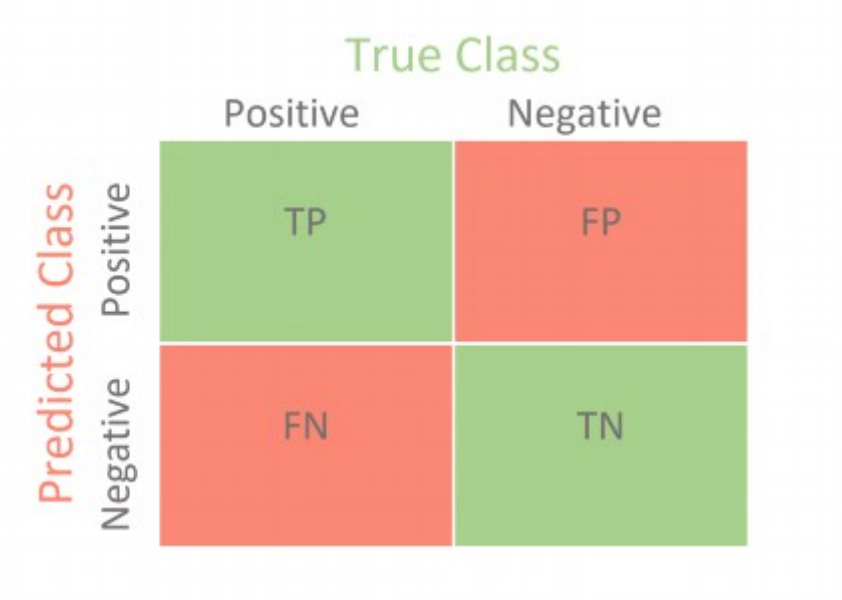


Рис. 2.3. Матриця невідповідності

2.3.2. Точність

Точність розраховується як відношення між кількістю позитивних зразків, правильно класифікованих, до загальної кількості зразків, класифікованих як позитивні (правильно чи неправильно). Точність вимірює точність моделі в класифікації зразка як позитивного. Математична формула:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Однак Precision лише відображає, наскільки надійна модель у класифікації зразків як позитивних. Іншими словами, якщо значення точності моделі дорівнює 0,93, це означає, що відсоток надійності моделі, коли вона каже, що зразок є позитивним, становить 93%.

2.3.3. Повнота (Recall)

Повнота розраховується як співвідношення між кількістю позитивних зразків, правильно класифікованих як позитивні, до загальної кількості реальних позитивних зразків. повнота вимірює здатність моделі виявляти типові зразки. Чим вище значення повноти, тим більше типових зразків виявляється. Коли Recall високий, це означає, що модель може правильно класифікувати більшість типових зразків. Таким чином, моделі можна довіряти в її здатності виявляти типові зразки. Однак все ще може бути багато зразків, які не є типовими, класифікованими як стандартні. Повнота не враховує цього. Формула обчислення Recall така:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Підсумовуючи, коли модель має високий рівень запам’ятовування, але низьку точність, тоді модель правильно класифікує більшість зразків за замовчуванням, але вона має багато FP (тобто класифікує багато зразків, які не є типовими, як типові). Коли модель має високу точність, але низький рівень запам’ятовування, тоді модель є точною, коли класифікує зразок як стандартний, але може класифікувати лише кілька позитивних зразків.

2.3.4. Середня геометрична оцінка

Через те, що я застосував лише SMOTE-TomekLinks до навчального набору, набори перевірки та тестування незбалансовані. Таким чином, баланс між продуктивністю класифікації як для більшості, так і для меншості класів вимірюється геометричним середнім (G-Mean). G-середнє намагається максимізувати точність кожного з класів, зберігаючи при цьому ці точності збалансованими. Навіть якщо негативні випадки правильно класифіковані як такі, низьке G-Mean вказує на низьку ефективність у класифікації позитивних випадків. Формула обчислення середнього G:

$$\text{G-Mean} = \sqrt{Recall * Specificity}$$

де специфічність:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

2.3.5. Оцінки ROC AUC, F1

Крива робочих характеристик приймача (ROC) — це діаграма, яка візуалізує компроміс між істинним позитивним результатом (TPR), який є повнотою і помилковим позитивним коефіцієнтом (FPR). По суті, для кожного порогового значення ми обчислюємо TPR і FPR і наносимо їх на одну діаграму.

$$\text{FPR} = \frac{FP}{FP + TN}$$

Площа під кривою (AUC) – це показник ROC моделі, який обчислюється на основі оцінок прогнозу. Будь-який класифікатор, що йде за лінією 45 градусів, вважається марним класифікатором. Ідеальний класифікатор класифікує користувача за замовчуванням як «за замовчуванням» у 100% випадків, тоді як продуктивність реального класифікатора знаходиться десь між марними та ідеальними класифікаторами. Природно, що вищий TPR і нижчий FPR для кожного порогу, тим краще. Тому класифікатори з більшою кількістю верхніх лівих кривих є кращими.

Оцінка F1 є середньозваженим значенням точності та запам’ятовування. Тому ця оцінка враховує як помилкові позитивні, так і помилкові негативні результати. Він в основному використовується для порівняння продуктивності двох класифікаторів. Припустимо, що класифікатор A має вищу пам’ятність, а класифікатор B має вищу точність. У

цьому випадку Fl-оцінки для обох класифікаторів можна використовувати, щоб визначити, який із них дає кращі результати. Математична формула:

$$F1\ \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Оцінка Брієра спочатку була запропонована в роботі про перевірку прогнозів погоди, окресленої ймовірностями. Вона оцінює точність прогнозів ймовірності. Якщо у нас є дві моделі, які правильно передбачили дефолт. Одна з імовірністю 0,51, а інша – 0,93. Обидва вони правильні та мають однакову точність (за умови порогового значення 0,5), але друга модель зробить вас більш впевненими.

Модель з нижчою оцінкою Браєра точніше прогнозує результат. Для бінарної класифікації оцінка Брієра надається таким чином:

$$\mathcal{L}_{\text{Brier}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - y_i)^2$$

де p_i – прогнозована ймовірність, а y_i – фактичний результат.

2.4. Запропонована структура підходу обробки фінансових даних

На рисунку 2.4 показано структуру запропонованого підходу. Спочатку я маю необроблені фінансові дані від Zanders, потім необроблені дані будуть оброблені модулем обробки даних.

Метою обробки даних є відновлення будь-яких даних, які відсутні, містять викиди, мають нерівний розподіл і незбалансований клас. Після того, як я отримаю попередньо оброблені дані, усі дані будуть розділені на 60% даних навчання, 20% даних перевірки та 20% даних тестування.

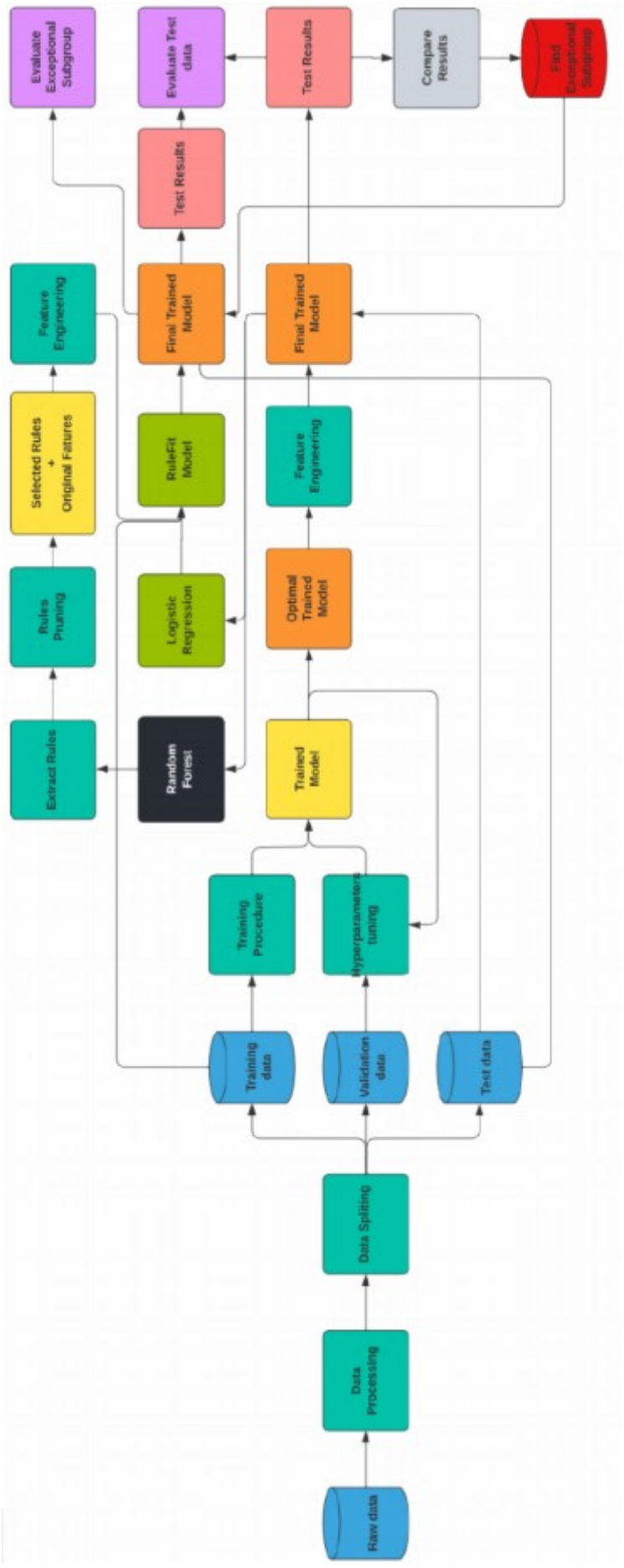


Рис. 2.4. Блок-схема, що показує роботу запропонованого методу

Навчальні дані використовуватимуться для навчання моделей логістичної регресії (LR), випадкового лісу (RF), дерева рішень (DT) і RuleFit. Навчання моделі DT є необхідним, оскільки вона служить базовою лінією для RuleFit (LR+DT) у порівнянні з RuleFit (LR+RF), а навчання моделі LR є необхідним, оскільки вона вважається базовою моделлю. Також потрібне навчання RF, оскільки це еталонна модель. Дані перевірки використовуватимуться для налаштування гіперпараметрів. Дані тестування будуть використані для перевірки ефективності 4 остаточних моделей.

Вкладена перехресна перевірка (Nested CV), метод, який поєднує процедуру навчання та налаштування гіперпараметрів, часто використовується для навчання моделей, які також потребують оптимізації гіперпараметрів. Після отримання моделі з оптимальними гіперпараметрами кожна оптимальна модель буде навчена на 80% даних навчання (60% даних навчання + 20% даних перевірки).

Крім того, процедура розробки ознак застосовується до навчених оптимальних моделей. Потім кожна модель буде перенавчена з вибраними функціями. Нарешті, тестові дані будуть використані для оцінки ефективності кожної кінцевої моделі. Однак процес навчання для моделі RuleFit буде відрізнятися від процесу для моделей LR, DT і RF, оскільки він вимагає правил прийняття рішень моделі на основі правил, наприклад DT і RF. Отже, коли я маю остаточну навчену модель DT і RF, мені потрібно витягти правила прийняття рішень, а потім я повинен скоротити правила, щоб позбутися тих, які є зайвими та ідентичними. Я інтегрую вибрані правила з оригінальними функціями, щойно їх отримаю, а потім застосую до них розробку функцій. Наступним етапом є навчання моделі LR з використанням 80% даних навчання з вибраними змішаними функціями, а потім оцінка продуктивності за допомогою тестових даних.

На рисунку 2.4 показано ще одну цікаву річ, виняткові дані підгрупи, які є різницею між результатами прогнозування LR і RF. Навіщо мені це вивчати? Однією з причин є те, що мені цікаво дізнатися про характеристики

зразків, які LR не може точно класифікувати, але RF може. Якщо я зможу зрозуміти це, я можу дізнатися, чому LR і RF відрізняються один від одного. Крім того, якщо запропонована модель RuleFit, LR зі змішаними функціями, може правильно класифікувати більшість даних із виняткової підгрупи, тоді ми можемо зробити висновок, що вона дійсно може підвищити продуктивність моделі LR.

2.5. Аналіз, попередня обробка та візуалізація фінансових даних

Дані, використані для роботи це реальний набір даних, який охоплює широкий спектр галузей. Необроблений набір даних містить понад 1 693 577 зразків із 112 характеристиками. Пояснення базується на фінансових даних 514 488 компаній з 2011 по 2018 рік. Змінна відповіді (або залежна змінна) — це default_indicator , який вказує, чи є компанія в дефолті чи ні в певний рік. Необроблений набір даних — це величезний незбалансований набір даних. Майже 99% даних не є типовими, і лише 1% даних є типовими. Тому на етапі навчання необхідно застосовувати техніку недостатньої та надлишкової вибірки. Рисунок 2.5 показує розподіл статусного року між компаніями, що не є дефолтом, і компаніями, які не є дефолтом.

