Issue 16, 2024

https://doi.org/10.23939/sisn2024.16.167

УДК 004.89

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЛОГІСТИЧНОЇ РЕГРЕСІЇ ДЛЯ БІНАРНОЇ КЛАСИФІКАЦІЇ

Петро Кравець[1], Володимир Пасічник[2], Микола Проданюк[3], Ярослав Кісь[4]

Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж, Львів, Україна
[1] E-mail: Petro.O.Kravets@lpnu.ua, ORCID: 0000-0001-8569-423X
[2] E-mail: Volodymyr.V.Pasichnyk@lpnu.ua, ORCID: 0000-0002-5231-6395
[3] E-mail: Mykola.M.Prodaniuk@lpnu.ua, ORCID: 0000-0001-9544-3792
[4] E-mail: Yaroslav.P.Kis@lpnu.ua, ORCID: 0000-0003-3421-2725

У статті розглянуто практичні аспекти застосування логістичної регресії для бінарної класифікації даних. Логістична регресія визначає ймовірність належності об'єкта до одного із двох класів. Ця ймовірність обчислюється за допомогою сигмоїдної функції, аргументом якої є лінійна згортка вектора ознак об'єкта з ваговими коефіцієнтами, отриманими в процесі мінімізації логарифмічної функції втрат. Прогнозовані мітки класу визначаються порівнянням обчисленої ймовірності із заданим пороговим значенням.

Дослідження логістичної регресії виконано методом комп'ютерного моделювання. Для цього розроблено програмний комплекс, робота якого відтворює основні етапи логістичної регресії: підготовка вхідних даних, навчання, тестування з визначенням метрик якості бінарної класифікації, застосування методу логістичної регресії для класифікації даних на практиці.

У роботі вивчено вплив перекриття та дизбалансу класів у вхідному наборі даних на ефективність бінарної класифікації. Перекриття класів змодельовано формуванням вхідних даних на основі двох зміщених одна відносно одної функцій густини нормального розподілу випадкових величин. Дизбаланс класів імітується ймовірністю перемикання між цими функціями.

Показано, що в разі зменшення відстані між математичними сподіваннями функцій густини нормального розподілу або зростання дисперсії випадкових величин перекриття актуальних класів зростає, що призводить до збільшення кількості об'єктів, які класифікатор може віднести як до одного, так і до іншого класу.

Наближення ймовірності перемикання між функціями розподілу випадкових величин до крайніх значень одиничного інтервалу призводить до зростання дизбалансу класів, що проявляється у збільшенні кількості елементів вхідного набору даних, маркованих міткою того самого класу.

Експериментально підтверджено, що популярна у задачах бінарної класифікації метрика AUC ROC є залежною від ступеня перекриття класів і відносно стійкою до дизбалансу класів.

Ключові слова: комп'ютерне моделювання, логістична регресія, бінарна класифікація, аналіз даних, машинне навчання, перекриття класів, дизбаланс класів, градієнтний спуск, метрики якості класифікації.

Вступ

Сучасні інформаційні системи оперують великими обсягами даних, тому важливим є застосування ефективних методів і засобів їхнього аналізу [1–2]. Одним із таких методів є логістична

регресія, яка використовується для вирішення задач розділення об'єктів на окремі класи залежно від набору інформаційних ознак, що характеризують ці об'єкти [3–6].

Бінарна логістична регресія є статистичним методом, який використовується для прогнозування ймовірності належності об'єкта до одного із двох можливих класів. Це досягається шляхом моделювання логарифмічної функції втрат, яка пов'язує залежну (вихідну) змінну з лінійною комбінацією незалежних (вхідних) змінних. На відміну від лінійної регресії, що використовується для прогнозування числових значень, логістична регресія може працювати з категоріальними даними. Результатом логістичної регресії є ймовірність, обмежена діапазоном від 0 до 1. Після порівняння ймовірності з пороговим значенням формується бінарна мітка класу. У машинному навчанні, зокрема у задачі бінарної класифікації, мітки класів позначаються числами: 0 – негативний, 1 – позитивний.

Бінарна класифікація є однією з найпоширеніших задач у машинному навчанні [7], а логістична регресія є ефективним методом для її вирішення. Наприклад, вона дає змогу розв'язувати такі задачі, як-от: передбачення ризику захворювань у медицині, кредитного ризику у фінансовій сфері, виявлення емоційного забарвлення текстових повідомлень у соціальних мережах тощо.

Логістична регресія є ефективним методом аналізу даних та прогнозування ймовірностей. Основними перевагами логістичної регресії є простота реалізації, інтерпретованість результатів та можливість працювати з даними різного типу. Однак як і будь-який інший метод, логістична регресія має свої обмеження. Вона вимагає відносно лінійної залежності між змінними, що може обмежувати її застосування у більш складних задачах. Окрім того, для великих та складних наборів даних можуть знадобитися додаткові методи оптимізації. Незважаючи на певні обмеження, цей метод залишається одним із найпоширеніших і найбільш ефективних у галузі комп'ютерного аналізу даних.

Комп'ютерне моделювання логістичної регресії є важливим інструментом у задачах бінарної класифікації [8]. Воно забезпечує можливість опрацювання великих обсягів даних, автоматизацію процесу моделювання та високу точність класифікації. Засоби комп'ютерного моделювання логістичної регресії дають змогу аналітикам і дослідникам ефективно вирішувати на практиці складні завдання опрацювання даних та приймати обґрунтовані рішення для розв'язування різноманітних задач.

Комп'ютерне моделювання логістичної регресії дає змогу автоматизувати процес аналізу даних, що значно підвищує його ефективність. За допомогою сучасних програмних засобів, як-от Python, R або SPSS, можна швидко будувати, навчати та тестувати моделі на великих наборах даних [9]. Завдяки цьому можна отримувати високоточні прогнози та здійснювати аналіз значущості окремих змінних у моделі.

Незважаючи на наявність численних готових програмних засобів, усе частіше дослідники віддають перевагу використанню кастомних (від англ. customize) рішень із можливістю налаштування під свої потреби або розробці власних програмних засобів моделювання логістичної регресії [10–11]. Такі інструменти забезпечують більшу гнучкість, можливість врахування специфічних вимог задачі та оптимізацію обчислювальних ресурсів [12–13].

Розроблення власних програмних засобів для дослідження логістичної регресії стає необхідною у разі, коли стандартні інструменти не задовольняють потреби конкретного дослідницького проєкту. Наприклад, коли потрібно врахувати специфічні аспекти даних, розробити нові методи опрацювання або ж інтегрувати регресію в складнішу систему аналізу даних. Власні засоби дають змогу реалізувати унікальні функції, які можуть бути відсутні у комерційних чи відкритих програмних пакетах.

Власні розроблені програмні засоби надають можливість врахувати специфічні особливості моделювання логістичної регресії. Це особливо важливо в наукових дослідженнях, де часто виникає необхідність у предметно-орієнтованих застосуваннях, які зазвичай вимагають унікальних методологій та адаптації алгоритмів під конкретні практичні умови. Проте розроблення власних

програмних засобів вимагає високого професіоналізму розробників і значних затрат часу на алгоритмізацію, програмування, відлагодження, тестування та документування програми.

Комп'ютерне моделювання логістичної регресії дає можливість досліджувати розширення функціональних можливостей моделі: здатність працювати з великими обсягами даних, високорозмірними ознаками або нечіткими даними; здійснювати адаптацію логістичної регресії до специфічних умов конкретної сфери застосування; вдосконалювати методи навчання, особливо для великих обсягів даних; створювати добре інтерпретовані моделі, які зрозумілі для користувачів, що дає змогу аналізувати важливість кожної ознаки та її вплив на прогноз; досліджувати ефективність моделі в нових галузях застосування; розробляти нові алгоритми оптимізації або методи регуляризації, завдяки яким можна досягнути вищої точності прогнозування; розробляти нові моделі організації наборів даних для дослідження їхнього впливу на ефективність класифікації методом логістичної регресії.

До нових результатів цієї статті належать розроблені авторами:

1) модель призначених для класифікації вхідних даних на основі злиття двох випадкових послідовностей ознак об'єктів, розподілених за нормальним законом, що дало змогу вивчити вплив дизбалансу та перекриття класів на ефективність бінарної класифікації;

2) алгоритмічні та програмні засоби для моделювання бінарної класифікації методом логістичної регресії, які дають можливість отримувати результати, адаптовані до специфічних умов предметно-орієнтованого дослідження.

Огляд літературних джерел

Логістична регресія є однією з основних технік у сфері машинного навчання та статистики, особливо корисною для задач бінарної класифікації. За допомогою цієї моделі можна передбачати ймовірність настання певної події. Комп'ютерне моделювання логістичної регресії стало невід'ємною частиною аналізу великих даних та прийняття рішень на основі даних.

Основні теоретичні засади логістичної регресії були закладені у працях П. Ферхюльста (P. F. Verhulst), Е. Вілсона (E. B. Wilson), Дж. Вустера (J. Worcester), Дж. Берксона (J. Berkson), Д. Кокса (D. R. Cox), Г. Тейла (H. Theil), Дж. Нелдера (J. A. Nelder), Р. Веддерберна (R. W. M. Wedderburn).

П. Ферхюльст у роботі "Notice sur la loi que la population suit dans son accroissement" (1838) запропонував логістичну функцію для опису популяційного зростання, що пізніше стало основою для моделей логістичної регресії.

Е. Вілсон і Дж. Вустер у публікації "The Determination of L.D.50 and Its Sampling Error in Bio-Assay" (1943) використали логістичну функцію для біоаналізу, пропонуючи її як альтернативу пробіт-моделі. У роботі "The Law of Mass Action in Epidemiology" (1945) дослідники одними з перших застосували логістичну функцію в епідеміології для опису поширення інфекцій у популяції. Вони ввели ідею логістичного зростання, яка пізніше стала базисом для логістичної регресії.

Дж. Берксон у праці "Application of the Logistic Function to Bio-Assay" (1944) формально запропонував використовувати логістичну функцію для моделювання ймовірностей у біологічних експериментах і в такий спосіб започаткував логістичну регресію в сучасному розумінні.

Д. Кокс у роботі "The Regression Analysis of Binary Sequences" (1958) зробив важливий внесок у розвиток логістичної регресії для аналізу бінарних результатів, заклавши основи для її використання в різних галузях науки.

Важливий внесок у вдосконалення логістичної регресії зробив Г. Тейл. У публікації "A Multinomial Extension of the Linear Logit Model" (1969) він виконав узагальнення класичної логістичної моделі для ситуацій з більш ніж двома можливими результатами. Ця робота була ключовою для розвитку мультиноміальної логістичної регресії.

Дж. Нелдер і Р. Веддерберн у роботі “Generalized Linear Models” (1972) зробили значний внесок у розвиток статистичних методів, зокрема в розробку узагальнених лінійних моделей (GLM), які є важливим інструментом у статистиці та включають логістичну регресію як один із випадків GLM.

З розвитком комп’ютерних технологій з’явилися потужні інструменти для реалізації логістичної регресії. Існують мови програмування, програмні пакети та бібліотеки, які підтримують побудову моделей логістичної регресії. Вони різняться за функціональністю, призначенням, зручністю використання та спеціалізацією.

