Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

УДК 004.624
DOI: https://doi.org/10.30837/2522-9818.2025.1.151

О. Кирсанов, С. Кривенко

МОДЕЛЬ *ML* ДЛЯ АНАЛІЗУ ВЛАСТИВОСТЕЙ РЕЧОВИНИ НА ОСНОВІ ЇЇ ФІЗИКО-ХІМІЧНИХ ОСОБЛИВОСТЕЙ

Предмет статті – розширення попередніх результатів бінарної класифікації на багатокласову за допомогою моделі *ML* для аналізу властивостей речовини на основі її фізико-хімічних особливостей. **Мета дослідження** – розроблення нової моделі *ML* і показників для порівняння якості аналізу різних моделей, зокрема для аналізу якості вина за його складом. **Завдання**: підготовка даних, вибір типу моделі, її навчання, налаштування, оцінювання, розгортання та моніторинг. У дослідженні використовується **метод** *AWS SageMaker* для підготовки даних, розроблення моделі, її навчання, налаштування, оцінювання, розгортання та моніторингу, а дані обробляються за допомогою блокнотів *Jupyter* і *Pandas*. **Досягнуті результати**. Аналіз даних передбачає описову статистику, кореляційні матриці та візуалізації, як-от гістограми та діаграми розсіювання, щоб зрозуміти взаємозв'язки та якість даних. Моделі було навчено за допомогою *XGBoost*, дані розподілено на набори для навчання, перевірки, тестування та оцінення за допомогою матриць невідповідності та показників *AUC-ROC*. Матриці невідповідності для двох моделей показали змішані результати, доводячи складність порівняння продуктивності моделі та необхідність подальших досліджень незбалансованих класів. Автоматичне налаштування гіперпараметрів *Amazon SageMaker* було використано для оптимізації продуктивності моделі за допомогою баєсівської оптимізації та регресії процесу Гаусса. У дослідженні застосовувалися показники *ROC-AUC* для оцінювання продуктивності моделі з підходами мікро- та макроусереднення, що показують різні значення *AUC* для двох моделей. Друга модель продемонструвала трохи кращу продуктивність на основі показників *AUC*, але аналіз матриці невідповідності показав потребу в моделях, адаптованих до незбалансованих класів. **Висновки**: у дослідженні успішно розроблено нову модель *ML* з метою багатокласової класифікації, продемонстровано її потенціал для покращення передбачення якості вина та запропоновано майбутні напрями досліджень.

Ключові слова: модель *ML*; хмара; матриця; матриця невідповідності; передбачена якість; фактична якість; *ROC-AUC*; дані; параметри; усереднення.

1. Вступ. Перетворення бізнес-проблеми на *ML*-проблему

Завданням статті є поширення нових результатів [1], що досягли автори в попередніх роботах для бінарної класифікації, на багатокласову класифікацію, а саме дослідження можливостей порівняння впливу змін моделі на якість аналізу. Класифікацію виконує модель *ML*. Для штучного інтелекту (*ML*) це завдання є класичним завданням аналізу та містить два елементи: набір даних і безпосередньо модель. Модель є сукупністю типових елементів, які на наступному рівні абстрагування розглядаються як системи, що потребують розроблення під час робочого процесу *MLOps*, який передбачає типову серію конвеєрних завдань: підготовка даних, розроблення моделі, її навчання та налаштування, оцінювання, розгортання та моніторинг. У межах цього завдання досліджуються метрики, що дають змогу порівнювати різні моделі багатокласової класифікації щодо якості аналізу. У цьому разі розробляється нова модель на основі аналізу проблеми й наявних методів, а саме створюється нова система – конвеєр з використанням *AWS SageMaker* та бібліотеки *Pandas* для підготовки даних, створення, навчання та оцінювання точності, експлуатації моделі.

Бізнес-проблема аналізу властивостей речовини на основі її фізико-хімічних особливостей перетворюється на *ML*-проблему. Цю логіку створення нової системи, а саме моделі *ML*, можна аргументувати за аналогією з дизелем. Історично дизель – це нова система, що містить елементи, застосовані в інших типах двигунів. Але сукупність цих елементів надає системі певні властивості, завдяки чому вона названа прізвищем винахідника. Цю аналогію можна розглядати як спрощене загальне визначення системи (моделі). За цією логікою побудована стаття. Перший розділ присвячений перетворенню бізнес-проблеми аналізу властивостей речовини на основі її фізико-хімічних особливостей на *ML*-проблему, що дає змогу сформулювати мету дослідження. У другому розділі розглянуто матеріали й методи. Для розв'язання проблеми *ML* було отримано дані, необхідні для навчання моделі *ML*. Хоча в *AWS* доступні керовані інструменти для

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 1 (31)

маніпулювання даними, сценарій був записаний у блокнот *Jupyter* для оброблення інформації. У третьому розділі наведено результати досліджень та їх обговорення, оцінено точність різних моделей.

Бізнес-проблема аналізу властивостей речовини на основі її фізико-хімічних особливостей перетворена на *ML*-проблему. Отримано надійну постановку проблеми бізнесу та її важливість на основі набору даних, що містить числову інформацію про склад вина разом з його якістю з огляду на склад напою. Розв'язання *ML*-проблеми вирішує бізнес-проблему передбачення якості, а отже, і ціни.

Аналіз публікацій. Мета дослідження

Набори даних отримано в репозиторії машинного навчання Каліфорнійського університету в Ірвіні [2]. Набір даних містить числову інформацію про склад вина разом з його якістю та використовується для досягнення мети аналізу – відповісти на запитання: чи можна, зважаючи на склад вина, передбачити якість напою, а отже, і ціну. Тобто цей набір даних дає змогу перетворити бізнес-проблему аналізу властивостей речовини на основі її фізико-хімічних особливостей на проблему *ML*-навчання з учителем.