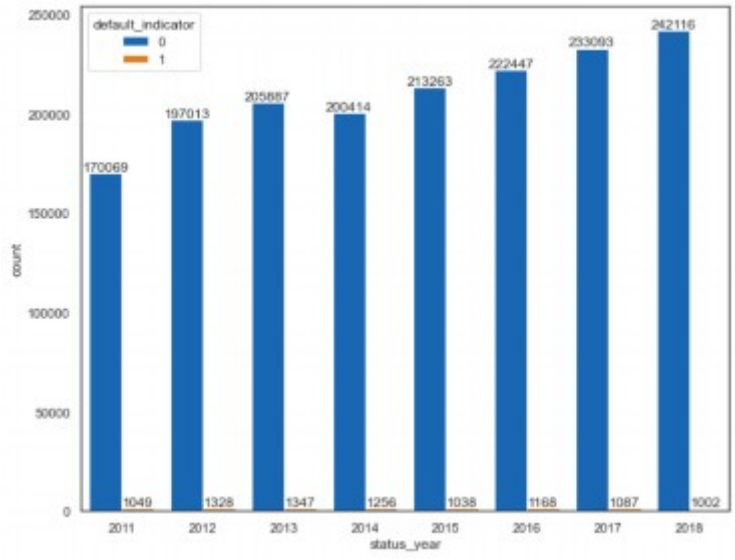


Рис. 2.5. Розподіл вибірок за замовчуванням і не за замовчуванням на основі кожного року

В таблиці 2.1 показано нерівність числових характеристик у необробленому наборі даних. Асиметрія turnover_growth_2 , stock_days_2 , debitor_days_2 , creditor_days_2 дорівнює NaN, оскільки з деяких причин дані повністю відсутні. Значення асиметрії більше 1 або менше -1 вказує на сильно спотворений розподіл. Значення від 0,5 до 1 або від -0,5 до -1 є помірно спотвореним. Значення від -0,5 до 0,5 вказує на те, що розподіл досить симетричний. Як видно з таблиці 2.1 , майже всі функції сильно спотворені.

Таблиця 2.1.

Асиметрія первинних даних

Features	skewness	Features	skewness
capital_employed_1	1298.822	return_on_sales_2	299.430
noncurrent_liabilities_0	1298.815	current_liabilities_2	299.199
operating_revenue_1	1298.736	current_assets_2	285.094
shareholders_funds_1	1298.710	loans_2	258.939
total_assets_1	1297.910	solvency_2	253.717
operating_pl_ebit_1	1296.712	stock_days_1	240.907
debtor_days_0	1296.647	total_assets_2	221.014
total_debt_0	1295.080	total_assets_0	217.738
noncurrent_liabilities_1	1294.281	debtor_days_1	187.124
tangible_net_worth_1	1293.982	return_on_sales_1	160.247
loans_1	1290.840	noncurrent_liabilities_2	158.012
pl_for_period_net_income_1	1290.378	total_debt_2	155.365
current_assets_1	1288.821	interest_coverage_ratio_1	124.551
liquidity_1	1287.718	intangible_fixed_assets_0	124.396
total_debt_1	1287.206	stock_2	123.089
current_liabilities_1	1286.966	intangible_fixed_assets_2	117.395
debtors_1	1286.381	stock_0	116.020
stock_1	1262.856	operating_revenue_2	105.057
intangible_fixed_assets_1	1245.696	operating_revenue_0	99.704
creditors_1	1234.940	capital_employed_2	90.901
gearing_1	1231.545	operating_pl_ebit_0	89.175
gearing_0	1212.654	operating_pl_ebit_2	79.324
interest_paid_0	1183.818	pl_for_period_net_income_0	74.724
turnover_growth_1	1155.629	pl_for_period_net_income_2	72.072
ebitda_0	1141.933	gross_profit_0	71.846
gearing_2	1139.659	ebitda_1	69.069
interest_paid_2	1055.874	ebitda_2	68.195
costs_of_goods_sold_1	793.019	capital_employed_0	68.077
gross_profit_1	790.640	costs_of_goods_sold_0	59.766
return_on_capital_employed_0	779.055	costs_of_goods_sold_2	58.455
creditor_days_0	712.039	return_on_sales_0	41.761

Деякі моделі ML вимагають, щоб усі вхідні та вихідні змінні були числовими, а це означає, що якщо набір даних містить категоричні дані, ці

дані мають бути закодовані в числа, перш ніж ми підберемо та оцінимо модель. Таким чином, кодування є обов'язковим етапом попередньої обробки під час роботи з категоріальними даними для алгоритмів машинного навчання. Існує три широко використовуваних методи: порядкове кодування, кодування з одним кроком і цільове кодування.

• Для порядкового кодування кожній категорії просто призначатиметься ціле число. Однак це корисно, лише якщо існує природний порядок у категоріях.

• Для одноразового кодування він просто додасть нову функцію 0/1 для кожної категорії, маючи 1, якщо зразок має таку категорію. Але якщо функція має багато значень, це спричинить ¬проблему великого розміру.

• Для цільового кодування буде створено новий стовпець для кожної функції. Закодоване значення — це ймовірність того, наскільки воно близьке до класу 1 . Це добре для багатьох значень категорій, оскільки створюється лише одна нова функція для кожної змінної. Але обмеження полягає в тому, що його можна використовувати лише для двійкової класифікації.

У моєму наборі даних категоричні стовпці включають country_code, industry_code, size_class. Усі вони не містять жодного пропущеного значення, тому я можу безпосередньо виконати перетворення даних. Для size_class, який вказує на розмір компанії, є лише три різні значення: Large, SME та Very large. Таким чином, я можу просто перетворити категоріальну змінну на фіктивні змінні, що також називається одноразовим кодуванням. Однак існує пастка під назвою Dummy Variable Trap.

Це відбувається, коли кількість створених фіктивних змінних дорівнює кількості значень категоріального значення, і це призводить до мультиколінеарності, що спричиняє неправильні розрахунки коефіцієнтів регресії та p-значень. Наприклад, розглянемо модель множинної лінійної регресії так:

$$y = \beta_0 + \beta_1 x_{\text{Large}} + \beta_2 x_{\text{SME}} + \beta_3 x_{\text{Very large}} + \epsilon$$

де y – змінна відповіді, x Large , x SME та x Very large – пояснювальні змінні, p 0 – відрізок, ві, в 2 і вз – коефіцієнти регресії, є – член помилки.

Оскільки ці три фіктивні змінні є мультиколінеарними, отже, ми знаємо, що якщо розмір компанії large, то він не може бути SME і Very large, що означає, що я можу замінити x Large на (1 — x SME — x Very large) у множинній лінійній регресії рівняння. Як видно, можна переписати рівняння регресії, використовуючи лише x SME та x Very large, де нові коефіцієнти, які потрібно передбачити, є ($в_0$ + $в_1$), ($в_2$ – $в_1$) і ($в_з$ – $в_1$). Ми можемо уникнути пастки, видаливши стовпець фіктивної змінної.

Для country_code та industry_code є 49 різних значень у коді країни та 19 окремих значень у коді галузі. Тому, оскільки це проблема двійкової класифікації, я буду застосовувати цільове кодування, а не одноразове кодування, щоб запобігти проблемам з великою розмірністю.

Формула цільового кодування:

$$\mathrm{Enc}(i) = \frac{1}{1+e^{-(n_i-1)}}\frac{n_{iY}}{n_i} + \left(1 - \frac{1}{1+e^{-(n_i-1)}}\right)\frac{n_Y}{n}$$

де n_{iY} – кількість зразків з категорією і та класом Y =1, n* – кількість зразків з категорією і. n Y – кількість зразків із класом Y =1, n – загальна кількість зразків у наборі даних.

Числові вхідні змінні можуть мати сильно викривлений або нестандартний розподіл. Це може бути спричинено викидами в даних. QT може зменшити вплив викидів, тому це надійна схема попередньої обробки. Крім того, QT відобразить розподіл ймовірностей змінної в інший розподіл ймовірностей. Багато алгоритмів ML працюють краще, коли числові вхідні змінні мають стандартний розподіл ймовірностей, наприклад гаусівський (нормальний) або рівномірний розподіл. Числова вхідна змінна може бути автоматично перетворена за допомогою QT, щоб мати інший розподіл даних, який можна використовувати як вхідні дані для прогнозної моделі. Трансформація застосовується до кожної функції окремо.

Розподіл Гауса для дійсних змінних є явним припущенням, зробленим деякими алгоритмами, такими як лінійна регресія та LR. Незважаючи на відсутність цього припущення, інші нелінійні алгоритми часто працюють краще, коли змінні мають розподіл Гауса. Це справедливо для цільових змінних із дійсними значеннями в задачах регресії, а також для вхідних змінних із дійсними значеннями в задачах класифікації та регресії.

Проблема незбалансованої класифікації полягає в тому, що існує занадто мало спостережень класу меншості, щоб модель могла ефективно засвоїти межу прийняття рішення. Щоб впоратися з цим, я пропоную поєднати методи недостатньої та надмірної вибірки.

Для надмірної вибірки найбільш широко використовуваний підхід називається технікою надмірної вибірки синтетичної меншості (SMOTE). Як показано на рисунку 2.6, метод надмірної вибірки буде неодноразово вибирати випадкову точку меншості та сусідню точку меншості. Після цього він створить нову штучну точку на лінії між цими двома пінтами.

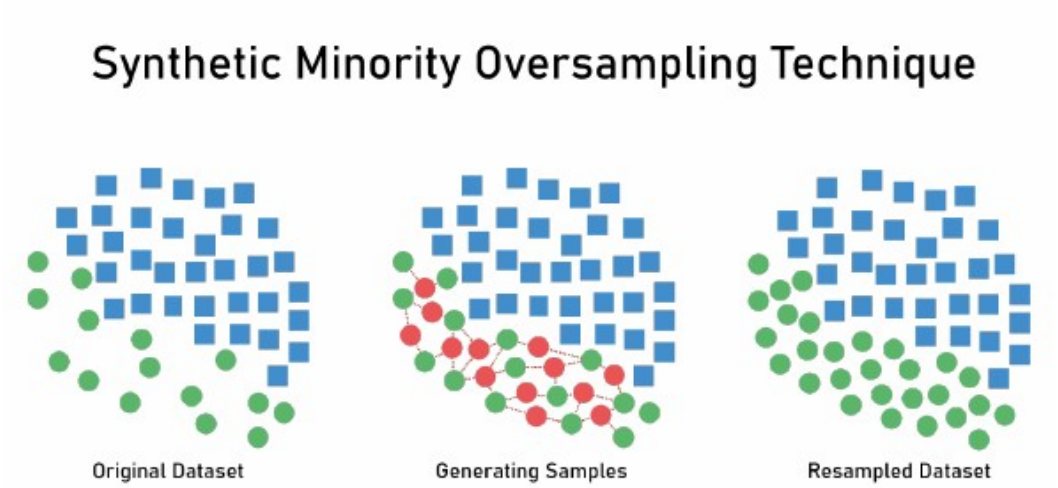


Рис. 2.6. Метод SMOTE

Однак загальним недоліком методу недостатньої вибірки є те, що синтетичні приклади створюються без урахування основного класу, що, можливо, призводить до неоднозначних прикладів, якщо існує сильне перекриття між класами.

Отже, після надмірної дискретизації нам потрібно видалити шумові дані за допомогою методів недостатньої дискретизації, що називається Tomek Links [22]. Tomek Links є однією з модифікацій техніки недостатньої вибірки Condensed Nearest Neighbors (CNN). На відміну від методу CNN, який лише випадковим чином вибирає вибірки з k найближчими сусідами з мажоритарного класу, який потрібно видалити, посилання Tomek видаляють усі мажоритарні вибірки, неправильно класифіковані k найближчими сусідами або які мають сусіда з іншого класу. Водночас це зніме їхній вплив на міноритарні вибірки.

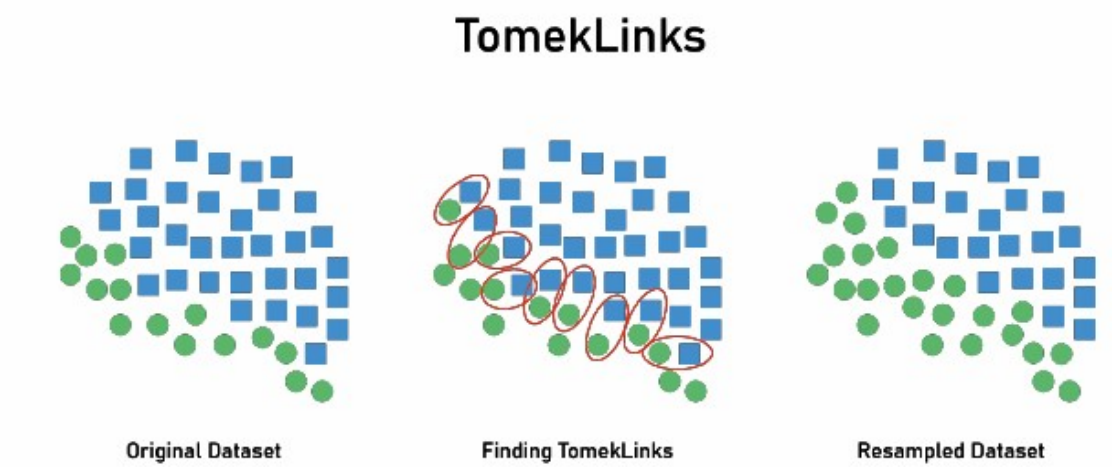


Рис. 2.7. Метод Tomek Links

SMOTE-Tomek Links [4] поєднує в собі здатність SMOTE генерувати синтетичні дані для класу меншості та здатність Tomek Links видаляти дані, які ідентифікуються як зв’язки Tomek, із основного класу.