Наприклад, scikit-learn – одна з найпопулярніших бібліотек машинного навчання в Python, яка містить інструменти для побудови логістичних моделей та інших методів класифікації [14]. Логістична регресія реалізується за допомогою функції LogisticRegression, яка забезпечує гнучкість і простоту використання. Пакет scikit-learn легко інтегрується з іншими бібліотеками Python, такими як Pandas (для опрацювання даних) та Matplotlib/Seaborn (для візуалізації). Перевагами scikit-learn є легкість у використанні, швидке навчання моделі та підтримка широкого спектра функцій машинного навчання. Недоліком є обмежена підтримка складних статистичних аналізів у порівнянні з іншими інструментами.

Бібліотека statsmodels є бібліотекою статистичного моделювання на Python, яка надає інструменти для виконання статистичного аналізу даних, зокрема побудову логістичних регресійних моделей зі статистичними виведеннями [15]. Перевагами statsmodels є можливість глибокого статистичного аналізу та висока деталізація статистичних показників. Недоліком є більша складність у використанні порівняно з scikit-learn.

У мові програмування R для статистичних обчислень та аналізу даних функція glm (generalized linear models) використовується для побудови різних типів лінійних моделей, включно з логістичними регресійними моделями. Пакет caret надає зручний інтерфейс для виконання машинного навчання в R, зокрема побудову логістичних моделей [16–17]. Перевагами R є розширені можливості для роботи зі статистичними моделями, потужні інструменти для візуалізації даних та результатів, можливість роботи з великими обсягами даних, підтримка численних пакетів для аналізу даних та статистичного моделювання. До недоліків можна віднести помірну складність та потребу у знанні мови R.

Система статистичного аналізу SAS (Statistical Analysis System) є потужним інструментом для статистичного аналізу та бізнес-аналітики, що використовується у великих компаніях та дослідницьких організаціях [18]. Процедура SAS Proc Logistic у мові статистичного та аналітичного моделювання SAS є основним інструментом для побудови логістичних регресійних моделей та проведення аналізу даних з категоріальними вихідними змінними. Також SAS Enterprise Miner є комплексним інструментом для дослідження даних, який має можливості для побудови логістичних моделей та їхньої візуалізації. Містить зручний графічний інтерфейс для виконання задач логістичної регресії без необхідності написання коду. Перевагами SAS є підтримка різновидів логістичної регресії, зокрема мультиноміальної логістичної регресії, розширені можливості опрацювання великих обсягів даних, висока продуктивність і надійність. Недоліками є висока вартість і складність в освоєнні, підвищені вимоги до апаратного забезпечення.

Статистичний пакет для соціальних наук SPSS (Statistical Package for the Social Sciences) містить модуль Logistic Regression, який дає змогу будувати логістичні регресійні моделі та виконувати аналіз даних [19–20]. Передбачає зручний графічний інтерфейс для виконання логістичної регресії без необхідності написання коду, автоматичне створення звітів, потужні інструменти для опрацювання та аналізу даних. Перевагами SPSS є підтримка інтеграції з іншими програмними засобами та системами, можливість роботи з великими наборами даних, інтуїтивно зрозумілий інтерфейс користувача, розширені можливості для візуалізації даних та результатів. Недоліками є висока вартість ліцензії та обмежена гнучкість порівняно з програмними засобами, що вимагають написання коду.

Бібліотеки для машинного та глибокого навчання TensorFlow і Keras на Python підтримують складні моделі, зокрема логістичну регресію [21]. Призначені для створення багатошарових нейронних мереж, які можуть використовувати логістичну функцію активації для класифікації. Містять потужні інструменти для паралельних обчислень та оптимізації. Перевагами є підтримка складних і масштабованих моделей, можливість роботи з великими наборами даних та паралельними обчисленнями, гнучкість у визначенні та налаштуванні архітектури моделі. Недоліками є певна складність у налаштуванні та освоєнні, інструмент не завжди зручний для виконання стандартних задач логістичної регресії.

У MATLAB є набір функцій Statistics and Machine Learning Toolbox для статистичного аналізу та машинного навчання, зокрема побудову логістичних регресійних моделей [22].

Це лише деякі з програмних пакетів та бібліотек, які можна використовувати для логістичного моделювання та аналізу даних. Вибір конкретного інструменту залежить від потреб, навичок програмування та предметної сфери дослідження.

В останні роки увага дослідників зосереджена на вдосконаленні алгоритмів логістичної регресії для опрацювання великомасштабних, різнотипних, неповних та нечітких даних. Це відображає сучасні виклики в обробці даних, де традиційні методи логістичної регресії не завжди справляються з реальними завданнями.

Логістична регресія потребує оптимізації для роботи з великими обсягами даних (big data), що вимагає розроблення ефективних алгоритмів для зменшення обчислювальних витрат та часу навчання. Використання логістичної регресії для задач з високою розмірністю даних і великою кількістю ознак вимагає нових рішень для зменшення складності моделі [23].

У задачах, де використовуються різнотипні дані (heterogeneous data), наприклад, структуровані (табличні) і неструктуровані (тексти, зображення тощо), постає питання інтеграції різних типів даних у єдину модель логістичної регресії [24].

Неповні дані (missing data) є поширеною проблемою в практичних задачах, тож необхідно вдосконалювати методи для коректного використання логістичної регресії за таких умов [25].

Моделі логістичної регресії не завжди добре справляються із зашумленими даними (noisy data) [26] та нечіткими даними (fuzzy data) [27], які містять значну кількість спотворених даних або нечіткість у визначенні класів. Це потребує адаптацій або нових підходів до попередньої опрацювання даних і побудови моделей.

Також потребують вирішення проблеми інтерпретації моделей, нелінійної взаємодії ознак об'єктів та балансування класів.

Із зростанням складності моделей даних та алгоритмів їхнього опрацювання виникає питання їхньої прозорості та зрозумілості для користувачів. Збереження інтерпретованості моделі є важливим аспектом у використанні логістичної регресії в багатьох галузях [28].

Логістична регресія за своєю природою є лінійним методом. Однак для складніших даних може знадобитися моделювання нелінійних взаємодій між ознаками [29].

Логістична регресія може погано працювати на малих обсягах класів диспропорційної вибірки, що вимагає розроблення нових стратегій для врахування цього фактору [30].

Вирішення цих проблем робить логістичну регресію більш придатною для сучасних складних та об'ємних завдань.

Також важливими є питання підвищення стійкості моделей логістичної регресії до перенавчання. Для цього використовують різні методи регуляризації функції логарифмічних втрат, наприклад, L1, L2 та інші [31–32].

Усе більшої популярності набувають гібридні моделі, що поєднують логістичну регресію з іншими методами машинного навчання, як-от нейронними мережами [33].

Логістична регресія знайшла застосування у багатьох галузях. Наприклад, у медицині вона використовується для прогнозування ймовірності захворювань [34]. У фінансах логістична регресія допомагає прогнозувати імовірність дефолту позичальників, що розглянуто у роботі [35]. Важли-

вим є також моніторинг відгуків у соціальних мережах, як це описано у [36]. Перелік застосувань логістичної регресії можна значно розширити. Ці та інші приклади ілюструють, як комп'ютерне моделювання логістичної регресії може бути використане для вирішення реальних задач.

Використання кастомних або розроблення власних програмних засобів для моделювання логістичної регресії дає змогу враховувати специфічні вимоги певних галузей. Це особливо важливо в наукових дослідженнях, де потрібно врахувати специфіку даних або вимоги до інтеграції з іншими моделями.

Отже, комп'ютерне моделювання логістичної регресії є важливою складовою сучасного аналізу даних. Розвиток інструментів і методів дає змогу дослідникам ефективно застосовувати цю модель у різних галузях. Однак виклики, пов'язані з якістю вхідних даних, їхньою структурою, різноманітністю типів, масштабованістю та інтерпретованістю у різних галузях застосування стимулюють подальший розвиток кастомних рішень.

Мета роботи

Метою цієї роботи є комп'ютерне моделювання логістичної регресії для дослідження впливу структури вхідних даних та регульованих параметрів моделі на ефективність бінарної класифікації.

Досягнення мети забезпечується виконанням таких завдань: 1) побудова алгоритмів та програм моделювання логістичної регресії для бінарної класифікації ознак об'єктів; 2) підготовка набору даних для дослідження моделі, розділення даних на тренувальний та тестовий набори; 3) вибір параметрів моделі, які впливають на точність класифікації, збіжність та швидкість навчання методу логістичної регресії; 4) навчання логістичної регресії на послідовності вхідних даних; 5) перевірка роботи моделі на тестових наборах даних, які не використовувалися для навчання; 6) оцінка ефективності методу логістичної регресії на основі метрик бінарної класифікації; 7) оптимізація моделі логістичної регресії корегуванням її параметрів; 8) застосування навченої логістичної регресії для бінарної класифікації нових об'єктів; 9) аналіз та інтерпретація отриманих результатів, формулювання рекомендацій щодо предметно-орієнтованого застосування методу логістичної регресії.

Основна частина

Моделювання вхідних даних

Навчальна і тестова вибірки для бінарної класифікації формуються на основі наявних експериментальних даних, вид і значення яких визначається конкретною предметною сферою. Так, для класифікації текстової інформації за її емоційним окрасом використовують масиви відгуків у комерційних чи соціальних мережах, у медицині – дані щодо перебігу певного захворювання у спостережуваної групи пацієнтів, у біології – бази даних біомедичної та геномної інформації, в економіці – національні підсумкові дані щодо макроекономічних та фінансових показників, інтегральні дані Світового банку, Міжнародного валютного фонду тощо.

Для дослідження логістичної регресії та інших методів аналізу даних існує декілька популярних ресурсів, баз даних, бібліотек та наборів даних.

Наприклад, UCI Machine Learning Repository – один із найвідоміших ресурсів для машинного навчання, який містить широкий спектр наборів даних для тестування алгоритмів класифікації, включаючи логістичну регресію. Змагальна платформа Kaggle для проєктів у сфері аналізу даних та машинного навчання з великою кількістю різноманітних наборів даних, зокрема ті, що можна використовувати для логістичної регресії. Відкрита платформа OpenML для спільного вивчення та експериментування з алгоритмами машинного навчання, що містить багато наборів даних, які можна використовувати для тестування логістичної регресії та інших алгоритмів. Популярна бібліотека scikit-learn (також відома як sklearn) для машинного навчання в Python, яка містить кілька вбудованих наборів даних, призначених для тестування алгоритмів класифікації, включно з логістичною регресією. Репозиторій StatLib, який містить набори даних для статистичних дослі-

джень, які можна використовувати для вивчення та тестування різних моделей, включно з логістичною регресією. Пошукова система Google Dataset Search від Google, яка дає змогу знаходити набори даних у відкритому доступі на різних вебсайтах, включаючи ті, що можна використовувати для логістичної регресії.

У разі дослідження ефективності методів бінарної класифікації зручно використати модельні дані, згенеровані за випадковими розподілами. Це може бути корисним для розуміння та оцінки роботи алгоритму логістичної регресії в різних умовах або для порівняння його з іншими методами аналізу даних.

Генератори випадкових величин можуть створювати дані з різною структурою та характеристиками, як-от: різні розподіли, різні взаємозв'язки між змінними, наявність аномальних відхилень, інші властивості, що відображають реальні дані. За допомогою таких даних можна вивчати те, як логістична регресія веде себе в різних сценаріях та які фактори впливають на її ефективність.