Зазначений набір даних також застосовується для перегляду статистики, роботи з викидами та масштабування числової інформації. Два набори даних пов'язані з червоним і білим варіантами португальського вина *Portuguese Vinho Verde*. Через проблеми конфіденційності та логістики доступні лише фізико-хімічні (вхідні) та сенсорні (вихідні) змінні. Відсутня інформація про тип винограду, марку чи ціну продажу вина [3]. Це дослідження має важливе значення, оскільки правильно структуровані дані підвищують точність моделей, що особливо актуально для застосування в медицині, де правильність передбачень є критичною.

Досягнення технологій, хмарних обчислень та розроблення алгоритмів сприяли зростанню можливостей і використанню навчання машин *ML*. Керовані сервіси, наприклад *AWS SageMaker*, спрощують навчання машин. Вони надають спеціальні інструменти для автоматизації цих процесів, допомагаючи командам оптимізувати свої робочі процеси та застосовувати найкращі практики для створення моделі навчання машин.

Хоча в *AWS* доступні керовані інструменти для маніпулювання даними, сценарій був записаний у блокнот *Jupyter* для оброблення інформації. Сценарій вилучення та завантаження містить три розділи *ETL*: імпорт і змінні (бібліотека *boto*3); завантаження та вилучення (вебзапит та збереження байтів з URL в архів ZIP); завантаження на *Amazon S*3.

Однією з найпопулярніших бібліотек *Python* з відкритим кодом є *pandas* [4]. Вона була розроблена для переформатування інформації з різних форматів, зокрема *CSV, JSON, Excel, Pickle* тощо, у табличне подання даних у рядках і стовпцях. Було також передбачено функції для аналізу та оброблення інформації, що застосовується в цьому модулі. *Pandas* забезпечує можливість завантаження даних у різних форматах, таких як *CSV, JSON, Excel* та *Pickle*. Було реалізовано зручний спосіб завантаження даних.

Під час завантаження даних у *Pandas* вони зберігаються в структурі *DataFrame* (таблиця *Pandas*). *DataFrame* описується як загальна двовимірна таблична структура з мітками, розмір якої можна змінювати, з потенційно неоднорідно типізованими стовпцями [5]. Структуру *DataFrame* було запропоновано у вигляді аналогії з електронною таблицею або таблицею *SQL* з екземплярами (рядками) та атрибутами (стовпцями).

Атрибут *shape DataFrame* було призначено для опису кількості екземплярів (рядків) і атрибутів (стовпців). Кожен стовпець у *DataFrame* є серією – одновимірним масивом із мітками, що здатний зберігати інформацію будь-якого типу.

$$\boldsymbol{df_{wine}.shape} = (1599,12) \tag{1}$$

Разом із даними було завантажено *DataFrame* з індексом (мітками рядків) та стовпцями (мітками стовпців) [6]. За замовчуванням у процесі завантаження інформації із *CSV*-файлу, що містить рядок-заголовок, стовпці створюються на основі першого рядка файлу. Однак можна змінювати цю поведінку.

$$\begin{aligned}&\boldsymbol{df_{wine}.columns = Index}\big(\big[\boldsymbol{'fixed\ acidity'},\\ &\boldsymbol{'volatile\ acidity'}, \boldsymbol{'citric\ acid'},\\ &\boldsymbol{'residual\ sugar'}, \boldsymbol{'chlorides'},\\ &\boldsymbol{'free\ sulfur\ dioxide'}, \boldsymbol{'total\ sulfur\ dioxide'},\\ &\boldsymbol{'density'}, \boldsymbol{'pH'}, \boldsymbol{'sulphates'}, \boldsymbol{'alcohol'},\\ &\boldsymbol{'quality'}\big], \boldsymbol{dtype = 'object'}\big)\end{aligned} \tag{2}$$

$$\begin{aligned}&\boldsymbol{df_{wine}.index} =\\ &= \boldsymbol{RangeIndex}\big(\boldsymbol{start} = 0, \boldsymbol{stop} = 1599, \boldsymbol{step} = 1\big)\end{aligned} \tag{3}$$

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31) ISSN 2524-2296 (online)

У разі, якщо у вихідному файлі відсутні назви стовпців, їх можна передати як параметр. Якщо мітки осей не вказані, стовпці автоматично впорядковуються відповідно до вставки (для версій *Python* від 3.6 та версій *Pandas* від 0.23). У попередніх версіях упорядкування стовпців відбувалося за лексичним порядком ключів словника.

Для аналізу даних було взято до уваги необхідність використання правильних типів даних [7]. Здебільшого типи даних було визначено автоматично бібліотекою *Pandas* під час завантаження, що дало змогу продовжити роботу без додаткових змін. Однак за наявності знань у предметній галузі або доступу до експерта було проведено додаткову перевірку для виявлення можливих проблем з типами даних.

Для отримання інформації про типи даних у стовпцях було використано або атрибут *dtypes*, або метод *info()*. Якщо типи даних виявлялися некоректними, рекомендовано з'ясувати причину. Це часто пов'язано з відсутніми значеннями в числових стовпцях або наявністю одиничного текстового значення.

Після аналізу інформації виконувалося перетворення типу даних за допомогою методу *python astype()*.

Метою роботи є розроблення нової моделі *ML* та її метрик, що поширюють досягнення попередньої моделі бінарної класифікації на модель багатокласової класифікації та дають змогу порівнювати різні моделі щодо якості аналізу.

2. Матеріали й методи

Після надання даним зручного для читання формату було запропоновано застосувати описову статистику для глибшого їх розуміння [8]. Описова статистика забезпечує цінну інформацію, що дає змогу ефективніше обробляти дані та готувати їх для використання в моделях машинного навчання. Далі розглянуто застосування описової статистики та її значущість.

Було залучено загальну статистику, що містить кількість рядків (екземплярів) і кількість стовпців (ознак або атрибутів) у наборі даних. Ця інформація відтворює розмірність даних, що є важливим аспектом, оскільки надмірна кількість ознак може призвести до високої розмірності, негативно впливаючи на продуктивність моделі.