Процес SMOTE-TomekLinks виглядає наступним чином:

1. Виберіть випадкові дані з класу меншості.

2. Обчисліть відстань між випадковими даними та їхніми k найближчими сусідами.

3. Помножте різницю на випадкове число від 0 до 1, а потім додайте результат до класу меншості як синтетичного зразка.

4. Повторюйте кроки 2-3, доки не буде досягнуто бажаної частки меншини.

5. Виберіть випадкові дані з більшості класів.

6. Якщо найближчим сусідом випадкових даних є дані з меншого класу (тобто створити Tomek Links), тоді видаліть Tomek Links.

На рисунку 2.8, техніку SMOTE-TomekLinks буде застосовано лише на навчальному наборі. Оскільки перевірка та тестовий набір є невидимими та невідомими реальними даними, які використовуються для перевірки наших моделей, на них не буде застосовано техніку SMOTE-TomekLinks. Вони міститимуть синтетичні дані та змінюватимуть реальний розподіл, якби я застосував надмірну та недостатню вибірку для перевірки та тестового набору.

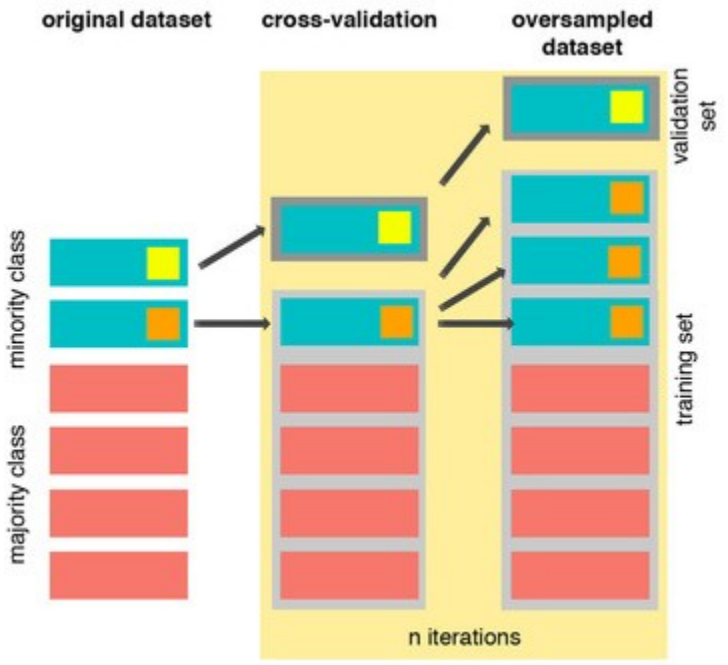


Рис. 2.8. Процес крос-валідації з використанням SMOTE-TomekLinks

Оскільки в нас є лінійну модель, необхідно розглянути, чи існує мультиколінеарність у незалежних змінних чи ні. При мультиколінеарності коефіцієнти регресії все ще є послідовними, але більше не є надійними, оскільки стандартні помилки завищені. Це означає, що прогностична

здатність моделі не зменшується , але коефіцієнти можуть бути статистично незначимими з помилкою типу II.

Наприклад, щоб проаналізувати зв’язок розмірів компанії та доходів за замовчуванням у регресійній моделі, ринкова капіталізація та доходи є незалежними змінними. Ринкова капіталізація компанії та її загальний дохід сильно корелюють.

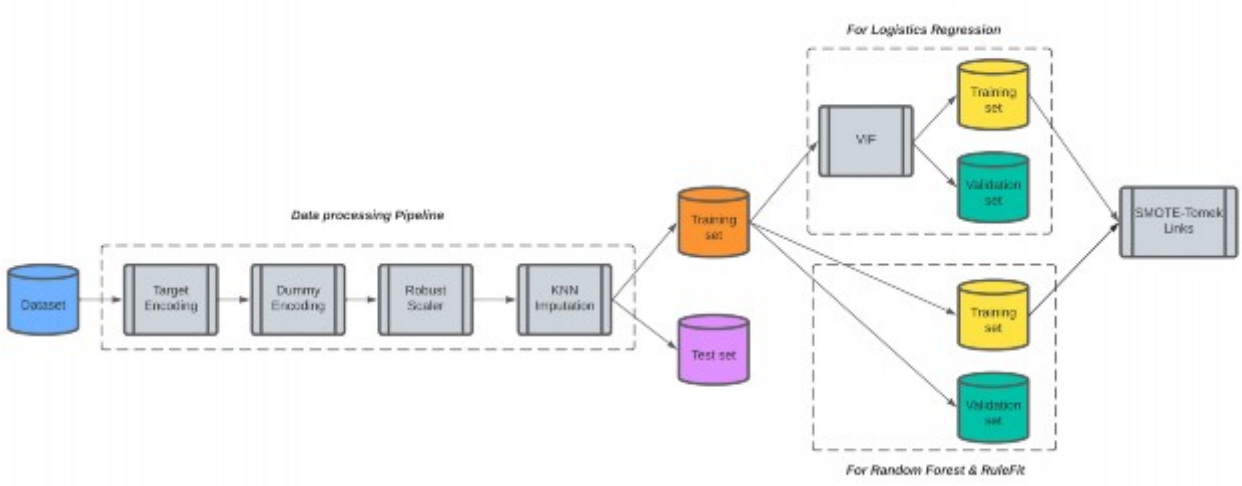


Рис. 2.9. Робочий процес попередньої обробки даних

2.6. Використання рекурсивного усунення ознак

Загалом, існує 3 різні типи: фільтри, обгортки та вбудовані методи. У даній роботі розглянуто рекурсивне виключення ознак (RFE).

RFE - це один із методів зворотного покрокового відбору ознак. RFE працює шляхом пошуку підмножини ознак, починаючи з усіх ознак у навчальному наборі даних і послідовно видаляючи ознаки, доки не залишиться бажана кількість. Це досягається шляхом навчання заданого алгоритму машинного навчання, який використовується в основі моделі, ранжування ознак за важливістю, відкидання найменш важливих ознак та повторного навчання моделі. Це повторюється доти, доки не залишиться задана кількість ознак. У роботі використовуються різні моделі для вибору відповідних важливих ознак. Алгоритм RFE такий:

Algorithm 3 Recursive Feature Elimination

1: Train the model on the training set with all explanatory variables
2: Compute the model performance
3: Compute the importance of each explanatory variable
4: **for** Each subset size S_i, $i = 1 \ldots s$ **do**
5: Select the S_i most important variables
6: Use the S_i most important variables to train the model
7: Compute the model performance
8: Compute the importance of each explanatory variable
9: **end for**
10: Compute the performance over the S_i
11: Determine the optimal number of explanatory variables

Для незбалансованого набору даних, на основі [6], автор стверджує, що вибір ознак перед SMOTE є кращим, оскільки він зазначив, що більшість методів вибору ознак припускають, що вибірки є незалежними. Таким чином, надмір і недостатність вибірки класу меншості за допомогою SMOTE порушує припущення незалежності.

Рисунок 2.10 показано робочий процес RFE, також буде застосовано процес перехресної перевірки, щоб уникнути переобладнання.

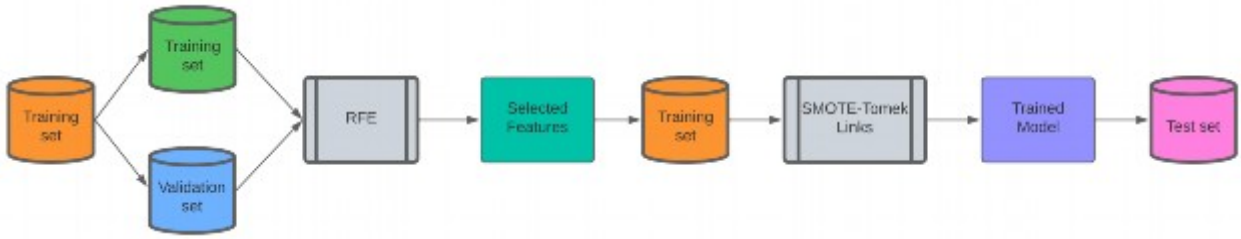


Рис. 2.10. Робочий процес рекурсивного усунення ознак

Експериментальні статистичні оцінки використовуються для перевірки статистичної значущості ознак, щоб отримати достовірний результат. Ми можемо визначити, чи залежна змінна y та незалежні змінні x у рівнянні логістичної регресії мають якісь суттєві зв'язки. Нульова гіпотеза полягає в тому, чи коефіцієнт в дорівнює нулю. Якщо будь-яка з нульових гіпотез дійсна, то x є статистично незначущим у моделі логістичної регресії. Тоді ми

не будемо відкидати нульову гіпотезу. Якщо статистичний тест нижчий за певний рівень значущості, наприклад 0,05, тоді нульову гіпотезу буде відхилено. Можна зробити висновок, що незалежні змінні x є значимими.

2.7. Застосування ієрархічного кластерного аналізу

Ієрархічний кластерний аналіз (HCA) – це алгоритм кластеризації, який групує подібні кластери об’єктів на основі певних критеріїв подібності. Він будує ієрархію кластерів, яка може бути представлена у вигляді дендрограми (деревоподібної діаграми).

Існують два типи алгоритмів ієрархічної кластеризації: агломеративна кластеризація, яка послідовно об’єднує подібні кластери, і роздільна кластеризація, яка послідовно розділяє несхожі кластери.

Причина, чому я представляю HCA, полягає в скороченні правил прийняття рішень у запропонованій моделі RuleFit. Існують тисячі правил прийняття рішень, згенерованих моделлю Random Forest , тому мені потрібно скоротити деякі правила, які мають подібний вплив. Ідея HCA базується на коефіцієнті рангового порядку Спірмена та вибирає одну функцію з кожного кластера на основі порогового значення відстані, яке є максимальною відстанню між кластерами. Значення порогу можна визначити, спостерігаючи за графіками дендрограми. Якщо поріг занадто малий, щоб дозволити будь-яким двом функціям утворити кластер, тоді кожна функція стане одним кластером. Якщо поріг занадто великий, щоб дозволити об’єднати всі функції, тоді буде лише один кластер.

Рисунки 2.11 і 2.12 візуалізують результат ієрархічної кластеризації, і я беру 20 функцій як приклад. Вісь ординат вказує відстань до кожного об’єкта. Клади, близькі до однакової висоти, схожі одна на одну, а клади з різною висотою — несхожі. Наприклад, є 7 пар ознак, які є досить близькими, як-от (stock_1 , stock_2) і (debitors_1 , debitors_2) у свою чергу близькі одна до одної.

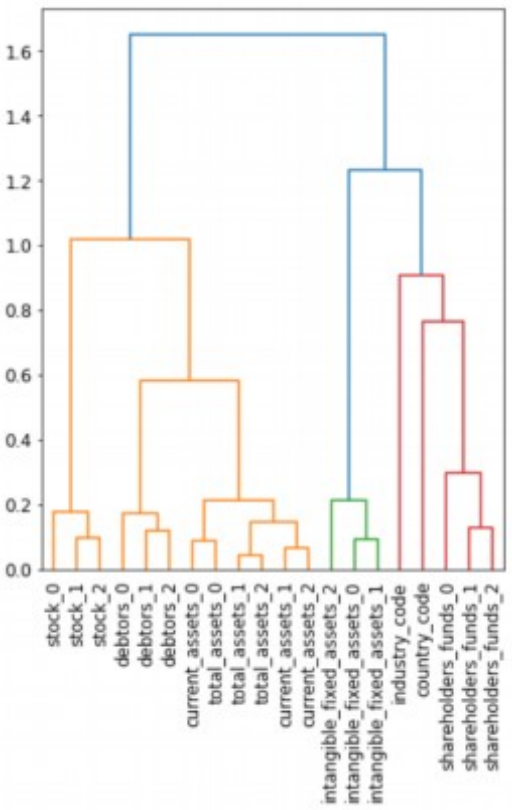


Рис. 2.11. Візуалізація ієрархічної кластеризації за допомогою дендограм

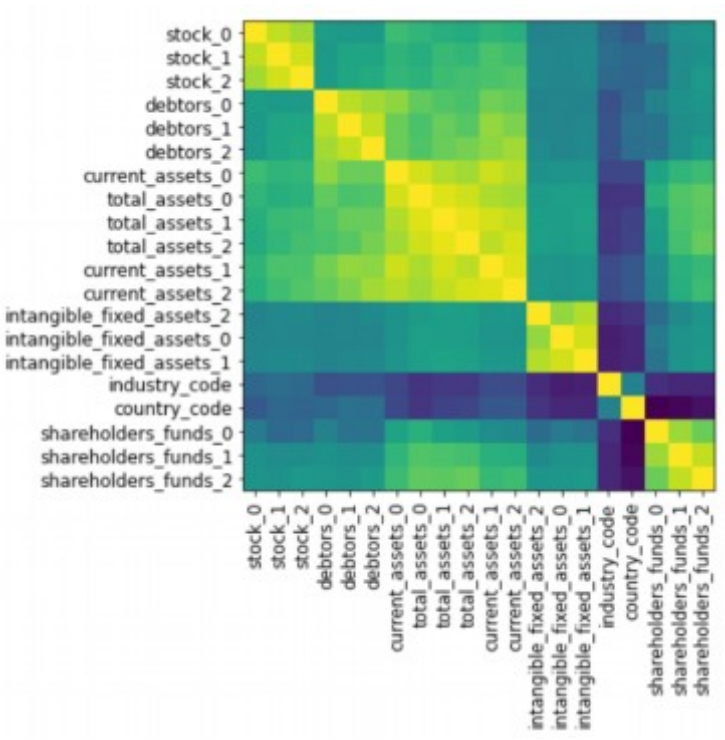


Рис. 2.12. Візуалізація ієрархічної кластеризації за допомогою коефіцієнта Спірмена