Проведення експериментів із логістичною регресією на основі штучних або модельних даних є корисним інструментом для дослідження та вдосконалення цього методу аналізу даних для розуміння його переваг та обмежень в різних умовах роботи.

Перевагами використання модельних даних є:

1. Контрольовані умови: використання модельних даних дає змогу контролювати різні аспекти експерименту, змінювати його параметри, систематично досліджувати вплив різних факторів на результати та зробити висновки про причинно-наслідкові зв'язки. Такий підхід дає змогу зрозуміти, як алгоритм логістичної регресії працює в умовах контрольованого експерименту і допомагає встановити його межі застосовності.

2. Швидкість та доступність: модельні дані можуть бути швидше та легше згенеровані, ніж збір реальних даних. Це дає змогу проводити багато експериментів за короткий час і дає змогу швидше отримувати та опрацьовувати результати.

3. Повторюваність: модельні дані можуть бути повністю документованими та відтворюваними, що дає змогу іншим дослідникам повторити дослідження та перевірити його результати.

Однак важливо враховувати, що модельні дані можуть не відображати всі аспекти реальних даних і можуть бути обмежені у своїй репрезентативності. Тому важливо забезпечити, щоб результати, отримані на модельних даних, були перевірені та порівняні з реальними даними там, де це можливо.

У цій роботі формування навчальної вибірки бінарної класифікації пропонується виконати за допомогою двох генераторів випадкових величин, розподілених за обраними законами розподілу, – рівномірним, експоненційним, нормальним або іншим. Схема генерування таких модельних даних зображена на рис. 1.

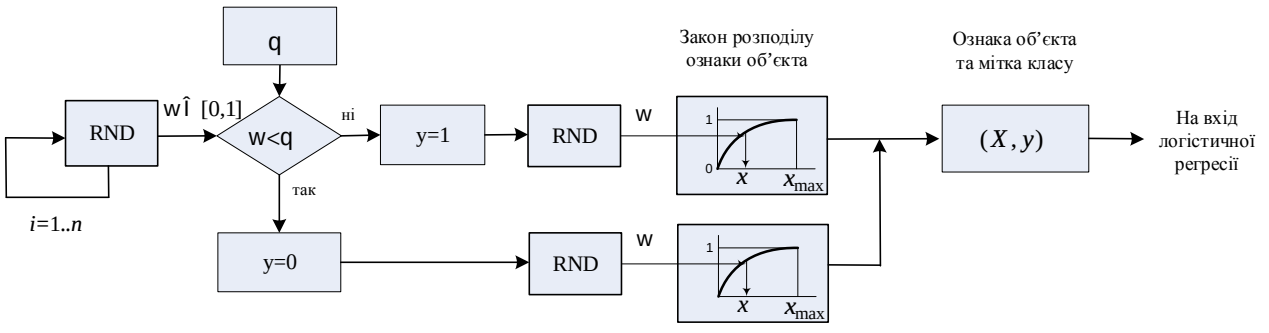


Рис. 1. Схема генерування вхідних даних

Кожен із генераторів формує ознаки об'єктів, що належать тому самому класу. Для того, щоб у вхідну вибірку потрапили об'єкти з двох класів, здійснюється перемикання між генераторами з імовірністю q:

$$y = \begin{cases} 0, & w < q \\ 1, & w \geq q \end{cases},$$

де $y \in \{0,1\}$ – мітка класу; $w \in [0,1]$ – випадкове число, рівномірно розподілене в інтервалі [0, 1].

Імовірність q регулює дизбаланс класів. Дизбаланс – це кількісна перевага у вибірці даних одного з двох класів. Якщо $q < 0.5$, то частіше буде вибиратися клас $y = 1$. Якщо $q > 0.5$, то частіше буде вибиратися клас $y = 0$. Якщо $q = 0.5$, то обидва класи будуть вибиратися рівноймовірно. Значення $S_{imbalance} = |2q - 1| \in [0,1]$ можна розглядати як ступінь дизбалансу класів.

Для генерування випадкових ознак у цьому експерименті використано функцію інтегрального (кумулятивного) нормального закону розподілу, яка приймає значення з інтервалу [0, 1]:

$$F(x) = \frac{1}{s\sqrt{2p}} \int_{-\infty}^{x} e^{-\frac{(t-m)^2}{2s^2}} dt,$$

де x – числова ознака об'єкта; m – її математичне сподівання; $s = \sqrt{d}$ – середньоквадратичне відхилення; d – дисперсія; t – змінна інтегрування.

Для цього генерується випадкове дійсне число w, рівномірно розподілене в інтервалі [0, 1], яке вважається значенням функції інтегрального закону розподілу. Далі за значенням функції відшукується її аргумент, який і буде однією із характеристичних ознак об'єкта (рис. 2):

$$x = \arg F(x) : F(x) = w.$$

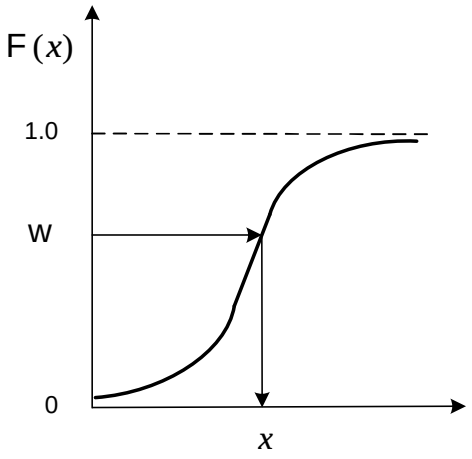


Рис. 2. Модель генерування характеристичних ознак об'єкта

Визначення аргумента функції $F(x)$ за її значенням w здійснюється за відомою формулою:

$$x = F^{-1}(w) = m + s\sqrt{2}\,\mathrm{erf}^{-1}(2w - 1),$$

де $w \in [0,1]$; erf – функція помилок Гауса, яка визначається як:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{p}} \int_{0}^{x} e^{-t^2} dt.$$

Зворотна функція помилок є сумою компонентів такого ряду:

$$\mathrm{erf}^{-1}(x) = \sum_{k=0}^{\infty} \frac{c_k}{2k+1}\left(\frac{\sqrt{\pi}}{2}x\right)^{2k+1},$$

де $c_0 = 1;\ c_k = \sum_{m=0}^{k-1} \frac{c_m c_{k-1-m}}{(m+1)(2m+1)}$.

Перекриття класів виникає тоді, коли у вибірці є дані про об'єкти, які мають однакові або подібні ознаки і які марковані різними класами, що значно ускладнює завдання їхньої класифікації.

Перекриття класів моделюється двома функціями густини $\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ імовірностей нормального розподілу випадкових величин, які є зміщеними одна щодо іншої і частково перекриваються, як це показано на рис. 3 (графіки функцій зображено схематично).

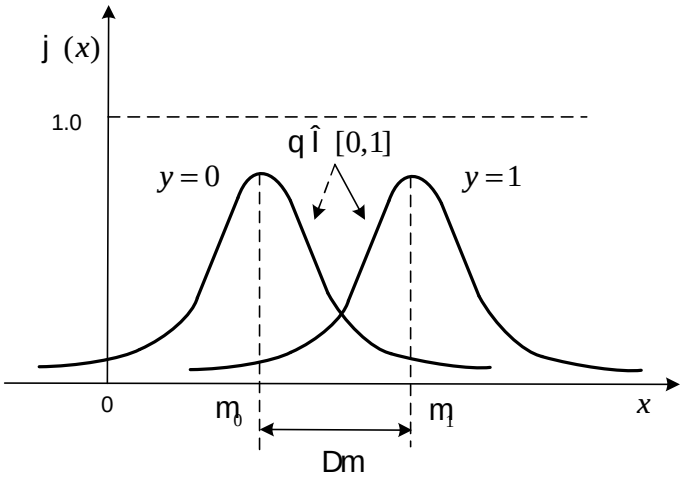


Рис. 3. Модель формування випадкових наборів вхідних даних

Ступінь перекриття класів визначається відношенням спільної площі під лініями перетину графіків цих функцій до максимально можливої такої площі. Оскільки $\int_{-\infty}^{\infty} \varphi_0(x)dx = \int_{-\infty}^{\infty} \varphi_1(x)dx = 1$, то ступінь перекриття класів дорівнює спільній площі під лініями перетину графіків:

$$S_{overlapping} = \int_{-\infty}^{\infty} \min(\varphi_0(x), \varphi_1(x))dx \Big/ \int_{-\infty}^{\infty} \varphi_0(x)dx = \int_{-\infty}^{\infty} \min(\varphi_0(x), \varphi_1(x))dx \in [0,1].$$

Відстань $\Delta m = |\mu_1 - \mu_0|$ між математичними сподіваннями μ_0 та μ_1 та дисперсія $d = \sigma^2$ випадкового розподілу впливають на ступінь перекриття класів, яке означає, що об'єкти з близькими за значеннями ознаками x можуть бути віднесені як до класу 0, так і до класу 1. У разі наближення Δm до нуля перекриття класів збільшується і за $\Delta m = 0$ класи стають нероздільними. У разі зростання Δm таке перекриття зменшується аж до практично повністю розділених класів.

Перекриття та дизбаланс класів у вхідному наборі даних будуть впливати на значення елементів матриці помилок і на показники якості класифікації [37].

Згенеровану вибірку $(X, y) = (X_i, y_i)_{i=1}^{n}$ далі використовують для навчання або тестування.

Цільова функція логістичної регресії

У схемі бінарної логістичної регресії значення сигмоїдної функції:

$$p_i = \sigma(z_i) = \frac{1}{1+e^{-z_i}} \tag{1}$$

інтерпретується як прогнозована імовірність належності i-го об'єкта до одного із двох класів. Параметр $z_i = W^{\mathrm{T}} X_i$ є згорткою значень ознак об'єкта X_i з ваговими коефіцієнтами W.

Цільова функція логістичної регресії описує процес мінімізації середньоквадратичного відхилення ймовірності $p_i \in [0,1]$ прогнозованого класу від актуального значення мітки класу $y_i \in \{0,1\}$ по всій вибірці об'єктів:

$$F = \frac{1}{n}\sum_{i=1}^{n} (p_i - y_i)^2 \to \min_W. \tag{2}$$

Мінімізація цільової функції здійснюється за параметром W, що є вектором ваг ознак об'єкта. Інакше необхідно підібрати таке значення параметра W, яке забезпечить найменше середньоквадратичне відхилення ймовірностей прогнозованих від актуальних міток класів об'єктів.

Побудована на основі сигмоїди (1) цільова функція (2) є багатоекстремальною, що ускладнює пошук глобального мінімуму. Тому замість (2) використовують унімодальну функцію логарифма втрат, яка виводиться з оцінки максимальної правдоподібності (maximum likelihood method), в основу якої покладено розподіл Бернуллі:

$$P[Y = y \mid X = x] = p^y (1-p)^{1-y},$$

де $P[Y = y \mid X = x]$ – умовна імовірність того, що на виході буде отримано значення $y \in \{0,1\}$, якщо на вхід класифікатора подано значення x; p та $(1-p)$ – імовірності результатів випадкового есперименту з двома можливими значеннями.