Статистика атрибутів була зосереджена на числових атрибутах, що дало змогу краще уявити форму розподілу даних, зокрема середнє значення, стандартне відхилення, дисперсію, мінімальні та максимальні значення.

Мультиваріативна статистика застосована для аналізу взаємозв'язків між кількома змінними, зокрема для визначення кореляцій та зв'язку між атрибутами.

Особливу увагу було приділено кореляції між атрибутами, оскільки висока кореляція між двома атрибутами може іноді погіршувати продуктивність моделі. Виявлення тісно пов'язаних ознак, що використовуються разом в одній моделі для аналізу вихідної змінної, сприяє уникненню можливих проблем, зокрема складнощів із досягненням мінімізації втрат моделі.

Середнє значення (*mean*) та медіану (*median*) розглянуто як дві різні міри центральної тенденції, що описують зосередженість інформації навколо певного значення або позиції. Середнє значення виявилося корисним для аналізу даних із симетричним розподілом. Однак у разі асиметричного розподілу або наявності вибірок із викидами (*outliers*) рекомендовано використовувати медіану як більш стійку до таких викидів міру. Наприклад, у разі присутності великих значень-викидів, які можуть зміщувати середнє, медіана забезпечує більш точне подання реального центру даних. Подальший аналіз викидів було заплановано для глибшого розуміння їх впливу на інформацію [9].

Для отримання статистичних показників для числових даних було використано метод *describe()*, що надає загальну інформацію про розподіл значень у стовпцях. Також можна переглядати статистику для окремих стовпців за допомогою методів *mean(), median(), min(), max(), skew()* та *describe()*.

Отже, розглянуто різні способи обчислення та аналізу статистичних показників для ефективнішого розуміння структури даних.

Для візуалізації числових ознак було використано гістограми як ефективний інструмент для оцінювання загальної поведінки даних. За допомогою гістограм проаналізовано такі аспекти: нормальність розподілу даних, кількість піків та наявність асиметрії.

Гістограми групують значення ознак у певні інтервали (*bins*), що дало змогу оцінити частотність значень у кожному інтервалі. Вищі піки на гістограмі вказували на найбільш поширені значення.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 1 (31)

Ця техніка групування (*binning*) також була визначена як корисна в контексті інженерії ознак.

Крім гістограм, для числових ознак було використано графіки щільності (*density plots*) та ящикові діаграми (*box plots*). Ці інструменти допомогли оцінити діапазон даних, визначати пік розподілу, ідентифікувати викиди та виявити особливості ознаки. Наприклад, графіки щільності показували розподіл даних у вигляді безперервної кривої, тоді як ящикові діаграми забезпечували подання даних за квартилями.

У ящиковій діаграмі [10] було зображено діапазон значень між $Q1$ (першим квартилем) і $Q3$ (третім квартилем), із медіаною $(Q2)$, яка позначалася лінією всередині ящика. Вуса (*whiskers*) простягалися за межі ящика на відстань, що не перевищувала $1,5 * IQR$ (діапазону між квартилями, $IQR = Q3 - Q1$). Дані, що виходили за межі вусів, позначалися як викиди (*outliers*).

Такі візуалізації сприяли глибшому розумінню структури даних, а також допомогли визначити, чи потрібне додаткове оброблення для корекції особливостей або викидів.

Для аналізу взаємозв'язків між числовими змінними використано діаграми розсіювання. Цей метод допоміг візуалізувати зв'язки між змінними, навіть якщо кореляція між ними була низькою через розсіяність даних. Наприклад, для змінних "сульфати" та "алкоголь" діаграма розсіювання дала змогу ідентифікувати потенційні позитивні зв'язки, що не були очевидними під час розгляду статистичних показників.

Було також розглянуто можливість аналізу кількох змінних одночасно за допомогою матриць діаграм розсіювання [11]. У бібліотеці *Pandas* реалізовано простий спосіб створення таких матриць, що дає змогу оцінити парні зв'язки між будь-якими двома змінними з обраного набору стовпців. Наприклад, за наявності трьох стовпців можна отримати всі припустимі парні діаграми розсіювання.

Цей підхід забезпечив ефективний інструмент для візуального аналізу числових ознак, що сприяло кращому розумінню їх взаємозв'язків і потенційного впливу на подальше моделювання.

З допомогою діаграми розсіювання [12] було виявлено особливі регіони, до яких належить певна підмножина даних. Визначено зв'язок між алкоголем, сульфатами та якістю. Ці значення для хорошого (якість > 5) та поганого (якість ≤ 5) вина зображені на діаграмі.

Побудова діаграм дала змогу уявити, наскільки корисними є певні змінні для задачі класифікації.

Було визначено, як можна кількісно оцінити лінійний зв'язок між змінними, розміщеними на діаграмі розсіювання. Кореляційна матриця [13] є корисним інструментом для визначення як сильного, так і слабкого лінійного зв'язку між числовими змінними.

Розглянуто використання теплових карт для візуалізації даних, що значно полегшує аналіз числових показників. Найвищий показник (1) позначено темно-зеленим кольором, а найнижчий (–1) – темно-коричневим. Колір відтворює як напрямок (позитивний чи негативний), так і силу кореляції.

Основною метою теплової карти (рис. 1) є виявлення кореляцій за кольором, навіть якщо текст на діаграмі не зчитається. Для створення такої візуалізації використано функцію теплової карти з бібліотеки *Seaborn*.

На графіку виявлено певну кореляцію між лимонною кислотою та фіксованою кислотністю. Два найбільші коефіцієнти лінійного зв'язку для якості вина демонструють кореляцію з алкоголем та сульфатами, вони мають коефіцієнти кореляції, відповідно, 0.48 та 0.25 в нижньому рядку теплової карти. На тепловій карті має місце деяка кореляція 0.67 між лимонною кислотою та фіксованою кислотністю, незначна кореляція –0.68 між фіксованою кислотністю та pH. Такий зв'язок пояснюється тим, що лимонна кислота впливає на загальну кислотність вина. Водночас між фіксованою кислотністю та pH визначено низьку кореляцію. Це пояснюється різницею між показниками: pH вимірює силу кислот, тоді як фіксована кислотність – їх кількість. У наборі даних явної залежності між цими двома властивостями не виявлено.