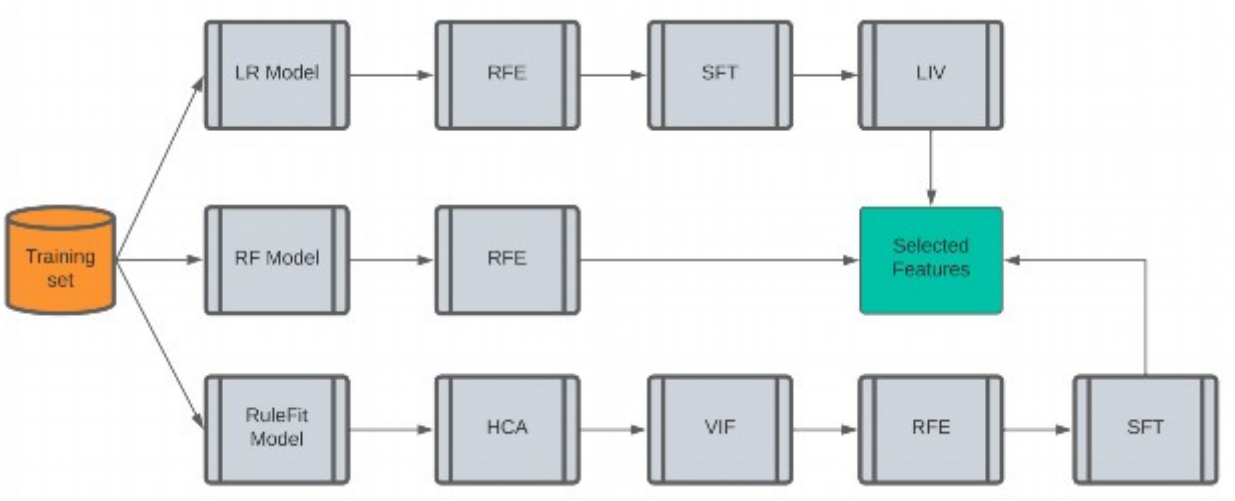


Рис. 2.13. Робочий процес вибору функції

Висновки до розділу

В даному розділі представлено інтелектуальні моделі та методологія прийняття рішень покращення ефективності оцінки кредитних ризиків. Пропонована методологія прийняття рішень включає використання моделі RuleFit, що забезпечує ефективне поєднання правил для інтерпретації прогнозів та точності оцінки. Це дозволяє надавати прозорі рекомендації для покращення кредитного скорингу.

Застосування різних метрик, таких як матриця невідповідності, повнота, середня геометрична оцінка, а також оцінки ROC AUC та F1, дозволяє забезпечити глибоку та всебічну оцінку ефективності моделей машинного навчання. Це дає можливість точніше оцінити ефективність рішень у задачах оцінки кредитних ризиків.

Аналіз та попередня обробка фінансових даних є важливими етапами, що сприяють підвищенню точності моделей, забезпечують ефективність візуалізації даних для подальшої інтерпретації результатів. Застосування рекурсивного усунення ознак сприяє оптимізації моделі, дозволяючи зменшити кількість ознак без втрати точності прогнозування, що знижує складність та підвищує ефективність моделей оцінки ризиків.

Ієрархічний кластерний аналіз допомагає визначити структурні залежності між даними, що дозволяє глибше зрозуміти поведінкові та структурні особливості клієнтів у задачах кредитного скорингу.

Ці аспекти формують основу для покращення процесу оцінки кредитних ризиків за допомогою інтелектуальних моделей та методів машинного навчання, підвищуючи їх точність та прозорість для прийняття обґрунтованих рішень.

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ІНТЕЛЕКТУАЛЬНИХ МОДЕЛЕЙ ДЛЯ ПОКРАЩЕННЯ ЕФЕКТИВНОСТІ ОЦІНКИ КРЕДИТНИХ РИЗИКІВ

3.1. Представлення набору даних для проведення імітаційного моделювання

Основна мета експерименту полягає в тому, щоб перевірити, чи може запропонована техніка ML – модель успішно працювати та вирішувати опубліковані дослідницькі запитання. Тому ми спочатку змоделюємо набір даних, щоб перевірити запропонований метод. Наступні пункти показують прогнозовані результати на моделі логістичної регресії (LR), випадкового лісу (RF) і пропонованої моделі RuleFit, а потім аналізуються результати моделювання.

3.1.1. Дані моделювання

У цьому розділі наведено короткий опис набору даних моделювання. Я використовую функцію sklearn make_blobs для створення випадкового двокласового фіктивного набору даних із 26 зразками та 2 функціями.

Таблиця 3.1.

Набір даних експерименту з 2 функціями та 1 міткою

Feature 1	Feature 2	label
9.963	4.596	1
11.032	-0.168	0
11.541	5.211	1
8.692	1.543	0
8.106	4.286	0
8.309	4.806	1
11.930	4.648	1
9.672	-0.202	0
8.348	5.134	1
8.674	4.475	1
9.177	5.092	1
10.240	2.455	1
8.689	1.487	0

Feature 1	Feature 2	label
8.922	-0.639	0
9.491	4.332	1
9.256	5.132	1
7.998	4.852	1
8.183	1.295	0
8.733	2.491	0
9.322	5.098	1
10.063	0.990	0
9.500	-0.264	0
8.344	1.638	0
9.501	1.938	0
9.150	5.498	1
11.563	1.338	0

3.1.2. Логістична регресія

На рисунку 3.1 показано, як LR знаходить межу, яка розділяє зразки кожного класу. Як бачимо, неправильно класифіковано 2 пункти. Тому для моделі LR не існує такої ідеальної межі, яка могла б належним чином розділити зразки кожного класу.

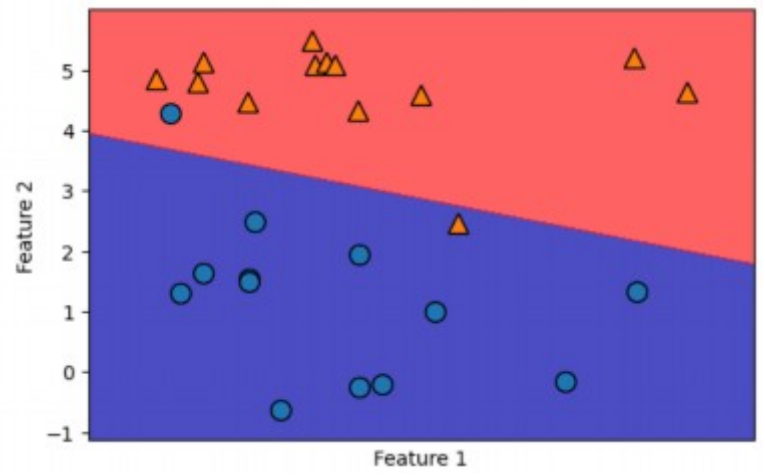


Рис. 3.1. Гіперплощина, згенерована моделлю логістичної регресії

3.1.3. Випадковий ліс

На рисунку 3.2 показано, як RF знаходить межу, яка розділяє зразки кожного класу. Як бачимо, межа більше не є лінійною. Усі зразки можна правильно класифікувати. В експерименті основна мета — перевірити, чи може LR правильно класифікувати неправильно класифіковані зразки, запровадивши правила прийняття рішень.

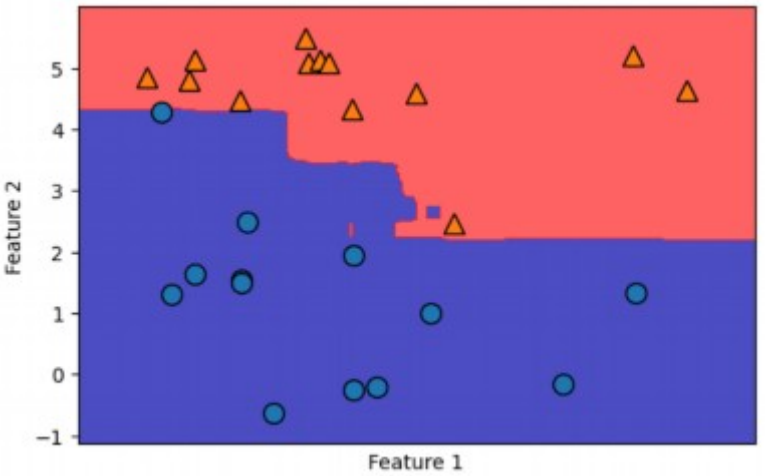


Рис. 3.2. Гіперплощина, згенерована моделлю Random Forest

3.1.4. Пропонована модель

Рисунок 3.3 показує, як LR + RF RuleFit знаходить межу, яка розділяє зразки кожного класу. Як ми бачимо, межа не є лінійною, і цього разу всі зразки можна класифікувати правильно.

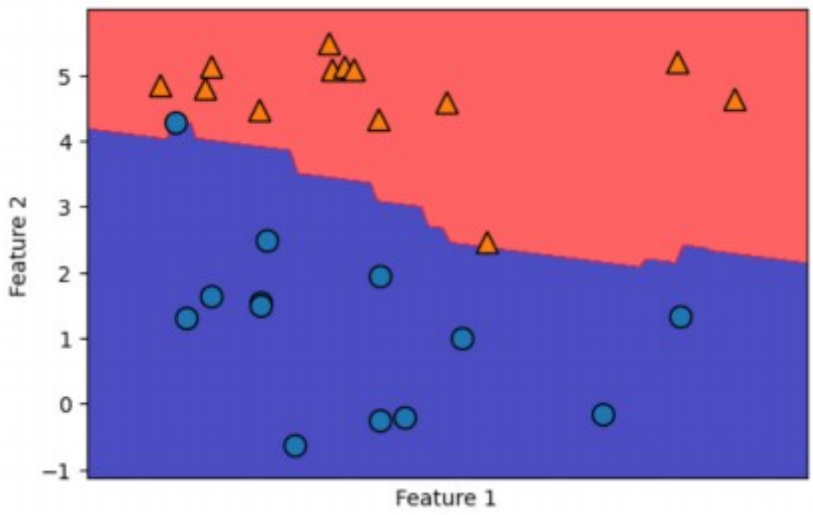


Рис. 3.3. Гіперплощина, згенерована пропонованою моделлю

Щоб змусити LR генерувати таку межу, видобуваються правила прийняття рішення з RF і обрізаються правила на основі важливості, остаточні вибрані правила наведені нижче в таблиці 3.2.

Таблиця 3.2.

Вибрані правила прийняття рішень з RF

Features
feature_1 $\leq$ 10.0137 & feature_2 $>$ 4.309
feature_1 $\leq$ 9.489 & feature_2 $\leq$ 4.569 & feature_2 $>$ 2.046
feature_1 $\leq$ 11.287 & feature_1 $>$ 8.682
feature_1 $>$ 9.956 & feature_2 $\leq$ 4.441 & feature_2 $>$ 1.143
feature_1 $\leq$ 8.942 & feature_1 $>$ 8.682
feature_1 $\leq$ 11.552 & feature_1 $>$ 9.818
feature_1 $\leq$ 8.052
feature_1 $>$ 8.942
feature_1 $>$ 11.747
feature_1 $\leq$ 8.208& feature_1 $>$ 8.052 & feature_2 $>$ 2.196

3.1.5. Експериментальні результати та аналіз

Згідно з моїм симуляційним експериментом, я можу зробити висновок, що мої припущення або запропоновані методи можуть працювати належним чином. Після застосування додаткових правил прийняття рішень до моделі логістичної регресії (LR), неправильно класифіковані зразки класифікуються з правильною міткою, що дійсно покращує продуктивність моделі LR. Більше того, після введення правил прийняття рішень межа виглядає досить схожою на згенеровану випадковим лісом (RF).

Тепер, на основі результатів симуляційного експерименту, я можу відповісти на наступні питання:

- Щодо загальної продуктивності, чи є запропонована модель з додатковими функціями правил кращою за класичну модель LR з оригінальними функціями на тестових даних, зберігаючи при цьому пояснюваність?

- Яка характеристика виняткових підгруп для тестових даних?

- Чи може пропонована модель з додатковими функціями правил правильно класифікувати виняткову підгрупу? Наскільки покращилася продуктивність на винятковій підгрупі?

Щодо загальної продуктивності, чи є запропонована модель з додатковими функціями правил кращою за класичну модель LR з оригінальними функціями на тестових даних, зберігаючи при цьому пояснюваність ?