Враховуючи (1), запишемо:

$$p^y (1-p)^{1-y} = \sigma(z)^y (1-\sigma(z))^{1-y}.$$

Тоді для незалежних випробувань імовірність спільного настання n подій визначається добутком імовірностей настання кожної події:

$$L(W) = \prod_{i=1}^{n} \sigma(z_i)^{y_i} (1-\sigma(z_i))^{1-y_i}. \tag{3}$$

Виходячи з критерію максимальної правдоподібності, необхідно знайти таке значення ваг W, яке максимізує функцію ймовірності $L(W)$.

Логарифмування функції (3) забезпечує перехід від множення до додавання, що значно спрощує обчислення на великих вибірках даних:

$$\ln L(W) = \sum_{i=1}^{n} \left[y_i \cdot \ln \sigma(z_i) + (1-y_i) \cdot \ln (1-\sigma(z_i)) \right] \to \max_W. \tag{4}$$

Для порівняння результатів класифікації на різних за обсягом вибірках даних суму (4) усереднюють по обсягу вибірки. Після інверсії знаку функції (4) отримаємо цільову функцію логістичної регресії у такому вигляді:

$$F(W) = -\ln L(W) = -\frac{1}{n}\sum_{i=1}^{n} \left[y_i \cdot \ln \sigma(z_i) + (1-y_i) \cdot \ln (1-\sigma(z_i)) \right] \to \min_W. \tag{5}$$

Функцію (5) називають функцією логарифмічних втрат LogLoss. Аналіз її складових показує, що вона є унімодальною з одним глобальним мінімумом. Для мінімізації такої функції зручно використати метод градієнтного спуску [38–39].

Знайдене оптимальне значення вагових коефіцієнтів W^* використовується для визначення ймовірності $p = \sigma\left(W^{*\mathrm{T}} X\right)$ належності ознак X деякого об'єкта до одного з бінарних класів.

Детальніше математичну модель логістичної регресії розглянуто у наших попередніх роботах [40–41].

Етапи логістичної регресії

Застосування логістичної регресії для бінарної класифікації даних складається з таких основних етапів: підготовка даних, навчання, тестування, інтерпретація результатів, верифікація моделі та практична (реальна) класифікація.

Етап *підготовки даних* передбачає збір даних, кодування категоріальних ознак, вибір значущих ознак, фільтрування пропущених значень, масштабування ознак із різними діапазонами значень.

Метою *навчання логістичної регресії* є визначення таких значень вагових коефіцієнтів ознак об'єктів, які мінімізують функцію логарифмічних втрат і які забезпечать високу точність класифікації нових об'єктів.

Алгоритм навчання логістичної регресії складається з таких кроків:

1. Ввести матрицю вхідних даних $[X, Y] = [X_i, y_i]_{i=1}^{n}$, де n – довжина вибірки об'єктів, які підлягають класифікації; $X_i = \left(x_{i,j}\right)_{j=0}^{m}$ – вектор ознак i-го об'єкта класифікації (рядок матриці), де $x_{i,0} = 1$ для $i = 1..n$; m – кількість ознак об'єкта; $y_i \in \{0,1\}$ – актуальна мітка класу i-го об'єкта; $W_0 = \left(w_j\right)_{j=0}^{m}$ – початкове значення вектора ваг ознак об'єкта.
2. Задати початковий номер ітерації $t = 0$ для обчислення вектора вагових коефіцієнтів.
3. Обчислити градієнт логарифмічної функції втрат за ваговими коефіцієнтами:

$$\nabla F(W_t) = \left(\frac{\partial F(W_t)}{\partial w_{t,j}}\right)_{j=0}^{m} = \left(\frac{1}{n}\sum_{i=1}^{n}\left(\sigma\left(z_i\right) - y_i\right)\times x_{i,j}\right)_{j=0}^{m},$$

де $z_i = W_t^{\mathrm{T}} X_i = \sum_{j=0}^{m} w_{t,j} x_{i,j}$ – згортка значень ознак i-го об'єкта з поточними вагами,

$\sigma(z_i) = \dfrac{1}{1+e^{-z_i}}$ – значення сигмоїдної функції.

4. Обчислити нове наближення вектора ваг:

$$W_{t+1} = W_t - \eta_t \nabla F(W_t),$$

де $\eta_t > 0$ – поточне значення кроку методу, яке визначається методом Барзилай-Борвейна (J. Barzilay, J. Borwein) [42] або з умови Вулфа (P. Wolfe) [43]. Також крок методу можна задати як монотонно спадну послідовність додатних величин:

$$\eta_t = \eta t^{-\alpha} > 0,$$

де $\eta > 0$ – його початкове значення, $\alpha > 0$ – параметр, що визначає швидкість зменшення кроку. Для збіжності градієнтного методу параметри η та α можна отримати із загальних умов збіжності Роббінса-Монро [44]: $\eta_t \underset{t\to\infty}{\to} 0$, $\sum_{t=1}^{\infty} \eta_t = \infty$, $\sum_{t=1}^{\infty} \eta_t^2 < \infty$.

5. Перевірити умову точності навчання алгоритму $\|W_{t+1} - W_t\| < \varepsilon$, де $\|\cdot\|$ – евклідова норма вектора; $\varepsilon > 0$ – задана точність навчання. Якщо умова не виконується, то перейти до наступної ітерації навчання $t := t+1$, початок якої визначається кроком 3.

6. Інакше, запам'ятати вектор оптимальних ваг $W^* = W_t$ і зупинити алгоритм.

Після навчання ваг здійснюється *тестування логістичної регресії*. Навчальна і тестова вибірки не повинні перекриватися. Рекомендоване відношення їхніх обсягів визначається як 7:3.

Метою тестування є перевірка здатності моделі узагальнювати отримані у процесі навчання знання на нові, невідомі дані та визначення її реальної ефективності на основі аналізу відповідних метрик. Метрики якості класифікатора обчислюються на основі елементів матриці помилок (Confusion Matrix) [45], які приймають такі значення:

- TP (True Positive) – істинно позитивний, що визначає кількість об'єктів, класифікованих як позитивні (такі, що належать позитивному класу) і які справді є такими;
- TN (True Negative) – істинно негативний, що визначає кількість об'єктів, класифікованих як негативні (такі, що належать негативному класу) і які справді є такими;
- FP (False Positive) – хибно позитивний, що визначає кількість об'єктів, класифікованих як позитивні (такі, що належать позитивному класу), але фактично є негативними (повинні належати негативному класу);
- FN (False Negative) – хибно негативний, що визначає кількість об'єктів, класифікованих як негативні (такі, що належать негативному класу), але справді є позитивними (повинні належати позитивному класу).

Алгоритм тестування логістичної регресії складається з таких кроків:

1. Увести матрицю ознак $[\hat{X}, Y] = \{\hat{X}_i, y_i\}_{i=1}^{k}$ об'єктів тестової вибірки, де k – довжина тестової вибірки; $\hat{X}_i = \left(\hat{x}_{i,j}\right)_{j=0}^{m}$ – вектор ознак i-го об'єкта тестової вибірки, де $\hat{x}_{i,0} = 1$ для $i = 1..k$; m – кількість ознак об'єкта; $y_i \in \{0,1\}$ – актуальна мітка класу i-го об'єкта.

2. Задати поріг $\tau \in (0,1)$ бінарної класифікації. Зазвичай поріг $\tau = 0.5$.

3. Кроки 4–7 виконати для кожного об'єкта $i = 1..k$ тестової вибірки.

4. Обчислити згортку z_i ознак $\hat{X}_i$ i-го об'єкта з попередньо навченими вагами W^*:

$$z_i = W^{*\mathrm{T}} \hat{X}_i = \sum_{j=0}^{m} w^*_{t,j} \hat{x}_{i,j}.$$

5. Обчислити значення сигмоїдної функції $\sigma(z_i) = \dfrac{1}{1+e^{-z_i}}$.

6. Порівняти значення сигмоїдної функції з порогом τ для вироблення прогнозованої мітки класу $\hat{y}_i$. Якщо $\sigma(z_i) < \tau$, то $\hat{y}_i = 0$, інакше $\hat{y}_i = 1$.

7. Порівняти прогнозовану мітку класу $\hat{y}_i$ з актуальною міткою y_i для обчислення елементів TP, TN, FP та FN матриці помилок.

8. На основі елементів матриці помилок обчислити метрики якості класифікації, наприклад, Acc, PPV, TPR, FPR, F_1, AUC ROC або інших [41], [45].

9. Аналізуючи значення метрик класифікації прийняти рішення про ефективність класифікатора. За необхідності змінити значення порогу $\tau \in (0,1)$ та повторити тестування з метою оптимізації метрик якості класифікації.

Інтерпретація результатів у контексті конкретного застосування та *верифікація моделі* на відповідність вимогам завершують формування моделі.

Якщо класифікатор на основі моделі логістичної регресії успішно пройшов усі попередні етапи, то він готовий до практичного розпізнавання класів нових об'єктів.

Практична класифікація методом логістичної регресії полягає у виконанні кроків 1–6 алгоритму тестування з використанням робочої вибірки $\tilde{X}$ даних замість даних тестової вибірки $\hat{X}$. Для визначення мітки прогнозованого класу $\hat{y}_i$ обчислюється сигмоїдна функція $\sigma\left(W^{*\mathrm{T}}\tilde{X}_i\right)$, значення якої порівнюється з порогом τ.

Програмні засоби комп'ютерного моделювання

Для комп'ютерного моделювання логістичної регресії розроблено об'єктно-орієнтований програмний комплекс мовою C++, який працює під керуванням операційної системи Windows та складається з таких компонентів:

1. Формування та підготовка даних, що передбачає формування модельних даних або попередню підготовку реальних даних: фільтрування недопустимих значень, кодування категоріальних ознак, нормування, розділення даних на тренувальний та тестовий набори.
2. Вибір та налаштування параметрів моделі, як-от: швидкість навчання, регуляризаційні параметри, значення порогу класифікації тощо.
3. Навчання моделі на тренувальному наборі даних для мінімізації функції логарифмічних втрат та визначення оптимального набору вагових коефіцієнтів ознак об'єктів.
4. Валідація моделі на тестовому наборі даних, обчислення метрик якості моделі (наприклад, точність, чутливість, специфічність тощо) для визначення її ефективності.
5. Експериментування з різними значеннями параметрів (наприклад, кроком алгоритму навчання, значенням порогу класифікації, дизбалансом та перекриттям класів), визначення оцінки їхнього впливу на ефективність класифікатора, оптимізація моделі.
6. Текстове та графічне представлення результатів навчання логістичної регресії, що включає у себе виведення оцінок ефективності моделі, візуалізацію процесу навчання моделі, візуалізацію впливу параметрів моделі (наприклад, порогу класифікації, дизбалансу та перекриття класів) на результати класифікації.

Загалом розроблений програмний комплекс не має підвищених вимог до характеристик апаратних засобів. Такі вимоги можуть виникати для великих обсягів даних.

Розроблений програмний засіб успішно протестовано на різних наборах модельних даних.