Розглянуто необхідність очищення даних через можливу наявність викидів. Викиди – це точки, розташовані на аномально великій відстані від інших значень у наборі даних. Вилучення викидів не завжди є обов'язковим, оскільки вони можуть додати цінності набору даних. Проте викиди здатні ускладнювати отримання точних передбачень через зміщення значень щодо інших, більш нормальних даних. Також зазначено, що викиди можуть свідчити про те, що точка даних помилково належить до іншого стовпця.

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

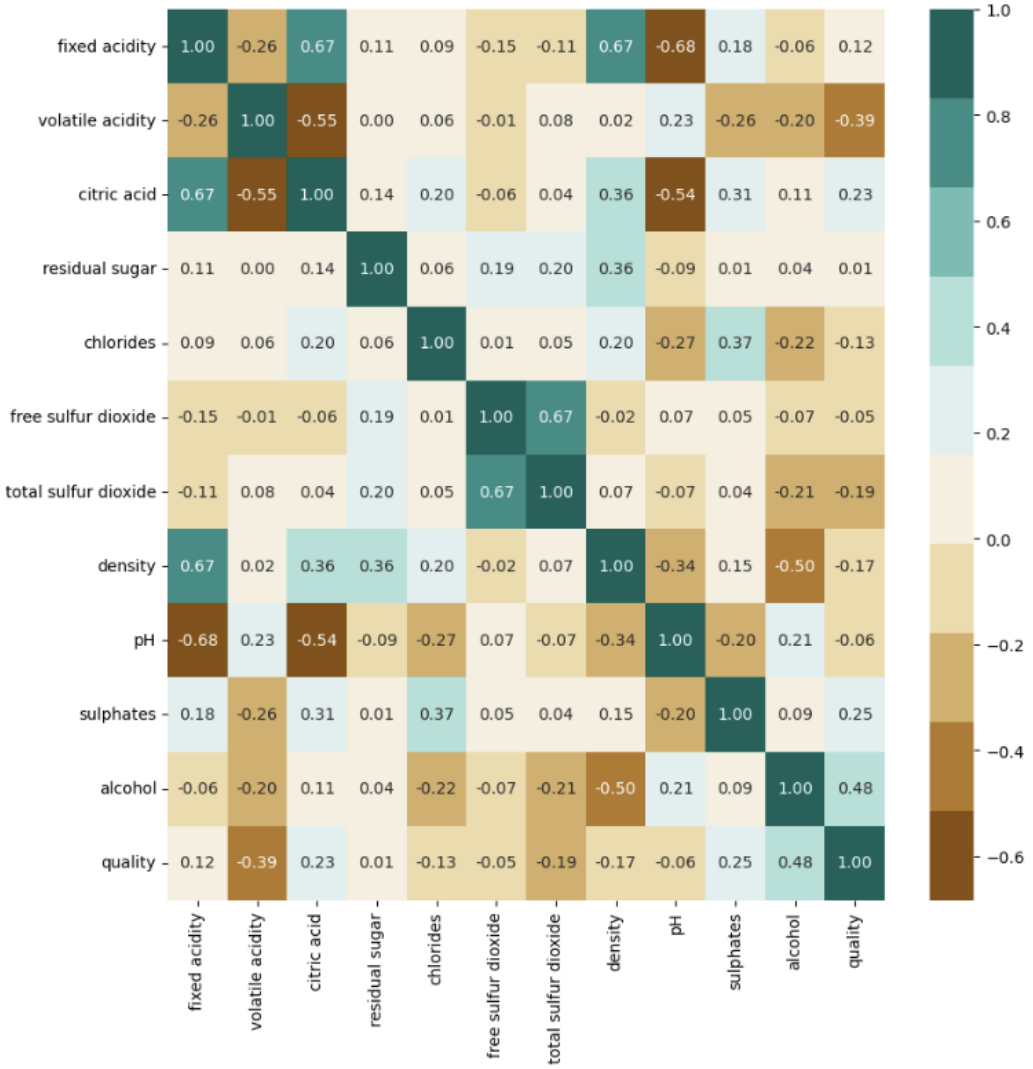

Рис. 1. Теплова карта кореляційної матриці

Запропоновано поділити викиди на дві категорії: уніваріантні (стосуються однієї змінної) та мультиваріантні (стосуються двох або більше змінних).

Одним із найбільш поширених способів знайти уніваріантні викиди є використання коробчастої діаграми (*box plot*), яка показує, наскільки далеко точка даних розташована від середнього значення для цієї змінної. Коробка на діаграмі показує значення даних у межах двох квартилів від середнього значення.

Точкова діаграма (*scatter plot*) може бути ефективним способом виявлення мультиваріантних викидів. Наприклад, у цій діаграмі показано кількість сульфатів і алкоголю в колекції вин. За допомогою точкової діаграми можна швидко побачити, чи існують мультиваріантні викиди для двох змінних.

Бізнес-проблема аналізу властивостей речовини на основі її фізико-хімічних особливостей перетворена на *ML*-проблему. Отримано надійну постановку проблеми бізнесу та її важливість на основі набору даних, що містить числову інформацію про склад вина разом з його якістю з огляду на склад напою. Розв'язання *ML*-проблеми вирішує бізнес-проблему передбачення якості, а отже, і ціни. Більшість бізнес-проблем підпадають під одну з двох категорій: класифікація чи регресія. Зазначена *ML*-проблема належить до категорії багатозначної класифікації. Ціль належить шістьом класам 0–5.