Метод машинного навчання, відомий як модель RuleFit, поєднує пояснювальну здатність LR з прогнозною здатністю RF. Згідно з таблицею 3.2, ми можемо бачити, що функція містить лінійні функції та функції правил. Кожна функція має власне значення коефіцієнта, яке є таким самим, як у моделі LR. Таким чином, модель RuleFit зберігає пояснюваність і може розглядатися як інтерпретована модель. Крім того, на основі результатів експерименту RuleFit, неправильно класифіковані зразки були правильно класифіковані після введення правил прийняття рішень. Це доводить, що

модель RuleFit дійсно покращила продуктивність LR. Таким чином, запропонований метод може вирішити перше дослідницьке питання.

Яка характеристика виняткових підгруп для тестових даних?

У моєму експериментальному наборі даних я створив лише 2 виняткові зразки, які розташовані на неправильній стороні. Як показано на рисунку 3.1, модель LR не може правильно класифікувати ці 2 зразки, що означає, що не існує лінійної межі, яка може правильно розділити всі зразки.

Однак пропонована модель може сформувати нелінійну межу, як показано на рисунку 3.3, завдяки правилам прийняття рішень, згенерованим з моделі RF. Функція правила може бути використана для опису цих 2 виняткових зразків, оскільки кожне правило є бінарним атрибутом. Якщо зразок відповідає всім умовам в одному правилі, то значення цієї функції правила буде 1, інакше 0. Отже, пропонована модель також може вирішити друге дослідницьке питання за допомогою своїх функцій правил.

Чи може пропонована модель з додатковими функціями правил правильно класифікувати виняткову підгрупу? Наскільки покращилася продуктивність на винятковій підгрупі?

Рисунок 3.3 вже візуалізує, що пропонована модель може правильно класифікувати виняткові дані. Однак у реальних даних важко візуалізувати дані високої розмірності. Тому я буду використовувати метрики оцінки, такі як AUC ROC або Accuracy, щоб перевірити третє дослідницьке питання. Нарешті, оскільки всі дослідницькі питання можна вирішити під час симуляційного експерименту, то можна застосувати модель RuleFit до реальних випадків.

Під час експерименту я виявив, що чим більше правил прийняття рішень я включаю в модель LR, тим більш гладкою стає межа.

Рисунок 3.4 показує межу, згенеровану з 714 правилами прийняття рішень. Проблема полягає в тому, що деякі правила прийняття рішень є надлишковими, оскільки вони мають високу кореляцію один з одним і,

можливо, не є важливими або значущими. Тому потрібно скоротити правила та знайти остаточні результати, які показані в таблиці 3.2.

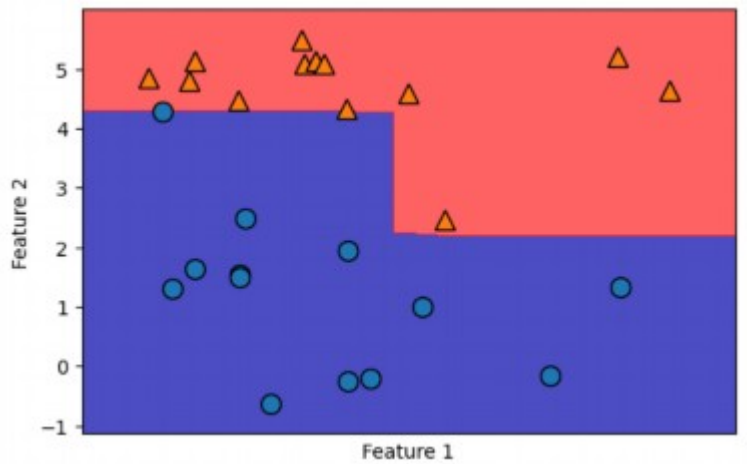


Рис. 3.4. Гіперплощина, згенерована пропонованою моделлю із 714 правилами прийняття рішень

3.2. Процес машинного навчання та тестування

У цьому розділі буде представлено процес створення та оцінки моделі. Для навчання та тестування моделей я використовував пакет Python ML scikit-learn. Дані, які я використав, є реальними даними Zanders. Як зазначено в попередньому розділі, я лише випадковим чином відбираю 30% даних із необробленого набору даних. Після обробки даних остаточно оброблені дані містять 406 458 вибірок із 109 залежними змінними та 1 незалежною змінною. Ще одна причина, чому я вибираю 30% даних, пов’язана з обмеженою оперативною пам’яттю. Співвідношення 80:20 для розподілу всіх попередньо оброблених даних, а методи вибірки використовуються для випадкового розподілу даних на 80% для навчання (включаючи 20 % даних перевірки) і 20 % для тестування моделей.

Таблиця 3.3.

Розмір необроблених даних навчального та тестового набору

	data size	non-default	default
Training set	406,458	404,286	2172
Test set	101,615	101,072	543

Таблиця 3.4.

Розмір даних навчального набору після SMOTE-TomekLinks

	data size	non-default	default
Training set	808,572	404,286	404,286

3.2.1. Модель Random Forest

У цій роботі RF є еталонною моделлю, і я використовую функцію RandomForestClassifier з бібліотеки scikit-learn для створення моделі. Через проблему мультиколінеарності навчальні та тестові набори відрізняються від наборів для моделі LR. RF використовує початкову вибірку та методи вибірки функцій, такі як вибірка рядків і стовпців. Як наслідок, мультиколінеарність не сильно впливає на RF, оскільки вона вибирає різні набори функцій для кількох моделей, і кожна модель бачить інший набір точок даних. Таким чином, 109 функцій тренуються на РЧ без застосування техніки VIF.

Перед навчанням я роблю пошук у сітці, щоб знайти оптимальні гіперпараметри. Пошук у сітці застосовується з вкладеною перехресною перевіркою. Навіщо мені потрібна перехресна перевірка? Уявіть, що я випадково вибираю простий набір тестів, що призводить до надто оптимістичних чи песимістичних оцінок.

Щоб анулювати сценарій, найпростішим способом є усереднення результатів після кількох ітерацій однієї процедури, яка називається перехресною перевіркою.

Перехресна перевірка — це техніка повторної вибірки, яка використовується для навчання та тестування моделі на різних ітераціях з використанням різних підмножин даних. Зазвичай його застосовують у ситуаціях, коли метою є передбачення, і потрібно оцінити, наскільки добре прогнозна модель функціонуватиме в реальних ситуаціях. Тому перехресна перевірка може забезпечити надійний спосіб оцінити продуктивність нашої моделі. Рисунок 3.5 показує, як працює перехресна перевірка.

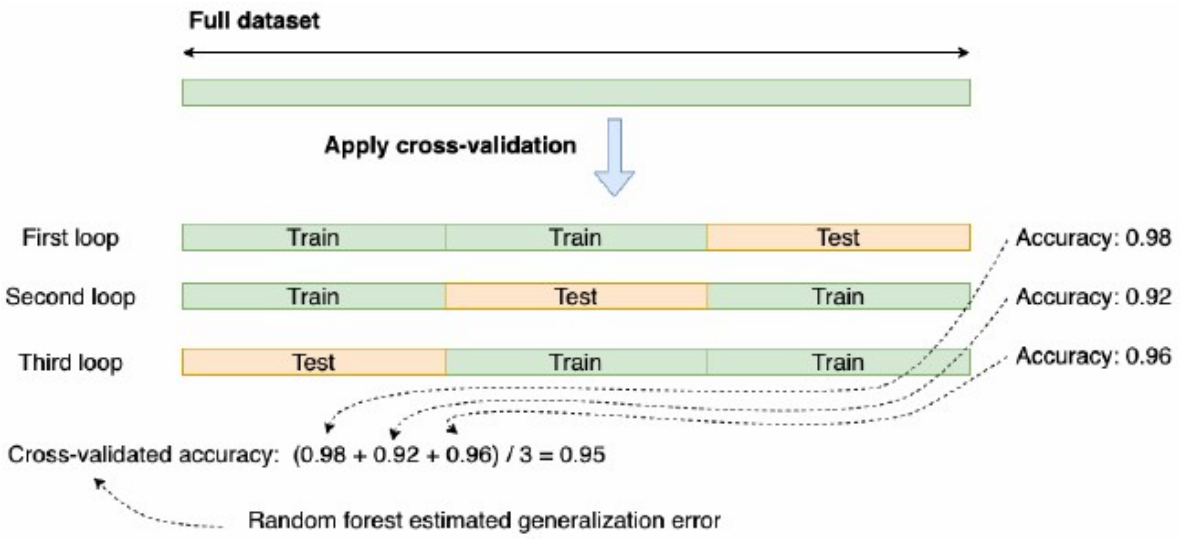


Рис. 3.5. Ілюстрація перехресної перевірки [7]

Крім того, чому мені потрібна вкладена перехресна перевірка замість перехресної перевірки ? У [14] автори показали, що коли ми застосовуємо метод перехресної перевірки для оптимізації гіперпараметрів, можна отримати надто оптимістичну оцінку, спричинену переобладнанням. Тому я представляю метод вкладеної перехресної перевірки, який забезпечує дуже точний метод оптимізації гіперпараметрів. На рисунку 3.6 показано, як працює вкладена перехресна перевірка. Є 2 перехресні ¬перевірки, одна називається зовнішньою перехресною перевіркою, а інша називається внутрішньою. Внутрішня перехресна перевірка допоможе нам знайти оптимальні гіперпараметри, які мають найкращий середній показник точності. Зовнішня перехресна перевірка використовуватиме найкращу модель з оптимальними гіперпараметрами для оцінки моделі.

Внутрішня перехресна перевірка використовує функцію GridSearchCV. Параметри такі:

- n_jobs = -1
- scoring = 'roc_auc'
- cv = 5

Зовнішня перехресна перевірка використовує функцію cross_val_score . Параметри такі:

- n_jobs = -1
- scoring = 'roc_auc'
- cv = 5

Для моделі RF фіксованим параметром є:

- n_jobs = -1
- random_state = 0
- oob_score = True

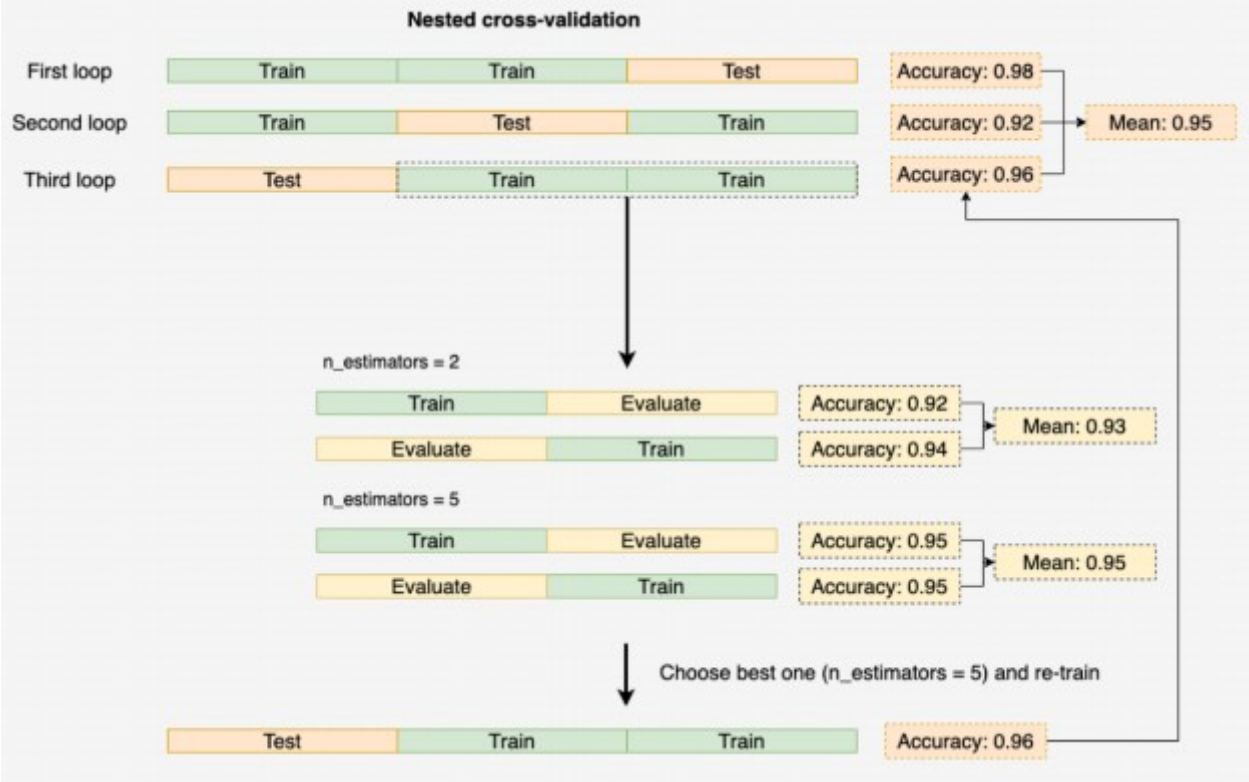


Рис. 3.6. Ілюстрація вкладеної перехресної перевірки

Таблиця 3.5.

Простір пошуку кожного гіперпараметра

hyperparameter	grid search space
n_estimators	64, 128, 256, 512
max_depth	range(5, 11, 1)

Решта параметрів є значеннями за замовчуванням. Час роботи пошуку гіперпараметрів становить близько 56 годин. Таблиця 3.6 показує результати пошуку по сітці з вкладеною перехресною перевіркою. Вкладена перехресна перевірка ROC AUC – 0,900, а ROC AUC – 0,832.