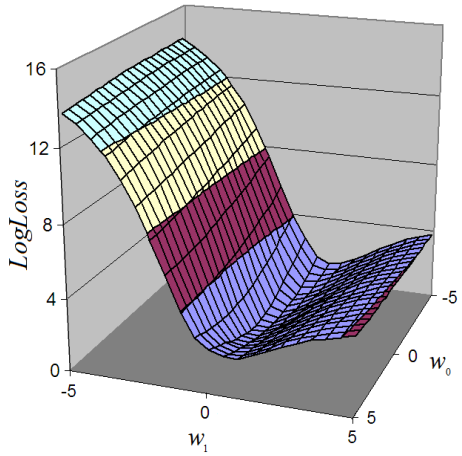
Поточний проєкт програми можна розглядати як draft-версію, особливо, що стосується його інтерфейсу, застосування різних методів навчання логістичної регресії для досягнення вищої точності прогнозування та можливості роботи з багатовимірними, нечисловими або нечіткими даними.

Контрольний приклад

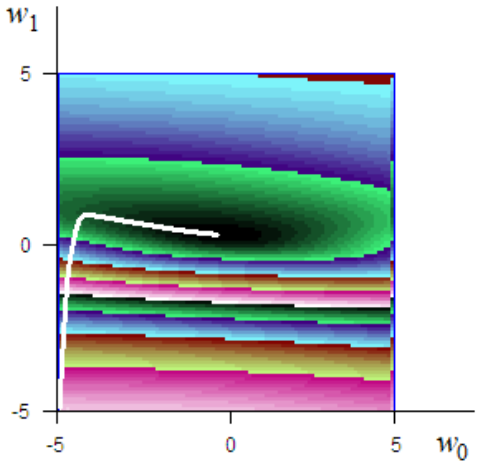
Навчання класифікатора, побудованого на основі бінарної логістичної регресіїї, виконано градієнтним методом найскорішого спуску на поверхні функції логарифмічних втрат LogLoss, визначеної у просторі вагових коефіцієнтів w_0 ´ w_1. Метою навчання є пошук оптимальних значень w_0^* та w_1^*, які мінімізують функцію LogLoss. Динамічний крок градієнтного методу визначається параметрами $\gamma = 1,\ \alpha = 0.1$. Точність методу задається значенням $\varepsilon = 0.001$. Початкові значення вагових коефіцієнтів $w_0 = w_1 = -5$. Обсяг вибірки в усіх експериментах прийнято рівним $n = 1000$ елементів даних. Усі параметри навчання є регульованими і можуть бути зміненими.

На рис. 4 подано 3D- та 2D-зображення функції LogLoss без масштабування та трьох варіантів масштабування вхідних даних: стандартизація, нормалізація та робастне масштабування [41]. Зображення отримано для таких параметрів змодельованих даних: математичні сподіваннями двох функцій густини нормального розподілу $m_0 = -1$, $m_1 = 5$; дисперсія розподілу $d = 25$; імовірність перемикання між функціями $q = 0.5$; поріг відсікання $t = 0.5$; ділянка визначення функції LogLoss: $w_0 \in [-5,5]$, $w_1 \in [-5,5]$. Зображення 2D отримано у вигляді набору проєкцій ліній рівня функції LogLoss на площину $w_0 \times w_1$. Також 2D-зображення містять траєкторії пошуку мінімального значення функції LogLoss.

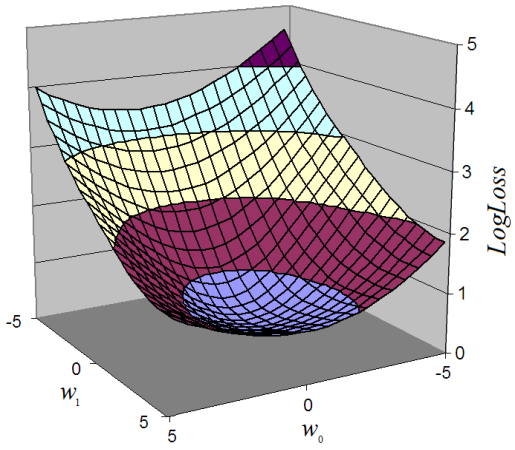
Масштабування означає приведення даних до певного діапазону значень. Масштабування вхідних даних сприяє прискоренню процесу навчання, покращує збіжність градієнтного методу мінімізації функції логарифмічних втрат та допомагає моделі знаходити оптимальні вагові коефіцієнти для кожної ознаки, що підвищує класифікаційні можливості алгоритму. Крім того, масштабування підвищує точність моделі, запобігаючи домінуванню ознак з великими значеннями над ознаками з меншими значеннями. Масштабування вирівнює внески різних ознак, використовує їх збалансовано і в такий спосіб позитивно впливає на результати класифікації. Масштабування допомагає зробити модель менш чутливою до коливань у даних, що сприяє її стабільності та більшій узагальненості.



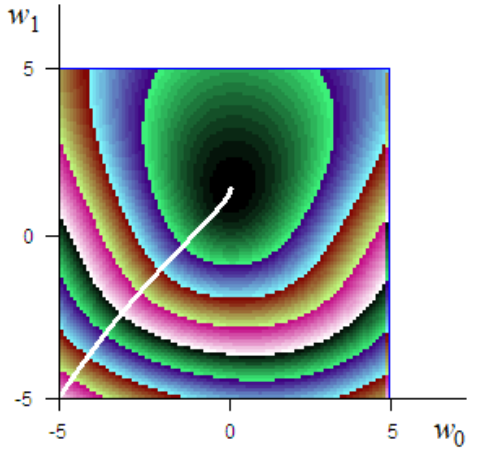
а) 3D LogLoss без масштабування



б) 2D LogLoss без масштабування

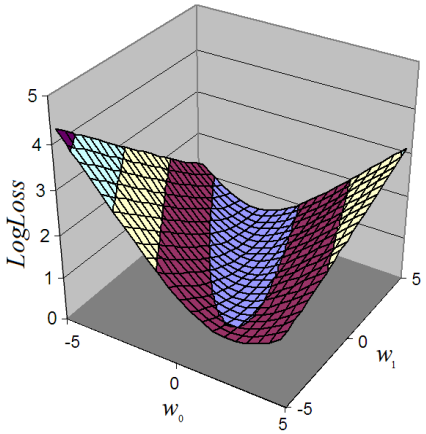


в) 3D LogLoss зі стандартизацією

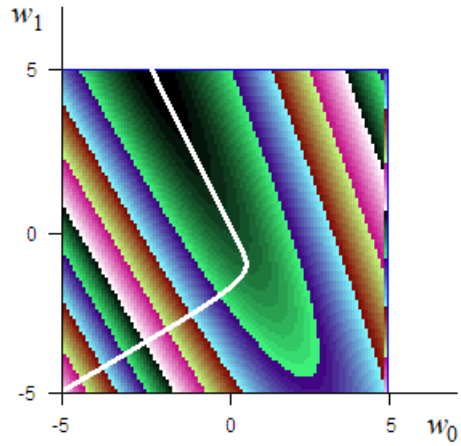


г) 2D LogLoss зі стандартизацією

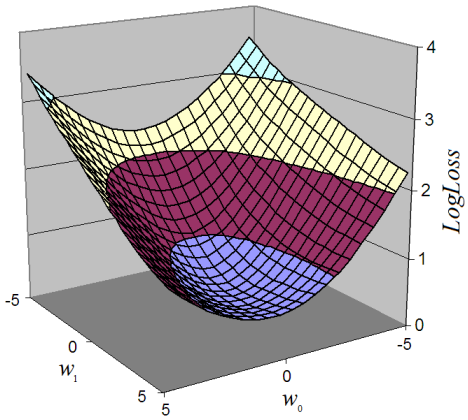
Рис. 4. Вплив масштабування вхідних даних на функцію логарифмічних втрат LogLoss



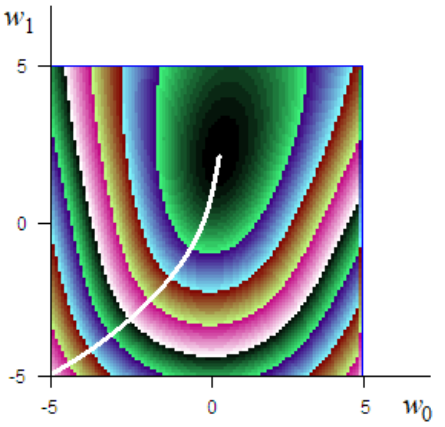
д) 3D LogLoss з нормалізацією



е) 2D LogLoss з нормалізацією



ж) 3D LogLoss з робастним масштабуванням



и) 2D LogLoss з робастним масштабуванням

Рис. 4. (Продовження). Вплив масштабування вхідних даних на функцію логарифмічних втрат LogLoss

Результатом масштабування є зміна діапазону значень вхідних даних, що, як видно на рис. 4, призводить до певної деформації поверхні функції LogLoss і відповідно до зміни оптимальних значень вагових коефіцієнтів w_0^*, w_1^* та до зміни траєкторії і часу їхнього пошуку.

Вивчимо вплив відстані $Dm = |m_1 - m_0|$ між математичними сподіваннями m_0, m_1 і дисперсії d -функції густини нормального розподілу ознак об'єктів вхідної вибірки на перекриття класів та вплив імовірності q на дизбаланс класів. Перекриття та дизбаланс класів по-різному впливають на ефективність роботи класифікатора.

Однією з найбільш інформативних оцінок якості класифікатора є метрика AUC (Area Under the Curve), яка визначає площу під кривою помилок ROC (Receiver Operating Characteristic). Крива ROC будується у декартовій системі координат. Вісь абсцис визначає значення метрики FPR, а вісь ординат – значення метрики TPR, які обчислюються на основі елементів матриці помилок для значень порогу класифікації $t \in [0,1]$ з кроком Dt . Поріг t використовується для порівняння з визначеною класифікатором імовірністю прогнозу для визначення належності об'єкта до одного з двох класів. Площу під отриманою кривою можна обчислити за методом трапецій.

Важливою особливістю метрики AUC є те, що вона є усередненим значенням продуктивності моделі для всіх можливих порогів класифікації. AUC оцінює загальну якість класифікаційної моделі без прив'язки до певного порогу, що робить її універсальним показником для порівняння моделей.

Окрім попередньо описаного способу, AUC можна обчислити безпосередньо за значеннями міток актуальних класів та імовірностей прогнозованих класів усієї вибірки вхідних даних:

$$AUC = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} c\left(y_i < y_j\right) c\phi\left(p_i < p_j\right)}{\sum_{i=1}^{n}\sum_{j=1}^{n} c\left(y_i < y_j\right)}, \qquad (6)$$

де $c\left(y_i < y_j\right) = \begin{cases} 0, & y_i \geq y_j \\ 1, & y_i < y_j \end{cases}$; $c\phi\left(p_i < p_j\right) = \begin{cases} 0, & p_i > p_j \\ 0.5, & p_i = p_j \\ 1, & p_i < p_j \end{cases}$; y_i – актуальна мітка класу; p_i – прогнозована ймовірність $p_i = s\left(W^{*T} X_i\right)$ належності об'єкта до позитивного класу; $s\left(W^{*T} X_i\right)$ – сигмоїдна функція; W^* – вектор навчених ваг; X_i – вектор числових ознак i-го об'єкта; n – обсяг вхідної вибірки.

Як видно з (6), метрика AUC визначає відносну кількість пар об'єктів, які правильно впорядковані за спаданням значень спрогнозованої ймовірності: об'єкт класу y_i розміщено у впорядкованому списку раніше об'єкта y_j.