Проведено очищення та підготовку даних, однак вони все ще можуть не відповідати вимогам для навчання алгоритму. Деякі алгоритми потребують даних в особливому форматі. Визначено кроки форматування для підготовки до навчання моделі.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 1 (31)

Деякі алгоритми не працюють із даними у форматі *DataFrame*. Дані для навчання *ML* зберігаються в різних форматах залежно від обраного алгоритму. Наприклад, багато алгоритмів *Amazon SageMaker* підтримують формат файлів *CSV* для навчання [14]. Файли *CSV*, які використовуються з *Amazon SageMaker*, не можуть містити заголовків, а цільова змінна має бути в першому стовпці.

Досліджено, що більшість алгоритмів *Amazon SageMaker* працюють найефективніше з оптимізованим форматом *protobuf recordIO*. У цьому форматі кожен екземпляр у наборі даних перетворюється на бінарне подання як набір 4-байтних чисел із рухомою точкою, після чого завантажується в поле значень *protobuf*.

Визначено, що цей формат дає змогу застосовувати режим *Pipe* для алгоритмів, що його підтримують. У режимі *Pipe* дані передаються безпосередньо з *Amazon S*3, тоді як у режимі *File* вся інформація завантажується з *Amazon S*3 на томи екземплярів навчання. Потокова передача в режимі *Pipe* забезпечує швидший запуск навчальних завдань і кращу пропускну здатність, а також допомагає зменшити розмір томів *Amazon Elastic Block Store (Amazon EBS)*, оскільки потрібно лише зберігати кінцеві артефакти моделі. Водночас режим *File* потребує більше дискового простору для зберігання як кінцевих артефактів, так і повного набору даних.

Цільова змінна в наборі навчальних даних має розташовуватися в першому стовпці ліворуч, а ознаки – у стовпцях праворуч. У використанні формату *CSV* важливо, щоб перший стовпець був цільовою змінною.

Тому цільова змінна в навчальному наборі даних була переміщена в перший стовпець ліворуч, у цьому разі ознаки розміщені праворуч від стовпця цільової змінної.

Оцінка моделі на тих самих даних, на яких вона навчалася, спричиняє перенавчання. Перенавчання виникає, коли модель надто добре запам’ятовує особливості навчального набору даних замість того, щоб вивчати зв’язки між ознаками та мітками. Як наслідок, модель не здатна застосовувати ці зв’язки та шаблони до нових даних у майбутньому.

Реалізовано метод утримання, що передбачає розподіл даних на кілька наборів: навчальні, валідаційні та тестові [15].

Навчальні дані, що містять ознаки та мітки, було подано на вхід обраному алгоритму для створення моделі. Потім модель використано для аналізу на валідаційному наборі даних. На етапі роботи з валідаційним набором виявлено аспекти, що потребують коригування, налаштування чи вдосконалення. Після внесення змін модель було протестовано на тестовому наборі даних, що передбачає лише ознаки, оскільки мітки мають бути передбачені.

Застосовано такі рекомендації щодо розподілу даних: 80 % – для навчання, 10 % – для валідації та 10 % – для тестування. Для великих обсягів даних альтернативний розподіл: 70 % – на навчання, 15 % – на валідацію та 15 % – на тестування.

Виявлено, що дані, розташовані в певному порядку, можуть призводити до упереджень у моделях, і це особливо актуально, коли використовуються структуровані дані. Зазначено, що дані можуть бути упорядковані, наприклад, за стовпцем якості. У разі, якщо запущено модель на тестових даних, цей упорядкований патерн застосовується, що створює упередження для моделі. Також деякі цілі можуть бути відсутні в навчальних даних.

Як правило, рекомендовано рандомізувати набір даних перед розподілом, і багато бібліотек надають функції для рандомізації [16]. Для менших наборів застосовується стратифікований вибірковий відбір. Він гарантує, що в навчальному й тестовому наборах збережено приблизно однаковий відсоток зразків кожного класу цілей, як і в повному наборі.

Функція *train_test_split з sklearn* дає змогу найпростішим чином виявити способи перемішування та розподілу даних [17].

Було застосовано *stratify*, щоб гарантувати збереження співвідношення у стовпці цілей під час розподілу даних.

Amazon SageMaker надає готові контейнери для підтримки фреймворків глибокого навчання, таких як *Apache MXNet, TensorFlow, PyTorch* і *Chainer*. Він також підтримує бібліотеки *ML*, зокрема *scikit-learn* і *SparkML*, надаючи попередньо зібрані образи *Docker*. Було використано *Amazon SageMaker Python SDK*, вони розгортаються за допомогою відповідного класу *Amazon SageMaker SDK Estimator*. Створено код *Python*, що реалізує алгоритм, і налаштовано попередньо зібране зображення для доступу до коду як точки входу.

XGBoost – це популярна та ефективна реалізація з відкритим кодом алгоритму градієнтного бустингу

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31) ISSN 2524-2296 (online)

дерев. Градієнтний бустинг – це алгоритм контрольованого навчання, який намагається точно передбачити цільову змінну. Він досягає свого передбачення внаслідок комбінування ансамблю оцінок з набору більш простих, слабших моделей. *XGBoost* добре зарекомендував себе на змаганнях з машинного навчання. Він стійко обробляє різні типи даних, відношення та розподіли, а також чимало гіперпараметрів, що можна ефективно налаштувати та підігнати. Ця гнучкість робить *XGBoost* хорошим вибором для проблем регресії, багато класової класифікації, тобто аналізу властивостей речовини за допомогою інтелектуального аналізу даних на основі її фізико-хімічних особливостей способом хмарної інтеграції

Щоб навчити модель в *Amazon SageMaker*, створено навчальне завдання, що містить таку інформацію: URL-адресу сегмента *Amazon S*3, де збережено навчальні дані; обчислювальні ресурси, які *Amazon SageMaker* має використовувати для навчання моделі. Обчислювальні ресурси – це екземпляри обчислень *ML*, якими керує *Amazon SageMaker*. *Amazon SageMaker* надає вибір типів екземплярів, оптимізованих для різних випадків використання машинного навчання. Типи екземплярів передбачають різні комбінації центрального процесора, оперативної пам'яті, сховища та мережевої ємності. Було обрано відповідне поєднання ресурсів для створення, навчання та розгортання моделей *ML*. Кожен тип екземпляра має один або кілька розмірів екземпляра, і ця функція дає змогу масштабувати ресурси відповідно до вимог цільового робочого навантаження.