Таблиця 3.6.

Результат пошуку в сітці моделі RF

model	n_estimators	max_depth	ROC AUC
Random Forest Classifier	128	9	0.900

3.2.2. Рекурсивне усунення ознак

Як обговорювалося в попередньому розділі, Recursive Feature Elimination (RFE) може допомогти нам вибрати важливі функції. Я використав функцію RFECV із scikit-learn, вона застосувала RFE із перехресною перевіркою для вибору кількості функцій. Параметри такі:

- n_jobs = -1
- cv = 5
- scoring = 'roc_auc'

Після RFE вибрано 24 важливі функції.

Важливість ознак – це метод, який оцінює вхідні функції відповідно до того, наскільки добре вони здатні передбачити дану цільову змінну. Рисунок 3.7 відображає важливість 20 основних функцій.

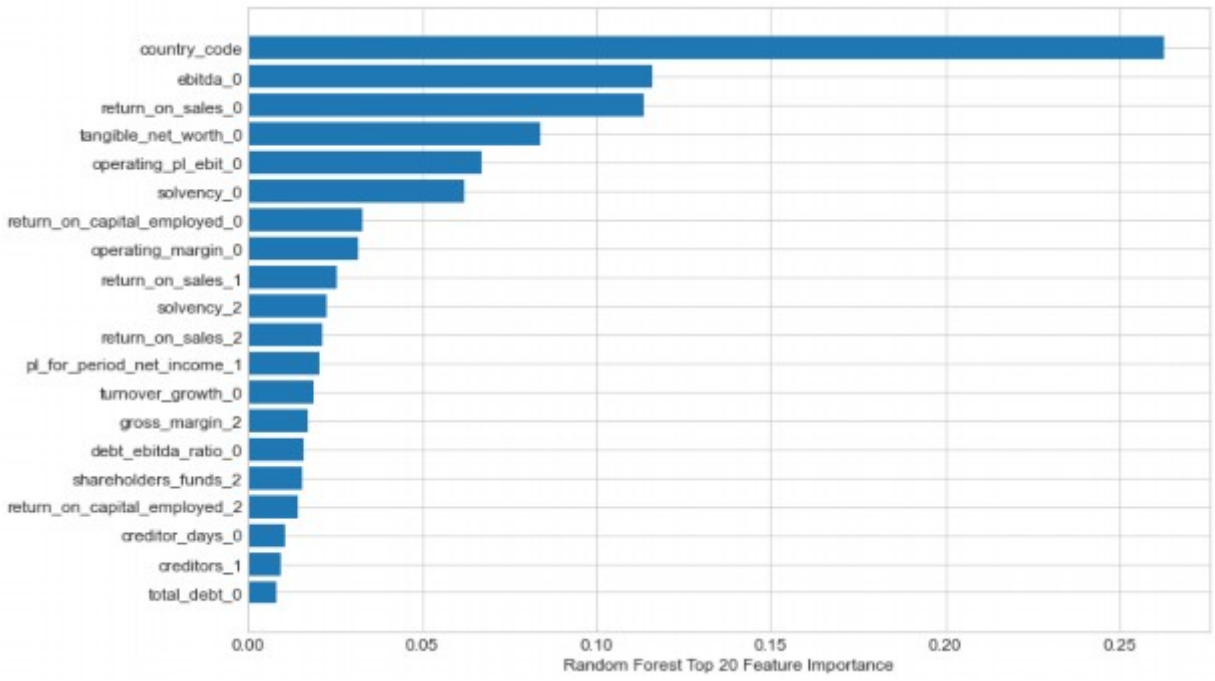


Рис. 3.7. Важливість 20 найпопулярніших функцій у RF-моделі на навчальному наборі

На рисунку 3.7 показано, що найважливішою функцією є country_code . Наступні дві важливі функції — це ebitda_0 і return_on_sales_0 . Однак спостерігається, що решта функцій не настільки важливі порівняно з основними функціями. Тому я провів тест, щоб показати, як кожна функція впливає на прогноз. На рисунку 3.8 показана оцінка ROC AUC для навчання та перевірки суттєво не збільшується після додавання 13 основних важливих функцій, що означає, що додавання додаткових функцій суттєво не покращить продуктивність. Таким чином, я можу просто вибрати 13 найкращих функцій як остаточну RF-модель.

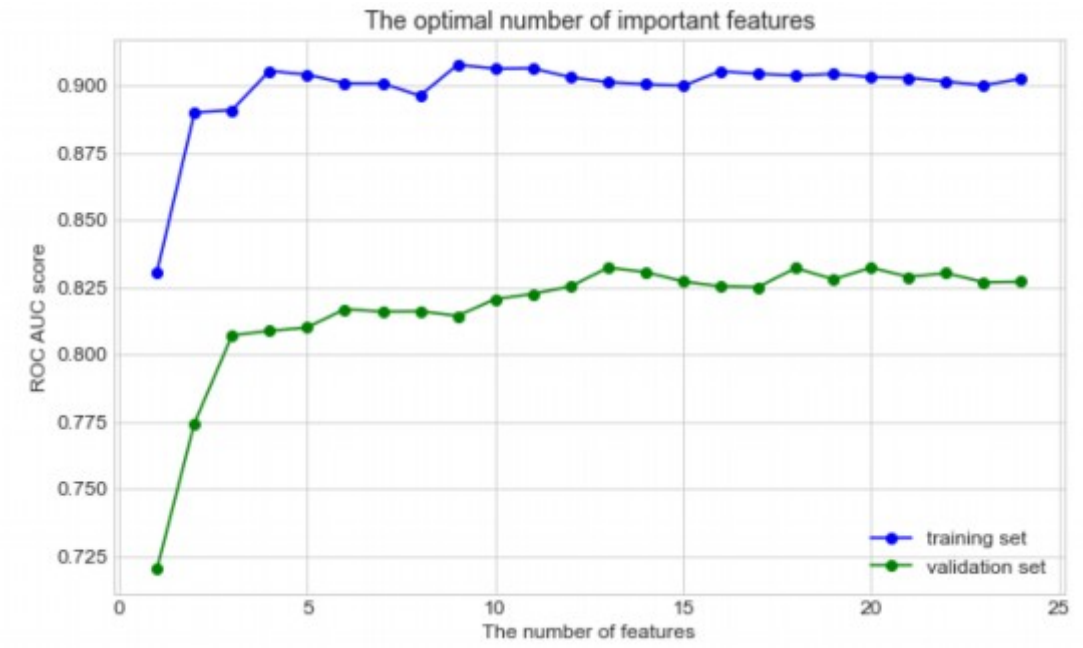


Рис. 3.8. Вибір оптимальної кількості функцій у RF моделі

Однак, виходячи з другого розділу, де вже обговорювалось, що є 2 недоліки важливості функції:

1) результати зміщені в бік функцій з високою кардинальністю.

2) результати обчислюються на навчальному наборі і тому не відображають здатність функції бути корисною для створення прогнозів, які узагальнюють тестовий набір.

Таким чином, я вводжу метод важливості перестановки, щоб перевірити важливість функції в тестовому наборі.

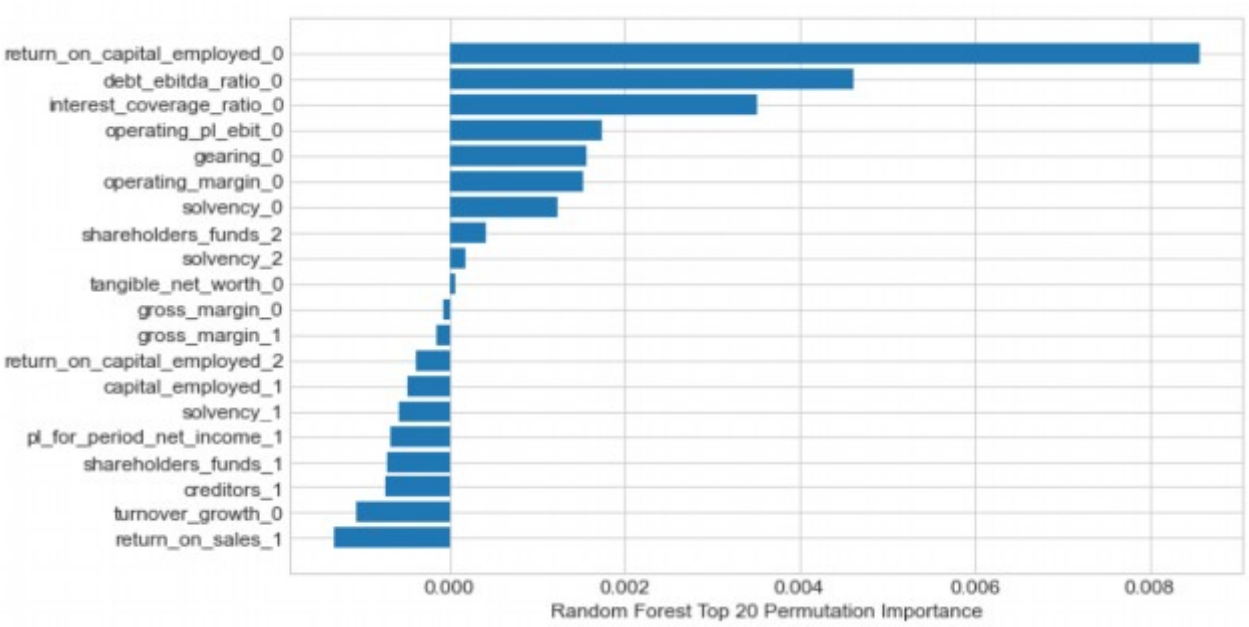


Рис. 3.9. Важливість перестановки 20 основних функцій у RF-моделі на основі даних тестування

З рис 3.9 , ми бачимо, що рейтинг зовсім інший. Деякі функції мають негативне значення, яке вказує на те, що прогнози на перетасованих (або зашумлених) даних точніші, ніж реальні дані. Це означає, що ця функція не робить значного внеску в прогнози (важливість близька до 0), але випадкова випадковість спричинила більшу точність прогнозів на перетасованих даних.

3.2.3. Оцінка проведеного навчання

Відповідно до розділу 2, метрики Accuracy, Precision, Recall, ROC AUC, F1 і Brier використовуються для оцінки моделі.

На рисунку 3.10 показано матрицю невідповідності. Оскільки тестовий набір із реального світу, ми бачимо, що він надзвичайно незбалансований. Співвідношення за умовчанням:не за замовчуванням становить приблизно 1:186. Однак RF-модель може правильно класифікувати 90 431 даних, які не є стандартними, і 418 даних за умовчанням. Таблиця 3.7 показує показники ефективності за різними показниками оцінювання. Радіочастотна модель як еталонна модель насправді має чудову загальну здатність прогнозувати, згідно з оцінкою продуктивності.

Таблиця 3.7.

Результати оцінки RF-моделі на тестовому наборі (кожна метрика зважена за кількістю істинних випадків для кожного класу)

	RF model (benchmark)
Selected Features	13
Gmeans	0.830
ROC AUC	0.832
Precision	0.993
Recall	0.894
F1 score	0.939
Brier score	0.081

Рис. 3.10. Матриця невідповідностей для RF-моделі з 13 вибраними функціями

3.3. Застосування методології RuleFit для оцінки кредитних ризиків

Методологія RuleFit була згадана в попередньому розділі. У цьому розділі буде представлено, як навчити модель RuleFit окремо з DT і RF і оцінити їх ефективність. Рисунок 3.11 показує робочий процес навчання моделі RuleFit. Як видно, спочатку окремо навчаються класифікатор RF і DT. Вирішальні правила взяті з RF і DT. У DT всього одне дерево, а в RF 128

дерев. У цьому випадку RF генеруватиме більше правил, ніж DT. Далі правила прийняття рішень скорочуються методом ієрархічного кластерного аналізу (HCA). Після цього скорочені правила та вихідні лінійні об’єкти об’єднуються, щоб сформувати новий набір даних. Оскільки ці нові ф'ючерси будуть вписуватися в модель LR, мультиколінеарність необхідно перевірити перед навчанням. Наступний крок – вибір функції. RFE буде використано знову для визначення важливих ознак, а після перевірки статистичної значущості несуттєві характеристики будуть виключені. Нарешті, вибрані суттєві характеристики будуть підігнані до моделі LR, яка використовувала ті самі гіперпараметри.