AUC вимірює здатність моделі ранжувати класи. Вона показує ймовірність того, що випадковий позитивний приклад (із класу 1) буде оцінений вищим балом (імовірністю), ніж випадковий негативний приклад (з класу 0) незалежно від конкретного порогу.

Виходячи з (6), метрика AUC сильно залежить від порядку слідування класів у впорядкованій за спаданням спрогнозованих імовірностей вхідній вибірці даних та від співвідношення цих імовірностей. Так, якщо спочатку впорядкованої вибірки будуть ознаки позитивного класу 1, то значення AUC буде наближатися до 1. Якщо ж спочатку будуть ознаки негативного класу 0, то AUC прямуватиме до 0. Випадкове розміщення класів 0 та 1 означатиме сліпе вгадування і AUC прийме значення приблизно рівне 0.5.

Перекриття класів – це ситуація в задачах класифікації, коли спостереження з різних класів мають подібні або однакові значення ознак (атрибутів), через що модель не може чітко розрізнити до якого класу належить кожен об'єкт. Це ускладнює процес навчання та знижує якість класифікації.

Перекриття класів означає те, що об'єкти вхідної вибірки можуть бути віднесені класифікатором як до класу 0 так і до класу 1. У нашій моделі на ступінь перекриття класів впливатиме відстань між математичними сподіваннями Dm двох функцій густини нормального розподілу випадкових величин та дисперсія d, які використовуються для генерування вхідної послідовності даних – ознак об'єктів класифікаційного експерименту.

Залежність метрики AUC від Dm показана графічно на рис. 5 для значень параметрів $q = 0.5$, $d = 25$. Значення $Dm = |m_1 - m_0|$ обчислюється зміщенням m_1 щодо постійного значення $m_0 = -1$.

Результати отримано для випадкової вхідної вибірки даних без масштабування. Графіки залежності AUC від досліджуваних параметрів апроксимовано лініями тренду.

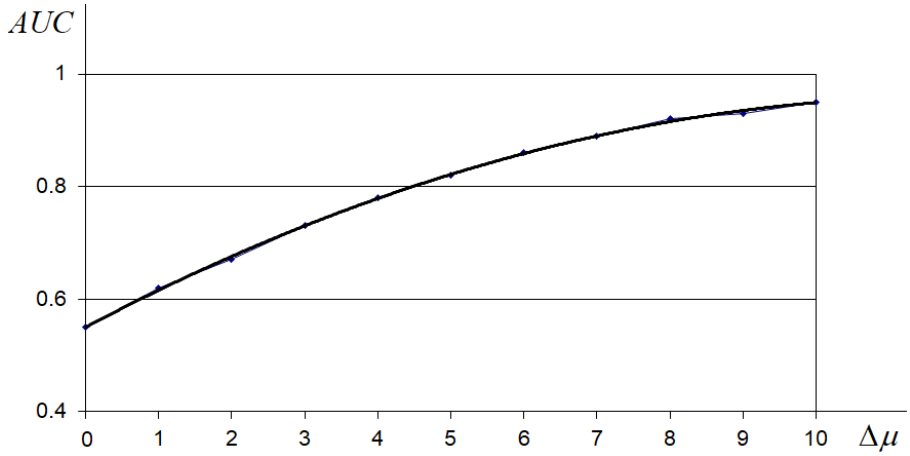


Рис. 5. Залежність метрики AUC від Dm

Збільшення відстані Dm між математичними сподіваннями функцій густини розподілу випадкових величин призводить до зменшення перекриття класів у вхідній вибірці, результатом чого є збільшення значення метрики AUC.

Зі зменшенням відстані Dm перекриття класів збільшується, що призводить до зростання ентропії класифікатора. Відповідно збільшується кількість об'єктів, які класифікатор може віднести як до одного так і до іншого класу. Зі зростанням перекриття класів збільшується кількість неправильно ідентифікованих об'єктів, а значить, будуть збільшуватися значення елементів FP або FN у матриці помилок, що призведе до погіршення загальної точності моделі та може негативно вплинути на залежні від них метрики.

Подібний ефект спостерігається також у разі зростання дисперсії d, яка є параметром нормального розподілу модельних даних. Вплив дисперсії на значення метрики AUC наведено на рис. 6. Результати отримано для $m_0 = -1$, $m_1 = 5$ та $q = 0.5$.

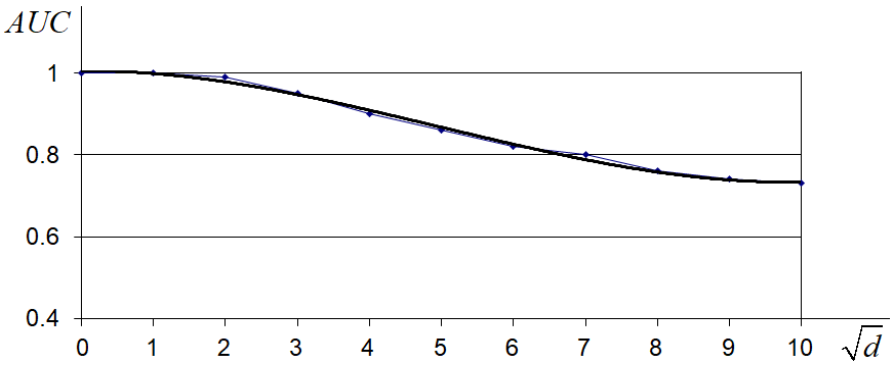


Рис. 6. Залежність метрики AUC від дисперсії d

Графік на рис. 6 демонструє, що метрика AUC зменшується із зростанням дисперсії, яке призводить до більшого перекриття класів. Зменшення дисперсії до 0 призводить до наближення значення AUC до 1, що вказує на найкращий класифікатор.

Для аналізу перекриття класів використовуються спеціалізовані метрики, які допомагають оцінити, наскільки сильно модель помиляється при класифікації даних, що належать до різних класів, але мають схожі характеристики. До таких метрик належать: Precision-Recall Curve, Cross-Entropy Loss, Class Overlap Ratio, Jaccard Index.

Крім цього, для аналізу перекриття класів використовують звичайні метрики: Confusion Matrix, Accuracy, Precision, Recall, F_1, AUC ROC. Так, через невизначеність між класами, пов'язану з їхнім перекриттям, можуть суттєво знижуватися метрики якості моделі, як-от: Accuracy, Precision та Recall. Метрика F_1 буде низькою, якщо моделі важко визначати рідкісний клас через перекриття з поширеним класом. Також, якщо метрика AUC низька, це може свідчити про те, що модель має труднощі з розділенням класів, що є ознакою їх перекриття.

Імовірність θ перемикання між двома функціями густини нормального розподілу з математичними сподіваннями моделює дизбаланс класів. На рис. 7 зображено графік залежності метрики AUC від θ. Параметрами функцій густини нормального розподілу є математичні сподіваннями $m_0 = -1$, $m_1 = 5$ та дисперсія $d = 25$. Одна із функцій нормального розподілу генеруватиме ознаку x_i об'єкта класу $y_i = 0$, а інша – класу $y_i = 1$. Значення ознаки x_i та актуальної мітки класу y_i запам'ятовується у масиві значень.

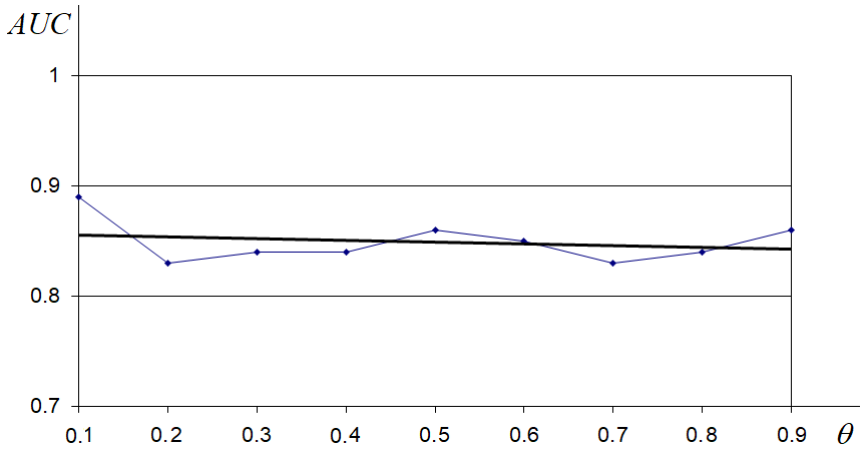


Рис. 7. Залежність AUC від параметра θ

Як видно на рис. 7, в умовах описаного експерименту метрика AUC є відносно стійкою до дизбалансу класів, оскільки вона характеризує здатність моделі розрізняти класи, незалежно від їхньої частки у даних. У зв'язку з цим, метрику AUC можна використати для аналізу даних з дизбалансом класів. Але, оскільки дизбаланс – це кількісне переважання одного класу над іншим у вхідній вибірці даних, то AUC не є найкращою метрикою для дослідження цього явища. AUC не враховує наскільки важливі різні класи в контексті конкретної задачі. У разі сильного дизбалансу метрика AUC може давати оманливі результати, оскільки може бути високою навіть за низької точності для рідкісного класу.

Окрім AUC, стійкою до дизбалансу класів є метрика Balanced Accuracy, яка є середнім арифметичним значенням між чутливістю (Sensitivity або Recall) для рідкісного класу і специфічністю (Specificity) для класу, що трапляється частіше.

Повнішу картину впливу дизбалансу на якість класифікатора на заданому наборі даних дають інші метрики, наприклад, Confusion Matrix, Accuracy, Precision, Sensitivity (Recall), Specificity, F_1. Однак слід підкреслити, якщо модель навчається зі значним дизбалансом класів, то вона може добре розпізнавати домінувальний клас, але буде гірше справлятися з рідкісним класом, що своєю чергою негативно впливає на загальну якість класифікації.

Масштабування ознак об'єктів у вхідних даних забезпечує подібні тренди впливу параметрів θ, d, Dm на метрику AUC. Масштабування ознак може зменшити перекриття класів за рахунок вирівнювання впливу на модель ознак із різними діапазонами значень. Однак масштабування не

вирішує проблеми дизбалансу класів, оскільки вона пов'язана з кількісним співвідношенням класів, а не з масштабом ознак.

Зменшення впливу перекриття класів та покращення результатів класифікації вирішується за допомогою належного аналізу даних (додавання нових інформативних ознак, застосування лінійного дискримінантного аналізу, використання методів кластеризації для кращого розрізнення класів) та застосування складніших моделей і технік (використання нелінійних моделей, штучних нейронних мереж, гібридних моделей опрацювання даних).

Проблема дизбалансу класів може бути ефективно вирішена за допомогою початкової підготовки даних (збільшення кількості прикладів рідкісного класу дублюванням наявних або синтезуванням нових даних, зменшення кількості прикладів домінувального класу видаленням частини даних), використання стійких до дизбалансу метрик і відповідного налаштування моделі (регулювання ваг моделі так, щоб вона підсилювала рідкісний клас у процесі навчання).

Висновки

1. У статті описано розроблену та апробовану комп'ютерну програмну модель логістичної регресії, яка дає змогу досліджувати вплив структури вхідних даних, регульованих параметрів та методів навчання моделі на ефективність бінарної класифікації і враховувати специфічні особливості її предметно-орієнтованого застосування.