Завантаження в *Amazon S*3 відбувається після розпакування файлів у папку, виконано перебір файлів папки та завантажено кожен файл у *Amazon S*3.

Після того, як модель була навчена, налаштована та протестована, вона готова до розгортання. У наступному розділі налаштування моделі змінюються. Може скластися враження, що фази конвеєра *ML* не в порядку. Щоб перевірити та отримати показники продуктивності запропонованої моделі, були зроблені висновки або передбачення на основі моделі, що зазвичай вимагає розгортання. Розгортання для тестування відрізняється від виробництва, хоча механіка однакова. *Amazon SageMaker* надає все необхідне для розміщення моделі для простого тестування та оцінювання. Оцінювання може передбачати будь-що: від кількох запитів до розгортань, що обробляють десятки тисяч запитів.

Модель можна розгорнути двома способами. Для окремих передбачень модель розгорнуто за допомогою послуг хостингу *Amazon SageMaker*. Він розгортає кілька обчислювальних екземплярів, які запускають модель за кінцевою точкою із збалансованим навантаженням. Програми можуть викликати *API* у кінцевій точці, щоб робити передбачення. За допомогою цієї моделі можна збільшити або зменшити кількість екземплярів залежно від попиту. Щоб отримати передбачення для всього набору даних, було застосовано пакетне перетворення *Amazon SageMaker*. Замість розгортання та підтримки постійної кінцевої точки *Amazon SageMaker* запускає модель і виконує передбачення для всього наданого набору даних. Він зберігає результати в *Amazon S*3 перед тим, як вимкнеться та припинить роботу інстанцій. Виконання пакетних передбачень під час тестування моделі є корисним, оскільки можна швидко запустити весь набір перевірки на моделі. Не потрібно писати жодного коду для оброблення та зіставлення окремих результатів.

Метою етапу розгортання є створення керованого середовища для розміщення моделей для забезпечення безпечного виходу з низькою затримкою. Після розгортання у виробництві здійснювався контроль виробничих даних. Модель перенавчалась, оскільки нові розгорнуті моделі мають відтворювати поточні виробничі дані. Нові дані накопичуються з часом, і вони потенційно можуть визначити альтернативні або нові результати. Тому розгортання моделі є безперервним процесом, а не одноразовою вправою.

Для високої доступності та резервування було перевірено розгортання моделі для автоматичного масштабування екземплярів *Amazon ML* у кількох зонах доступності. Було вказано тип екземпляра, а також максимальну та мінімальну кількість, *Amazon SageMaker* подбав про інше. Він запускає екземпляри, розгортає модель і налаштовує безпечну кінцеву точку *HTTPS* для програми. Програма містила виклик *API* до цієї кінцевої точки, щоб досягти низької затримки та високої пропускної здатності. Ця архітектура дає змогу інтегрувати нові моделі в програму за лічені хвилини, оскільки зміни моделі більше не потребують змін коду програми. *Amazon SageMaker* керує виробничою обчислювальною інфраструктурою від імені клієнта. Він перевіряє працездатність, застосовує виправлення безпеки та проводить інше планове обслуговування. *Amazon SageMaker* виконує всі ці завдання за допомогою

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 1 (31)

вбудованого моніторингу та журналювання *Amazon CloudWatch.*

Після того, як навчено модель, створена кінцева точка в коді за допомогою консолі *Amazon SageMakerSDK*. Оскільки передбачено розміщення двох моделей, була створена багатомодельна кінцева точка.

Багатомодельні кінцеві точки забезпечують масштабоване та економічно ефективне рішення для розгортання значної кількості моделей. Використовується спільний контейнер обслуговування, що дає змогу розміщувати кілька моделей. Цей підхід знижує витрати на розміщення внаслідок підвищення використання кінцевих точок порівняно з одномодельними кінцевими точками. Також оптимізовано витрати на розгортання завдяки тому, що *Amazon SageMaker* завантажує моделі в пам'ять і масштабує їх відповідно до шаблонів трафіку.

Коли розгортаються моделі навчання машин у виробництві, для передбачень нових даних узгодженість важлива. Було перевірено, що ті самі кроки оброблення даних, які використовувалися під час навчання, також застосовувались до кожного запиту на вихід. Інакше можна отримати неправильні результати передбачення. Використовуючи конвеєри виходу, можна повторно впроваджувати етапи оброблення даних, які застосовувалися під час навчання моделі для логічного виходу. Немає необхідності підтримувати дві окремі копії одного коду. Таке повторне використання коду сприяє точності передбачень і знижує витрати на розроблення. Оскільки *Amazon SageMaker* є керованою службою, конвеєрами виходів можна повністю керувати. Коли розгортається модель конвеєра, служба встановлює та запускає послідовність контейнерів у кожному екземплярі *EC*2 у кінцевій точці або в завданні пакетного перетворення. Крім того, послідовність оброблення функцій і виходів виконується з низькою затримкою, оскільки контейнери розташовані в тих самих екземплярах *EC*2.

3. Результати досліджень та їх обговорення. Оцінювання точності моделі

Після того, як було навчено й налаштовано дві моделі, щоб прийняти рішення щодо того, яка з них найкраще підходить для бізнес-проблеми, вони обидві були оцінені. Було визначено, чи будуть моделі ефективними в передбаченні мети на нових і майбутніх даних. Оскільки майбутні екземпляри мають невідомі цільові значення, було оцінено, як модель працює на даних, для яких невідома цільова відповідь. Потім ця оцінка використовувалась як проксі для продуктивності майбутніх даних. З цієї причини було надано зразок даних для оцінювання чи тестування.