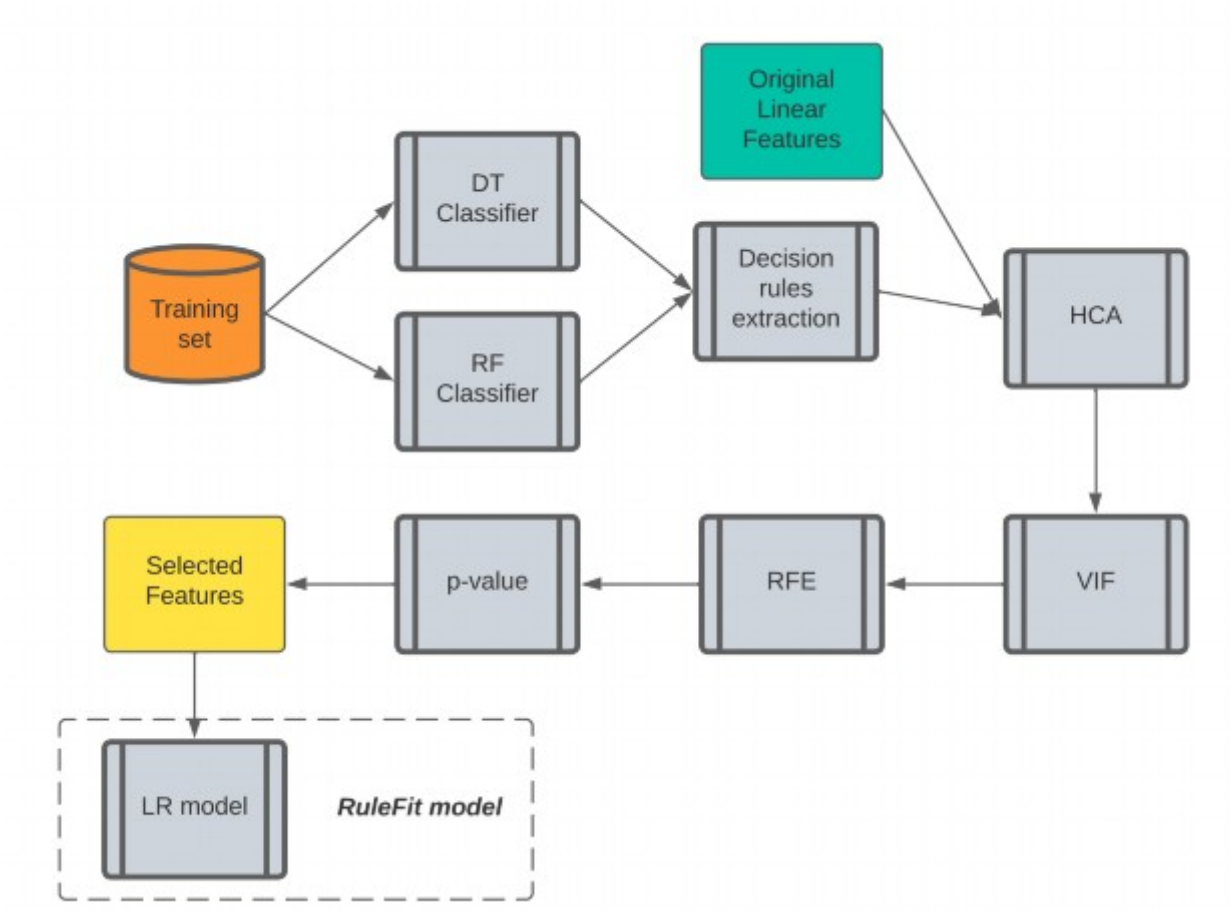


Рис. 3.11. Блок-схема для навчання моделі RuleFit

Як я вже згадував раніше, RF класифікатор містить 128 дерев. Після навчання він створить тисячі правил. Тому ми повинні відфільтрувати зайві та подібні правила, на відміну від класифікатора DT. HCA, який також

вважається методом скорочення правил, може допомогти нам позбутися зайвих правил. Отже, для радіочастотного класифікатора є 1813 правил, згенерованих до HCA. Після застосування техніки HCA всього лише 112 правил. Після об'єднання перших 7 ознак ми маємо 119 змішаних функцій. Для класифікатора DT створено лише 6 правил і поєднано з початковими 7 функціями, у нас є 13 змішаних функцій.

Наступним кроком є перевірка мультиколінеарності. Підхід VIF застосовано знову, і коли він використовується, залишається 50 функцій для моделі LR + RF Rulefit.

Процес RFE однаковий у RF та LR частині. Параметри такі:

- n_jobs = -1
- cv = 5
- scoring = 'roc_auc'

Після застосування RFE було вибрано 38 важливих функцій.

Тест статистичної значущості той самий що в попередньому розділі. Після того, як я крок за кроком виключаю несуттєві ознаки (p-value>0,05), для моделі LR + RF RuleFit є 30 значущих об'єднаних ознак, а для моделі LR + DT RuleFit – 10 значущих змішаних характеристик.

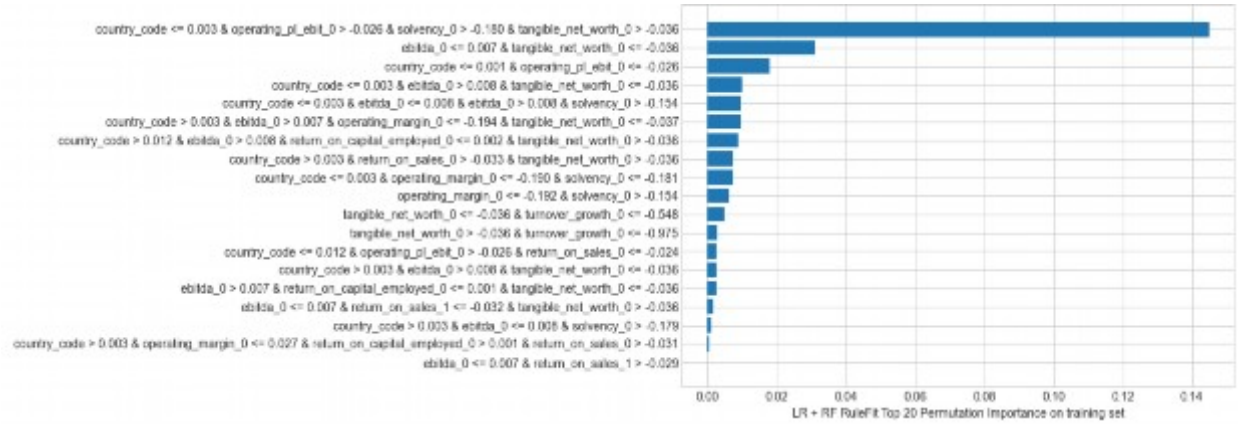


Рис. 3.12. Важливість перестановки моделі LR + RF RuleFit для тестових даних

Рисунок 3.12 показує важливість перестановки 20 основних функцій у навчальному наборі з моделлю LR + RF RuleFit. Як видно, порівняно з

іншими характеристиками, ' country_code < 0,003 & operating_pl_ebit_0 > -0,026 & solvency_0 > -0,180 & tangible_net_worth_0 > -0,036' є найважливішою характеристикою.

Щоб вибрати оптимальну кількість функцій, я провів ще один тест, щоб показати вплив продуктивності, додавши решту функцій одну за одною. Результат можна побачити на рисунку 3.13. Було відмічено, що оцінка ROC AUC навчання та перевірки суттєво не змінюється після 19 функцій, що означає, що додавання додаткових функцій не суттєво допоможе моделі прогнозувати.

Таким чином, я можу просто вибрати 19 найкращих функцій як остаточну модель LR + RF RuleFit.

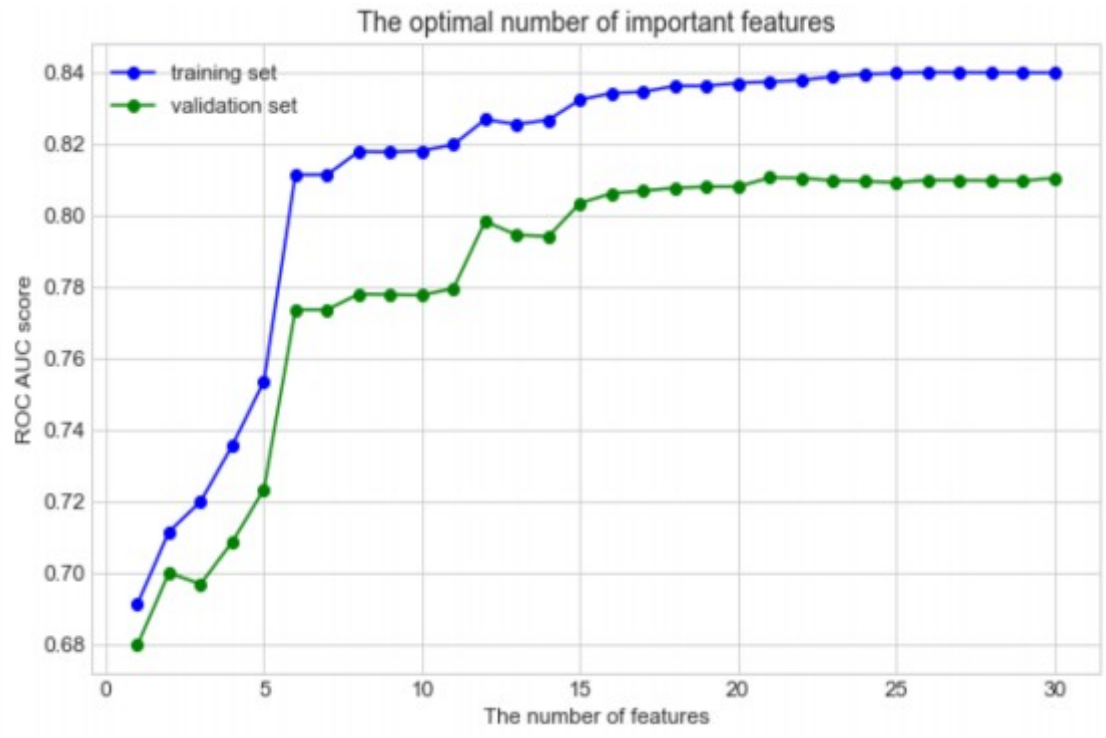


Рис. 3.13. Вибір оптимальної кількості функцій у моделі LR + RF RuleFit

Рисунок 3.14 показує важливість перестановки 20 основних функцій у навчальному наборі з моделлю LR + DT RuleFit. Як видно, порівняно з іншими функціями ' country_code < -0.375 & return_on_sales_0 > -0.373' є найважливішою функцією.

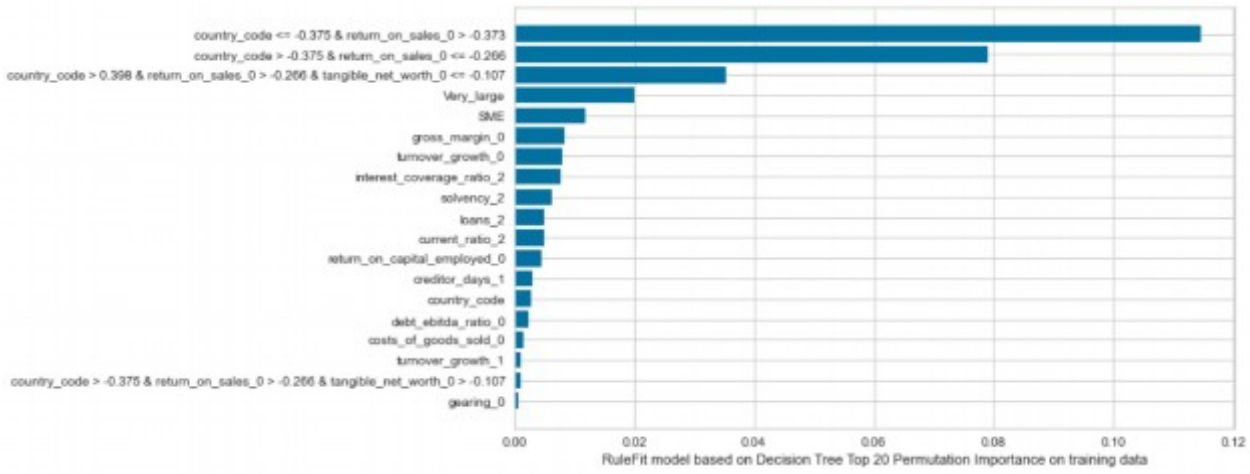
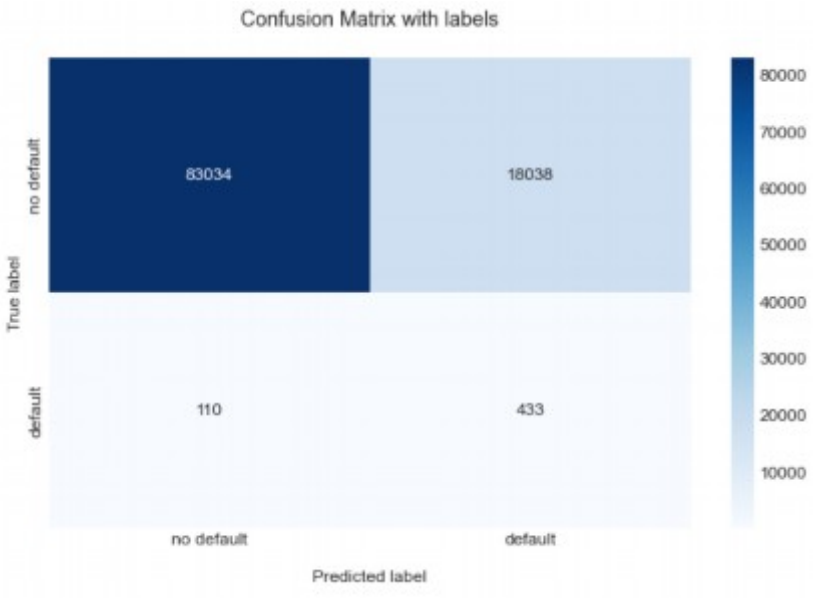


Рис. 3.14. Важливість перестановки моделі LR + DT RuleFit на тестових даних

Подібно до RF і LR, показники Gmeans, Precision, Recall, ROC AUC, F1 і Brier використовуються для оцінки 2 моделей RuleFit. На рисунку 3.15 показана матриця невідповідності моделі LR + RF і моделі LR + DT на тестовому наборі. Як можна помітити, кількість правильно передбачених зразків за замовчуванням для моделей LR + RF і LR +DT є відносно однаковою, однак кількість правильно передбачених зразків за замовчуванням для моделі LR + RF вища, ніж для моделі LR + DT . Таблиця 3.8 показує показники ефективності за різними показниками оцінювання.