2. Побудовано керовану модель вхідних даних на основі механізму імовірнісного перемикання двох генераторів нормально розподілених випадкових величин зі зміщеними математичними сподіваннями, що дало змогу дослідити вплив перекриття та дизбалансу класів на ефективність бінарної класифікації методом логістичної регресії. Оскільки кожен генератор формує ознаки об'єктів тільки одного класу, то на ступінь перекриття класів впливає відстань між математичними сподіваннями та значення дисперсії випадкових значень класифікаційних ознак, а на дизбаланс класів – імовірність перемикання між генераторами цих ознак.

3. Показано, що зменшення відстані між математичними сподіваннями та (або) збільшення дисперсії розподілу випадкових класифікаційних ознак призводять до зростання перекриття класів. Наближення імовірності перемикання генераторів випадкових величин до крайніх меж одиничного інтервалу призводить до зростання дизбалансу класів.

4. Встановлено, що зростання перекриття та (або) дизбалансу класів навчальної вибірки негативно впливає на ефективність роботи класифікатора, погіршуючи здатність моделі правильно розрізняти класи.

5. Виявлення перекриття та (або) дизбалансу класів у вхідному наборі даних забезпечується комплексним аналізом декількох відомих метрик, оскільки кожна з них окремо не дає достатньої інформації про ці явища.

6. Експериментально підтверджено, що швидкість та результативність навчання логістичної регресії значною мірою визначається структурою вибірки вхідних даних та динамікою градієнтного методу.

7. Можливими напрямками вдосконалення розробленої програмної моделі є забезпечення здатності працювати з нечіткими даними, розроблення підсистеми підтримки прийняття рішень для оптимізації показників логіт-класифікатора та інтеграція з іншими моделями машинного навчання.

СПИСОК ЛІТЕРАТУРИ

1. Ewens, W. J. Brumberg, K. (2023). *Introductory Statistics for Data Analysis*. Springer.

2. Friedman, J. (2011). *The Elements of Statistical Learning*. Springer.

3. Hosmer, D. W., Lemeshow, S. (2000). *Applied Logistic Regression*. John Wiley & Sons, Inc. https://doi.org/https://doi.org/10.1002/0471722146.

4. Hilbe, J. M. (2009). *Logistic Regression Models (1st ed.)*. Chapman and Hall/CRC. https://doi.org/https://doi.org/10.1201/9781420075779.

5. Kleinbaum, D. G., Klein, M. (2010). *Logistic Regression: A Self-Learning Text* (3rd ed.). Springer.

6. Harrell, F. E. (2015). *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis* (2nd ed.). Springer.

7. Басюк, Т. М., Литвин, В. В., Захарія, Л. М., Кунанець, Н. Е. (2019). *Машинне навчання:* навч. посібник. Львів: Видавництво "Новий Світ – 2000".

8. Ponniah, P. (2007). *Data Modeling Fundamentals: A Practical Guide for IT Professionals.* Wiley. John Wiley & Sons, LTD.

9. Дубровін, В. І., Дейнега, Л. Ю., Яценко, А. К. (2023). Програмне забезпечення статистичного аналізу. *Електроніка та електротехніка, Автоматизація та комп'ютерно-інтегровані технології, 3*, 25–32. https://doi.org/10.15588/1607-6761-2023-3-3.

10. Blokdyk, G. (2019). *What is Custom Software Development? Your Guide to Building Software That Works for You*. Emereo Pty The Limited. https://multishoring.com/blog/what-is-custom-software-development/.

11. Baruah, R., Ramani, S. S., Chandratrey, K. (2024). *Data Science Toolkit - Logistic regression custom model service.* https://learn.microsoft.com/en-us/xandr/data-science-toolkit/logistic-regression-custom-model-service.

12. *Build vs. buy. A strategic framework for evaluating third-party solutions*. (2022). https://www.thoughtworks.com/content/dam/thoughtworks/documents/e-book/tw_ebook_build_ vs_buy_2022.pdf.

13. *How do you weigh using software development tools versus building your own solutions?* (2024). https://www.linkedin.com/advice/0/how-do-you-weigh-using-software-development.

14. Hackeling, G. (2014). *Mastering Machine Learning With Scikit-learn: Apply Effective Learning Algorithms to Real-world Problems Using Scikit-learn*. Packt Publishing.

15. Adams, S. A. (2020). *An Introduction to Logistic Regression in Python with statsmodels and scikit-learn. Level Up Coding*. https://levelup.gitconnected.com/an-introduction-to-logistic-regression-in-python-with-statsmodels-and-scikit-learn-1a1fb5ce1c13.

16. Wiley, M., Wiley, J. F. (2019). *Advanced R Statistical Programming and Data Models: Analysis, Machine Learning, and Visualization*. APress.

17. Agresti, A., Kateri, M. (2021). *Foundations of Statistics for Data Scientists: With R and Python*. CRC Press.

18. Allison, P. D. (2018). *Logistic Regression Using SAS. Theory and Application, Second Edition*. SAS Institute.

19. Nasser, H. (2020). *Logistic Regression Using SPSS*. https://doi.org/10.13140/RG.2.2.21524.12162. https://www.researchgate.net/publication/344138306_Logistic_Regression_Using_SPSS.

20. George, D., Mallery, P. (2021). *IBM SPSS Statistics. 27 Step by Step: A Simple Guide and REFERENCES*. Taylor & Francis.

21. Geron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition*. O'Reilly Media.

22. Karimpour, A. (2020). *Fundamentals of Data Science with MATLAB: Introduction to Scientific Computing, Data Analysis, and Data Visualization*. Amazon.

23. Lin, M., Chen, J. (2023). Research on Credit Big Data Algorithm Based on Logistic Regression. *Procedia Computer Science, 228*, 511–518. https://doi.org/10.1016/j.procs.2023.11.058. https://www.sciencedirect.com/science/article/pii/S1877050923018823.

24. Tutz, G. (2020). Modelling heterogeneity: on the problem of group comparisons with logistic regression and the potential of the heterogeneous choice model. *Advances in Data Analysis and Classification, 14*, 517–542 (2020). https://doi.org/10.1007/s11634-019-00381-8.

25. Gibbons, L. E., Hosmer, D. W. (1991). Conditional logistic regression with missing data. *Communications in Statistics-Simulation and Computation, 20*(1), 109–120.

26. Bootkrajang, J., Kabán, A. (2012, September). Label-noise robust logistic regression and its applications. *In Joint European conference on machine learning and knowledge discovery in databases*,143–158. Berlin, Heidelberg: Springer Berlin Heidelberg.

27. Sohn, S. Y., Kim, D. H., Yoon, J. H. (2016). Technology credit scoring model with fuzzy logistic regression. *Applied soft computing, 43*, 150–158.

28. Larsen, K., Petersen, J. H., Budtz□Jørgensen, E., Endahl, L. (2000). Interpreting parameters in the logistic regression model with random effects. *Biometrics, 56*(3), 909–914.

29. *Can logistic regression be used for non linear relationships between the independet variables?* (2024). https://typeset.io/questions/can-logistic-regression-be-used-for-non-linear-relationships-426h46ziek.

30. Zhang, L., Geisler, T., Ray, H., Xie, Y. (2021). Improving logistic regression on the imbalanced data by a novel penalized log-likelihood function. *Journal of applied statistics, 49*(13), 3257–3277. https://doi.org/10.1080/02664763.2021.1939662.

31. Jing, Q., Yifei, L. (2019). L 1–2 Regularized Logistic Regression. *53rd Asilomar Conference on Signals, Systems, and Computers*, 779–783. IEEE. https://doi.org/10.1109/IEEECONF44664.2019.9048830.

32. Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, 58*(1), 267–288.

33. Munkhdalai, L., Lee, J. Y., Ryu, K. H. (2020). A Hybrid Credit Scoring Model Using Neural Networks and Logistic Regression. *In Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, 156*. Springer, Singapore. https://doi.org/10.1007/978-981-13-9714-1_27.

34. Мацуга, О. М., Дудукіна, С. О., Григорук, С. П. (2020). Побудова моделі прогнозування результату лікування на прикладі однієї медичної задачі. *Актуальні проблеми автоматизації та інформаційних технологій, 24*, 47–56.

35. Зубченко, В. П., Авраменко, А. В. (2023). Дослідження скорингової моделі для кредитопозичальників банку. *Вісник Київського національного університету імені Тараса Шевченка. Серія: Фізико-математичні науки, 2*, 44–53. https://doi.org/10.17721/1812-5409.2023/2.5.

36. Кравець, П., Твердохліб, Ю. (2023). Інформаційна система моніторингу відгуків у соціальних мережах для формування рекомендацій придбання товарів. *Вісник Національного університету "Львівська політехніка". Серія: Іформаційні системи та мережі, 13*, 218–234. https://doi.org/10.23939/sisn2023.13.218.

37. Rahman, H. A. A., Yap, B. W. (2016). Imbalance Effects on Classification Using Binary Logistic Regression. *In International Conference on Soft Computing in Data Science, SCDS 2016, Communications in Computer and Information Science, Springer, Singapore, 652*, 136–147. https://doi.org/10.1007/978-981-10-2777-2_12.

38. Sun, T., Tang, K., Li, D. (2022). Gradient Descent Learning With Floats. *IEEE Transactions on Cybernetics, 3* (52), 1763–1771. https://doi.org/10.1109/TCYB.2020.2997399.

39. Fehrman, B., Gess, B., Jentzen, A. (2020). Convergence Rates for the Stochastic Gradient Descent Method for Non-Convex Objective Functions. *Journal of Machine Learning Research, 21* (136), 1–48. https://www.jmlr.org/papers/volume21/19-636/19-636.pdf.

40. Кравець, П., Пасічник, В., Проданюк, М. (2024). Математична модель логістичної регресії для бінарної класифікації. Ч. 1. Регресійні моделі узагальнення даних. *Вісник Національного університету "Львівська політехніка". Серія: Іформаційні системи та мережі, 15*, 290–321. https://doi.org/10.23939/sisn2024.15.290.

41. Кравець, П., Пасічник, В., Проданюк, М. (2024). Математична модель логістичної регресії для бінарної класифікації. Ч. 2. Процеси підготовки, навчання і тестування даних. *Вісник Національного університету "Львівська політехніка". Серія: Іформаційні системи та мережі, 15*, 322–340. https://doi.org/10.23939/sisn2024.15.322.

42. Barzilai, J., Borwein, J. M. (1988). Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis, 8*, 141–148. https://doi.org/10.1093/imanum/8.1.141.

43. Wolfe, P. (1969). Convergence Conditions for Ascent Methods. *SIAM Review. 11* (2), 226–235. https://doi.org/10.1137/1011036. JSTOR 2028111.

44. Walton, N. (2019). *Robbins-Monro – Applied Probability Notes.* https://appliedprobability.blog/2019/01/26/robbins-munro-2/.

45. Hossin, M., Sulaiman, M.N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process, 5*(2), 1–11. https://doi.org/10.5121/ijdkp.2015.5201.

REFERENCES

1. Ewens, W. J., Brumberg, K. (2023). *Introductory Statistics for Data Analysis*. Springer.

2. Friedman, J. (2011). *The Elements of Statistical Learning*. Springer.

3. Hosmer, D. W., Lemeshow, S. (2000). *Applied Logistic Regression*. John Wiley & Sons, Inc. https://doi.org/https://doi.org/10.1002/0471722146.