Важливою частиною цього етапу є вибір показника, що найбільше підходить для бізнес-ситуації, яка досліджується. Під час цього етапу на основі визначення бізнес-проблеми та її результату розроблена бізнес-метрика для оцінювання успіху. Показник моделі, обраний на цьому етапі, тісно пов'язаний із бізнес-метрикою. Ці два показники мають високу кореляцію. Крім розгляду бізнес-проблеми та показника успіху, тип проблеми *ML,* яка досліджувалась, впливає на метрику обраної моделі. У цьому розділі наведені загальні показники, що застосовувалися в задачах класифікації. Бізнес-проблема визначена як проблема багатокласової класифікації.

У випадку багатокласової класифікації ключові поняття *TPR (true positive rate)* або *FPR (false positive rate)* отримуються лише після бінарного виходу. Це можна зробити двома способами: схема *OvR (One-v-Rest)* порівнює кожен клас з усіма іншими (які вважаються одним); схема "один проти одного" *OvO* порівнює кожну унікальну попарну комбінацію класів. У дослідженні застосовано схему *OvR*.

Імпортовано набір даних про вино, що містить шість класів, кожен з яких відповідає якості напою $(0, 1, ..., 5)$. Один клас лінійно відділяється від інших п'яти; останні не є лінійно віддільними один від одного. Тут ми бінаризували вихід. Були навчені дві моделі *XGBoost*, що можуть природним чином обробляти багатокласові проблеми.

Ключові поняття *TPR* або *FPR* на основі спрощеної моделі бінарної класифікації вина, яка позначає дані як "гарне" або "негарне". Після того, як модель навчена, можна використовувати тестовий набір даних, який був прихований, з метою виконання передбачення. Щоб допомогти перевірити продуктивність моделі, можна порівняти передбачені значення з фактичними. Крім того, можна подати значення в таблиці, щоб уявити, наскільки добре працює модель.

У матриці невідповідності можна отримати порівняння на високому рівні того, як передбачені класи збігаються з фактичними. Припустимо, що

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31) ISSN 2524-2296 (online)

фактичною міткою або класом є "гарне", що ідентифікується як *P* для позитивного значення в матриці невідповідності. І припустимо, що передбачена мітка або клас також є "гарне". Це означає, що отримано справді позитивний результат *TP (true positive)*, який є хорошим для моделі. Так само припустимо, що є фактична мітка "негарне", яка ідентифікується як *N* для негативного значення в матриці невідповідності. І припустимо, що передбачена мітка або клас також не є "негарне". Тоді маємо справжній негативний результат *TN (true negative)*, що також є хорошим для моделі. В обох цих випадках модель, використовуючи дані тестування, передбачила правильний результат. Можливі два інші результати, обидва з яких не ідеальні. Перший – коли фактичний клас негативний, тобто "негарне", але передбачений клас позитивний, отже, "гарне". Цей результат називається хибно позитивним *FP (false positive)*, оскільки передбачення є позитивним, але неправильним. Помилково негативний результат *FN (false negative)* виникає, коли фактичний клас позитивний, тобто "гарне", але передбачений клас негативний, отже, "негарне".

Аналіз матриці невідповідності сформованих на основі роботи двох різних моделей на тих самих даних не дає змогу визначити, яка модель краща. Чи доцільніше переконатися, що модель – всі гарні проби вина, навіть якщо це означає багато помилкових спрацьовувань? Або краще переконатися, що модель найточніша? Це важко зрозуміти, зважаючи на дві матриці невідповідності. Навіть важко уявити, що буде, якщо виникне необхідність порівняти багато моделей.

З метою порівняння досліджуваних моделей визначено додаткові показники. Першим показником є чутливість *TPR*.

$$\boldsymbol{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}. \quad (4)$$

Цей показник є часткою позитивних ідентифікацій. У прикладі про вино це означає, який відсоток напою було правильно ідентифіковано. Для обчислення чутливості кількість справжніх позитивних результатів (позитивних ідентифікацій проб вина) поділено на загальну кількість фактичних проб.

Специфічність *FPR*, яку іноді називають селективністю, – це частка правильно визначених негативів.

$$\boldsymbol{FPR} = \frac{\boldsymbol{TN}}{\boldsymbol{TN} + \boldsymbol{FP}}. \quad (5)$$

У нашому прикладі про вино кількість неякісних проб визначена як "негарне". Для обчислення специфічності кількість справжніх негативів розділена на загальну кількість фактичних негативів. У цьому прикладі це значення – кількість правильно визначених проб неякісного вина, поділена на загальну кількість фактичних проб неякісного вина.

Щоб допомогти перевірити продуктивність моделі, порівняно передбачені значення з фактичними значеннями (табл. 1), що дає деяку уяву про ефективність роботи моделі [18].

Таблиця 1. *Матриця невідповідності першої моделі*

Фактична якість	**Кількість передбачень для якості**					
	0	**1**	**2**	**3**	**4**	**5**
0	0	0	0	1	0	0
1	0	0	2	3	0	0
2	0	1	43	23	1	0
3	0	0	17	44	3	0
4	0	0	2	5	13	0
5	0	0	0	1	1	0

Для 2-го класу якості напою отримано 43 правильних передбачення з 68 проб вина 2-го класу якості.

Для 3-го класу якості напою отримано 44 правильних передбачення з 64 проб вина 3-го класу якості.

Процес налаштування гіперпараметрів для першої моделі був трудомісткий і виконаний штатними автоматичними засобами *Amazon SageMaker (tuning job)*. У табл. 2 подано інформацію щодо ефективності роботи другої моделі.

Таблиця 2. *Матриця невідповідності другої моделі*

Фактична якість	**Кількість передбачень для якості**					
	0	**1**	**2**	**3**	**4**	**5**
0	0	0	0	1	0	0
1	0	0	3	2	0	0
2	0	1	49	18	1	0
3	0	0	19	40	5	0
4	0	0	5	4	11	0
5	0	0	0	1	1	0

Для 2-го класу якості вина отримано 49 правильних передбачень з 68 проб напою 2-го класу якості, що на шість передбачень більше, ніж у запропонованій моделі.