а) Матриця невідповідності моделі LR + RF

б) Матриця невідповідності моделі LR + DT

Рис. 3.15. Матриці невідповідностей двох моделей RuleFit у наборі для тестування

Таблиця 3.8.

Результати оцінки всіх моделей на тестовому наборі (кожна метрика зважена за кількістю істинних випадків для кожного класу)

	RF model (benchmark)	LR model (base)	DT model (base)	LR + RF (proposed)	LR + DT (base)
Selected Features	13	7	7	19	9
Gmeans	0.830	0.706	0.750	0.813	0.732
AUC ROC	0.832	0.707	0.755	0.809	0.734
Precision	0.993	0.992	0.992	0.993	0.992
Recall	0.894	0.747	0.839	0.821	0.832
F1 score	0.939	0.850	0.907	0.897	0.903
Brier score	0.081	0.183	0.150	0.121	0.130

В таблиці 3.8 наведено результати оцінки для всіх моделей. Запропонована модель LR + RF RuleFit сильніша за модель LR + DT RuleFit і краща за базову модель LR і DT. Крім того, продуктивність наближена до еталонної моделі RF.

Коефіцієнти моделі RuleFit можна пояснити подібно до коефіцієнтів LR. Візьмемо, наприклад, LR + RF, наше запитання полягає в тому, як інтерпретувати функції правила? Наприклад, 'country_code < 0,003 &

operating_pl_ebit_0 > -0,026 & solvency_0 > -0,180 & tangible_net_worth_0 > -0,036'. Коефіцієнт першого правила становить -2,893, а відношення шансів (ймовірність дефолту: ймовірність недефолту) дорівнює 0,055. Тлумачення таке: якщо всі умови правила прийняття рішень задовольняються, то ймовірність дефолту проти недефолту є на 0,055 нижчою для компаній, які відповідають правилу, порівняно з компаніями, які не відповідають, припускаючи, що всі інші змінні залишаються постійними. Тому я можу пояснити всі функції правила, оскільки всі вони є двійковими категоріальними змінними.

Висновки до розділу

Отже, в цьому розділі представлено представлено набір даних, необхідний для проведення імітаційного моделювання кредитних ризиків, що є важливим для належної оцінки моделей та їхньої подальшої оптимізації. Це включає основні характеристики даних, які використовуються для навчання моделей, і формування вибірки для тестування. Логістична регресія та модель випадкового лісу були випробувані як базові методи для оцінки кредитних ризиків. Порівняння їхніх результатів надало можливість визначити сильні та слабкі сторони кожного підходу, що дало базу для подальшого вдосконалення пропонованої моделі.

Було розроблено і впроваджено нову модель, метою якої є підвищення точності прогнозування кредитних ризиків. Результати експериментів продемонстрували її переваги над стандартними методами за точністю та стабільністю оцінок. Аналіз результатів підтвердив ефективність запропонованого підходу.

Здійснення процесу машинного навчання та тестування на різних моделях, таких як Random Forest, дозволило оптимізувати процес прийняття рішень і покращити точність прогнозування.

Метод рекурсивного усунення ознак був використаний для зменшення кількості ознак, що зберігають ключову інформацію для прогнозування, і дозволяє покращити продуктивність моделі, зменшивши її складність і обчислювальні ресурси, необхідні для роботи. Використання RuleFit забезпечило додаткову прозорість у процесі прийняття рішень, що дозволяє зрозуміти основні фактори ризику та їхній вплив на фінальну оцінку. Це робить модель більш інтерпретованою, що є важливим для довіри до результатів у фінансових установах.

Узагальнюючи, реалізація інтелектуальних моделей у поєднанні з експериментальним аналізом підтверджує доцільність використання машинного навчання для оцінки кредитних ризиків. Запропоновані методи підвищують точність, інтерпретованість і стабільність прогнозів, що сприяє надійнішому управлінню кредитними ризиками.

ВИСНОВКИ

У магістерській роботі проведено дослідження інтелектуальних моделей та методів покращення ефективності оцінки кредитних ризиків. Запропоновано модель RuleFit, яка перетворює модель Random Forest (RF) з «чорної скриньки» на інтерпретовану модель «білої скриньки» шляхом вилучення правил прийняття рішень та подання їх у формі, зрозумілій для користувачів. Модель RuleFit використовує ієрархічний кластерний аналіз (HCA) для обробки вилучених із лісу дерев правил, застосовуючи скорочення правил для усунення надмірних та схожих правил прийняття рішень. Додатково, фактор інфляції дисперсії (VIF) та рекурсивне усунення ознак (RFE) застосовуються для аналізу правил прийняття рішень, щоб усунути мультиколінеарність та виділити найбільш значущі характеристики. Для визначення фінальної групи важливих ознак проводиться тест на статистичну значущість.

Запропонований метод RuleFit порівнюється з базовою моделлю (логістична регресія, LR) та еталонною моделлю (RF) за показниками якості класифікації, такими як точність (Accuracy), прецизійність (Precision), повнота (Recall), AUC ROC, F1 score та Brier score. Для перевірки ефективності та здійсненності запропонованої моделі RuleFit проведено експерименти на контрольованому (фіктивному) наборі даних. Результати експериментів свідчать про те, що запропонована модель RuleFit здатна успішно класифікувати деякі випадки, з якими логістична регресія не справляється, завдяки використанню правил прийняття рішень.

Для оцінки продуктивності моделі RuleFit проведено тестування на реальному фінансовому наборі даних. Результати оцінки свідчать про те, що запропоновану модель RuleFit можна успішно використовувати для вирішення реальних бізнес-задач, таких як оцінка кредитного ризику.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the operational research society, 54(6):627–635, 2003.
2. G. E. Batista, A. L. Bazzan, M. C. Monard, et al. Balancing training data for automated annotation of keywords: a case study. In WOB, pages 10–18, 2003.
3. Altman, E. I. (1968). Financial Ratios, Discriminant Analysis, and the Prediction of Corporate Bankruptcy. The Journal of Finance, 23(4), 589–609.
4. Anderson, R., & Hardin, C. (2014). Credit Scoring and Credit Control. Oxford University Press.
5. Basel Committee on Banking Supervision. (2000). Principles for the Management of Credit Risk. Bank for International Settlements.
6. Baesens, B., Roesch, D., & Scheule, H. (2016). Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS. Wiley.
7. Bellotti, T., & Crook, J. (2009). Support Vector Machines for Credit Scoring and Discovery of Significant Features. Expert Systems with Applications, 36(2), 3302–3308.
8. Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling Small-Business Credit Scoring by Using Logistic Regression, Neural Networks and Decision Trees. Intelligent Systems in Accounting, Finance & Management, 13(3), 133–150.
9. Bharath, S. T., & Shumway, T. (2008). Forecasting Default with the Merton Distance to Default Model. The Review of Financial Studies, 21(3), 1339–1369.
10. Breeden, J. L. (2014). Revolution in Credit Risk Management: The Basel II Securitization Framework and Beyond. Risk Books.

11. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.
12. Brown, I., & Mues, C. (2012). An Experimental Comparison of Classification Algorithms for Imbalanced Credit Scoring Data Sets. Expert Systems with Applications, 39(3), 3446–3453.
13. Caouette, J. B., Altman, E. I., Narayanan, P., & Nimmo, R. (2008). Managing Credit Risk: The Great Challenge for Global Financial Markets. John Wiley & Sons.
14. C. Bolton et al. Logistic regression and its application in credit scoring. PhD thesis, University of Pretoria, 2010.
15. P. Bracke, A. Datta, C. Jung, and S. Sen. Machine learning explainability in finance: an application to default risk analysis. Bank of England Working Paper, 2019.
16. Chen, S., & Zhang, G. (2019). Big Data Applications in the Management of Credit Risks: A Survey. Technological Forecasting and Social Change, 144, 210–220.
17. Cheng, C.-H., & Cheng, H.-W. (2017). An Improved Prediction of Credit Risk Using Gradient Boosted Decision Trees. Journal of Risk and Financial Management, 10(2), 9.
18. Cielen, D., & Dumoulin, A. (2016). Credit Risk Modelling with Python and Machine Learning. Addison-Wesley.
19. G. C. Cawley and N. L. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. The Journal of Machine Learning Research, 11:2079–2107, 2010.
20. Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent Developments in Consumer Credit Risk Assessment. European Journal of Operational Research, 183(3), 1447–1465.
21. Das, S. R., & Stein, R. M. (2018). Credit Scoring Models and Artificial Intelligence: Applications and Implications. Palgrave Macmillan.
22. Dev, A. (2019). Credit Risk Management in the Age of AI. MIT Press.

23. Dermine, J. (2009). Bank Valuation and Value-Based Management: Deposit and Loan Pricing, Performance Evaluation, and Risk Management. McGraw-Hill.
24. Duffie, D., & Singleton, K. J. (2003). Credit Risk: Pricing, Measurement, and Management. Princeton University Press.
25. Eisenbeis, R. A. (1977). Pitfalls in the Application of Discriminant Analysis in Business, Finance, and Economics. The Journal of Finance, 32(3), 875–900.
26. Fawcett, T. (2006). An Introduction to ROC Analysis. Pattern Recognition Letters, 27(8), 861–874.
27. Goodhart, C., & Hofmann, B. (2007). House Prices, Money, Credit, and the Macroeconomy. Oxford Review of Economic Policy, 24(1), 180–205.
28. Hand, D. J., & Henley, W. E. (1997). Statistical Classification Methods in Consumer Credit Scoring: A Review. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523–541.
29. Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
30. Hensher, D. A., & Jones, S. (2007). Forecasting Corporate Bankruptcy: A Recursive Partitioning Approach. Journal of Forecasting, 26(8), 523–537.
31. Hosmer, D. W., & Lemeshow, S. (2000). Applied Logistic Regression. Wiley.
32. Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit Scoring with a Data Mining Approach Based on Support Vector Machines. Expert Systems with Applications, 33(4), 847–856.
33. Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer Credit-Risk Models via Machine-Learning Algorithms. Journal of Banking & Finance, 34(11), 2767–2787.
34. Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking State-of-the-Art Classification Algorithms for Credit

Scoring: An Update of Research. European Journal of Operational Research, 247(1), 124–136.

35. Li, C., & Cao, Y. (2018). Deep Learning for Credit Risk Analysis. Journal of Financial Risk Management, 7(2), 73–81.
36. Liu, Y., & Schumann, L. (2005). Data Mining and Big Data Analytics in Financial Services: Credit Risk Prediction. Computational Intelligence Magazine, 2(2), 30–39.
37. Makridakis, S., & Hibon, M. (2000). The M3-Competition: Results, Conclusions, and Implications. International Journal of Forecasting, 16(4), 451–476.
38. Merton, R. C. (1974). On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. The Journal of Finance, 29(2), 449–470.
39. Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
40. Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. Journal of Accounting Research, 18(1), 109–131.
41. Ong, S. (2015). Artificial Intelligence in Credit Scoring and Financial Decision Making. Routledge.
42. Qiu, J., & Freeman, S. (2017). An Application of Machine Learning to Credit Risk. Journal of Risk Management in Financial Institutions, 10(4), 365–375.
43. Saunders, A., & Allen, L. (2010). Credit Risk Measurement In and Out of the Financial Crisis: New Approaches to Value at Risk and Other Paradigms. Wiley.
44. Sohn, S. Y., & Kim, H. S. (2012). A Study on Optimal Data Mining Techniques for Credit Scoring Models. Expert Systems with Applications, 39(11), 9825–9831.
45. Thomas, L. C., Crook, J. N., & Edelman, D. B. (2017). Credit Scoring and Its Applications. SIAM.

46. Tsai, C. F., & Wu, J. W. (2008). Using Neural Network Ensembles for Credit Risk Assessment. Expert Systems with Applications, 34(4), 2639–2649.
47. Vapnik, V. N. (1998). Statistical Learning Theory. Wiley.
48. Wang, X., & Hu, Y. (2010). Improved Credit Risk Evaluation Models with Data Mining Techniques. International Journal of Information Technology & Decision Making, 9(2), 329–354.
49. West, R. C. (1973). On the Prediction of Corporate Bankruptcy: Models and Tests. Journal of Finance, 28(1), 121–137.
50. Zhang, G., & Wei, M. (2016). A Survey of Credit Risk Modelling and Management Techniques. Journal of Financial Economics, 12(3), 289–305.
51. G. E. Batista, M. C. Monard, et al. A study of k-nearest neighbour as an imputation method. His, 87(251-260):48, 2002.
52. R. Blagus and L. Lusa. Smote for high-dimensional class-imbalanced data. BMC bioinformatics, 14(1):1–16, 2013.
53. E. Blancas. Model selection done right: A gentle introduction to nested cross-validation, 2022. 38, 39