4. Hilbe, J. M. (2009). *Logistic Regression Models (1st ed.)*. Chapman and Hall/CRC. https://doi.org/https://doi.org/10.1201/9781420075779.

5. Kleinbaum, D. G., Klein, M. (2010). *Logistic Regression: A Self-Learning Text* (3rd ed.). Springer.

6. Harrell, F. E. (2015). *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis* (2nd ed.). Springer.

7. Basyuk, T. M., Lytvyn, V. V., Zakharia, L. M., & Kunanets, N. E. (2019). *Machine learning: a study guide (in Ukrainian)*. Lviv: "Novyy Svit – 2000" Publishing House.

8. Ponniah, P. (2007). *Data Modeling Fundamentals: A Practical Guide for IT Professionals*. Wiley. John Wiley & Sons, LTD.

9. Dubrovin, V. I., Deinega, L. Yu., Yatsenko, A. K. (2023). Statistical analysis software (in Ukrainian). *Electronics and electrical engineering, Automation and computer-integrated technologies, 3*, 25–32. https://doi.org/10.15588/1607-6761-2023-3-3.

10. Blokdyk, G. (2019). *What is Custom Software Development? Your Guide to Building Software That Works for You*. Emereo Pty The Limited. https://multishoring.com/blog/what-is-custom-software-development/.

11. Baruah, R., Ramani, S. S., Chandratrey, K. (2024). *Data Science Toolkit - Logistic regression custom model service*. https://learn.microsoft.com/en-us/xandr/data-science-toolkit/logistic-regression-custom-model-service.

12. *Build vs. buy. A strategic framework for evaluating third-party solutions*. (2022). https://www.thoughtworks.com/content/dam/thoughtworks/documents/ebook/tw_ebook_build_vs_buy_2022.pdf.

13. *How do you weigh using software development tools versus building your own solutions?* (2024). https://www.linkedin.com/advice/0/how-do-you-weigh-using-software-development.

14. Hackeling, G. (2014). *Mastering Machine Learning With Scikit-learn: Apply Effective Learning Algorithms to Real-world Problems Using Scikit-learn*. Packt Publishing.

15. Adams, S. A. (2020). *An Introduction to Logistic Regression in Python with statsmodels and scikit-learn. Level Up Coding*. https://levelup.gitconnected.com/an-introduction-to-logistic-regression-in-python-with-statsmodels-and-scikit-learn-1a1fb5ce1c13.

16. Wiley, M., Wiley, J. F. (2019). *Advanced R Statistical Programming and Data Models: Analysis, Machine Learning, and Visualization*. APress.

17. Agresti, A., Kateri, M. (2021). *Foundations of Statistics for Data Scientists: With R and Python*. CRC Press.

18. Allison, P. D. (2018). *Logistic Regression Using SAS. Theory and Application, Second Edition*. SAS Institute.

19. Nasser, H. (2020). *Logistic Regression Using SPSS*. https://doi.org/10.13140/RG.2.2.21524.12162. https://www.researchgate.net/publication/344138306_Logistic_Regression_Using_SPSS.

20. George, D., Mallery, P. (2021). *IBM SPSS Statistics. 27 Step by Step: A Simple Guide and REFERENCES*. Taylor & Francis.

21. Geron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition*. O'Reilly Media.

22. Karimpour, A. (2020). *Fundamentals of Data Science with MATLAB: Introduction to Scientific Computing, Data Analysis, and Data Visualization*. Amazon.

23. Lin, M., Chen, J. (2023). Research on Credit Big Data Algorithm Based on Logistic Regression. *Procedia Computer Science, 228*, 511–518. https://doi.org/10.1016/j.procs.2023.11.058. https://www.sciencedirect.com/science/article/pii/S1877050923018823.

24. Tutz, G. (2020). Modelling heterogeneity: on the problem of group comparisons with logistic regression and the potential of the heterogeneous choice model. *Advances in Data Analysis and Classification, 14*, 517–542 (2020). https://doi.org/10.1007/s11634-019-00381-8.

25. Gibbons, L. E., Hosmer, D. W. (1991). Conditional logistic regression with missing data. *Communications in Statistics-Simulation and Computation, 20*(1), 109–120.

26. Bootkrajang, J., Kabán, A. (2012, September). Label-noise robust logistic regression and its applications. *In Joint European conference on machine learning and knowledge discovery in databases*, 143–158. Berlin, Heidelberg: Springer Berlin Heidelberg.

27. Sohn, S. Y., Kim, D. H., Yoon, J. H. (2016). Technology credit scoring model with fuzzy logistic regression. *Applied soft computing, 43*, 150–158.

28. Larsen, K., Petersen, J. H., Budtz□Jørgensen, E., Endahl, L. (2000). Interpreting parameters in the logistic regression model with random effects. *Biometrics*, *56*(3), 909–914.

29. *Can logistic regression be used for non linear relationships between the independet variables?* (2024). https://typeset.io/questions/can-logistic-regression-be-used-for-non-linear-relationships-426h46ziek.

30. Zhang, L., Geisler, T., Ray, H., Xie, Y. (2021). Improving logistic regression on the imbalanced data by a novel penalized log-likelihood function. *Journal of applied statistics, 49*(13), 3257–3277. https://doi.org/10.1080/02664763.2021.1939662.

31. Jing, Q., Yifei, L. (2019). L 1-2 Regularized Logistic Regression. *53rd Asilomar Conference on Signals, Systems, and Computers*, 779–783. IEEE. https://doi.org/10.1109/IEEECONF44664.2019.9048830.

32. Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, 58*(1), 267–288.

33. Munkhdalai, L., Lee, J. Y., Ryu, K. H. (2020). A Hybrid Credit Scoring Model Using Neural Networks and Logistic Regression. *In Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, 156*. Springer, Singapore. https://doi.org/10.1007/978-981-13-9714-1_27.

34. Matsuga, O. M., Dudukina, S. O., Hryhoruk, S. P. (2020). Building a model for predicting the outcome of treatment on the example of one medical problem (in Ukrainian). *Actual problems of automation and information technologies, 24*, 47–56.

35. Zubchenko, V. P., Avramenko, A. V. (2023). Study of the scoring model for bank borrowers (in Ukrainian). *Bulletin of Taras Shevchenko Kyiv National University, series: physical and mathematical sciences, 2*, 44–53. https://doi.org/10.17721/1812-5409.2023/2.5.

36. Kravets, P., Tverdokhlib, Yu. (2023). An information system for monitoring reviews in social networks for the formation of recommendations for the purchase of goods (in Ukrainian). *Bulletin of the Lviv Polytechnic National University, series: information systems and networks, 13*, 218–234. https://doi.org/10.23939/sisn2023.13.218.

37. Rahman, H. A. A., Yap, B. W. (2016). Imbalance Effects on Classification Using Binary Logistic Regression. *In International Conference on Soft Computing in Data Science, SCDS 2016, Communications in Computer and Information Science, Springer, Singapore, 652*, 136–147. https://doi.org/10.1007/978-981-10-2777-2_12.

38. Sun, T., Tang, K., Li, D. (2022). Gradient Descent Learning With Floats. *IEEE Transactions on Cybernetics, 3* (52), 1763–1771. https://doi.org/10.1109/TCYB.2020.2997399.

39. Fehrman, B., Gess, B., Jentzen, A. (2020). Convergence Rates for the Stochastic Gradient Descent Method for Non-Convex Objective Functions. *Journal of Machine Learning Research, 21* (136), 1–48. https://www.jmlr.org/papers/volume21/19-636/19-636.pdf.

40. Kravets, P., Pasichnyk, V., Prodanyuk, M. (2024). A mathematical model of logistic regression for binary classification. Part 1. Regression models of data generalization (in Ukrainian). *Bulletin of the Lviv Polytechnic National University, series: information systems and networks, 15*, 290–321. https://doi.org/10.23939/sisn2024.15.290.

41. Kravets, P., Pasichnyk, V., Prodanyuk, M. (2024). Mathematical logistic regression model for binary classification. Part 2. Data preparation, training and testing processes (in Ukrainian). *Bulletin of the Lviv Polytechnic National University, series: information systems and networks, 15*, 322–340. https://doi.org/10.23939/sisn2024.15.322.

42. Barzilai, J., Borwein, J. M. (1988). Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis, 8*, 141–148. https://doi.org/10.1093/imanum/8.1.141.

43. Wolfe, P. (1969). Convergence Conditions for Ascent Methods. *SIAM Review. 11* (2), 226–235. https://doi.org/10.1137/1011036. JSTOR 2028111.

44. Walton, N. (2019). *Robbins-Monro – Applied Probability Notes.* https://appliedprobability.blog/2019/01/26/robbins-munro-2/.

45. Hossin, M., Sulaiman, M. N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process, 5*(2), 1–11. https://doi.org/10.5121/ijdkp.2015.5201.

COMPUTER MODELLING OF LOGISTIC REGRESSION FOR BINARY CLASSIFICATION

Petro Kravets[1], Volodymyr Pasichnyk[2], Mykola Prodaniuk[3], Yaroslav Kis[4]

[1–4] Lviv Polytechnic National University,
Information Systems and Networks Department, Lviv, Ukraine
[1] E-mail: Petro.O.Kravets@lpnu.ua, ORCID: 0000-0001-8569-423X
[2] E-mail: Volodymyr.V.Pasichnyk@lpnu.ua, ORCID: 0000-0002-5231-6395
[3] E-mail: Mykola.M.Prodaniuk@lpnu.ua, ORCID: 0000-0001-9544-3792
[4] E-mail: Yaroslav.P.Kis@lpnu.ua, ORCID: 0000-0003-3421-2725

This article discusses the practical aspects of applying logistic regression for binary data classification. Logistic regression determines the probability of an object belonging to one of two classes. This probability is calculated with the help of a sigmoid function, the argument of which is a linear convolution of the feature vector of the object with the weighting coefficients obtained during the minimization of the logarithmic loss function. Predicted class labels are determined by comparing the calculated probability with a given threshold value.

The logistic regression study was performed using the computer simulation method. For this, a software complex was developed, the work of which reproduces the main stages of logistic regression: preparation of input data, training, testing with determination of quality metrics of binary classification, application of the logistic regression method for data classification in practice.

The paper examines the effect of overlapping and imbalance of classes in the input data set on the efficiency of binary classification. The overlapping of classes is modeled by the formation of input data based on two shifted relative to each other density functions of the normal distribution of random variables. Class imbalance is simulated by the probability of switching between these features.

It is shown that when the distance between the mathematical expectations of the density functions of the normal distribution decreases or when the dispersion of random variables increases, the overlapping of relevant classes increases, which leads to an increase in the number of objects that the classifier can assign to one or another class.

Approaching the probability of switching between the distribution functions of random variables to the extreme values of the unit interval leads to an increase in class imbalance, which is manifested in an increase in the number of elements of the input data set labeled with the label of the same class.

It has been experimentally confirmed that the AUC ROC metric, popular in binary classification problems, is dependent on the degree of class overlap and relatively resistant to class imbalance.

Keywords: computer modeling, logistic regression, binary classification, data analysis, machine learning, class overlap, class imbalance, gradient descent, classification quality metrics.