Для 3-го класу якості вина отримано 40 правильних передбачень з 64 проб напою 3-го класу якості, що на чотири передбачення менше, ніж у першій моделі.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 1 (31)

Отже, з матриці невідповідності не можна зробити висновок, яка модель *ML* краще передбачає якість вина. Однак аналіз матриць демонструє проблему суттєво різної кількості екземплярів у кожному класі, що дає змогу запропонувати напрям майбутніх досліджень – розроблення моделей для незбалансованих класів.

Налаштування гіперпараметрів розглядається як інструмент для створення складних рішень машинного навчання та його додано до частини процесу застосування наукового методу.

Amazon SageMaker дає змогу виконувати автоматичне налаштування гіперпараметрів. Автоматичне налаштування моделі *Amazon SageMaker* (відоме також як налаштування гіперпараметрів) знаходить найкращу версію моделі за допомогою виконання багатьох навчальних завдань на наборі даних. Він використав вказані алгоритми й діапазони гіперпараметрів, потім обрав значення гіперпараметрів, що приводять до найкращої моделі, виміряної обраною метрикою. Він застосовує регресію процесу Гаусса, щоб передбачити, які значення гіперпараметрів можуть бути найбільш ефективними для покращення відповідності, а також упроваджує баєсівську оптимізацію, щоб збалансувати дослідження простору гіперпараметрів і використання їх певних значень, коли це доречно. І що важливо, автоматичне налаштування моделі можна застосовувати із вбудованими алгоритмами *Amazon SageMaker*.

У дослідженні використано метрики робочої характеристики *ROC* (*receiver operator curve*) для оцінювання якості багатокласових класифікаторів. Площа під кривою *AUC* (*Area under the curve*) робочої характеристики приймача (*AUC-ROC*) є ще одним показником оцінки. Площа під кривою оператора приймача (*AUC-ROC*) – ще один показник оцінки. Частина *AUC* в абревіатурі – це площа під нанесеною лінією. Що більша площа *AUC*, то краще модель передбачає властивості речовини за допомогою інтелектуального аналізу даних на основі її фізико-хімічних особливостей. У дослідженні було застосовано площу *AUC* для швидкого порівняння моделей між собою.

Криві *ROC* зазвичай містять чутливість *TPR* на осі *Y* і специфічність *FPR* на осі *X*. Це означає, що верхній лівий кут графіка є "ідеальною" точкою – *FPR* дорівнює нулю, а *TPR* – одиниці. Це не дуже реалістично, але це означає, що більша площа під кривою *AUC* зазвичай є кращою. Крутизна кривих *ROC* також важлива, оскільки ідеально максимізувати *TPR* за умови мінімізації *FPR*.

Застосована в дослідженні багатокласова стратегія "один проти решти" *OvR* (також відома як "один проти всіх") полягає в обчисленні кривої *ROC* для кожного із шести класів якості вина. На кожному кроці цей клас розглядається як позитивний, а решта класів – як негативні. Не потрібно плутати стратегію *OvR*, що використовується для оцінювання багатокласових класифікаторів, зі стратегією *OvR*, яка застосовується для навчання багатокласового класифікатора способом підлаштування набору бінарних класифікаторів (наприклад, за допомогою метаоцінювання *OneVsRestClassifier*). Оцінювання *OvR ROC* можна використовувати для ретельного вивчення будь-яких моделей класифікації незалежно від того, як вони були навчені.

У досліджені використано *LabelBinarizer* для бінаризації цілі внаслідок однократного кодування в режимі *OvR*. Це означає, що цільова форма (*n_samples*) зіставляється із цільовою формою (*n_samples, n_classes*). Було перевірено кодування певного класу. Побудовано криву *ROC*, що показує певний клас.

Побудовано графік, що демонструє результуючу криву *ROC*, для третього класу якості вина проти всіх інших. Побудовано криву *ROC* з використанням мікроусередненого *OvR*.

Мікроусереднення виконує агрегатування внесків від усіх класів (за допомогою *numpy.ravel*) для обчислення середніх показників таким чином:

$$\boldsymbol{TPR} = \frac{\sum_c \boldsymbol{TPc}}{\sum_c \left(\boldsymbol{TPc} + \boldsymbol{FNc}\right)}; \tag{6}$$

$$\boldsymbol{FPR} = \frac{\sum_c \boldsymbol{FPc}}{\sum_c \left(\boldsymbol{FPc} + \boldsymbol{TNc}\right)}. \tag{7}$$

Виконана демонстрація ефекту *numpy.ravel*.

У багатокласовій системі класифікації з дуже незбалансованими класами мікроусереднення є кращим, ніж макроусереднення. У таких випадках можна альтернативно використати зважене макроусереднення. У разі, коли основним інтересом є не графік, а сама оцінка *ROC-AUC*, можна відтворити значення, показане на графіку, за допомогою методу *roc_auc_score()* [19]. Це еквівалентно обчисленню кривої *ROC* із застосуванням методу *roc_curve()*,

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

а потім площі під кривою за допомогою методу *auc()* для незбалансованих (*raveled*) фактичних і передбачених класів.

Обчислення кривої *ROC* додає одну точку за умови максимальної частоти помилкових позитивних результатів за допомогою лінійної інтерполяції та корекції Макклiша [20].

Побудовано криву *ROC* з використанням макросереднього *OvR*.

Отримання макросереднього вимагає обчислення метрики незалежно для кожного класу, а потім узяття середнього за ними, отже, розгляду всіх класів однаково апріорі. Спочатку підсумовуємо істинні / хибні позитивні результати для кожного класу. Це обчислення еквівалентне простому виклику методу *roc_auc_score()*.

Усі криві *OvR ROC* для першої моделі були побудовані разом (рис. 2).

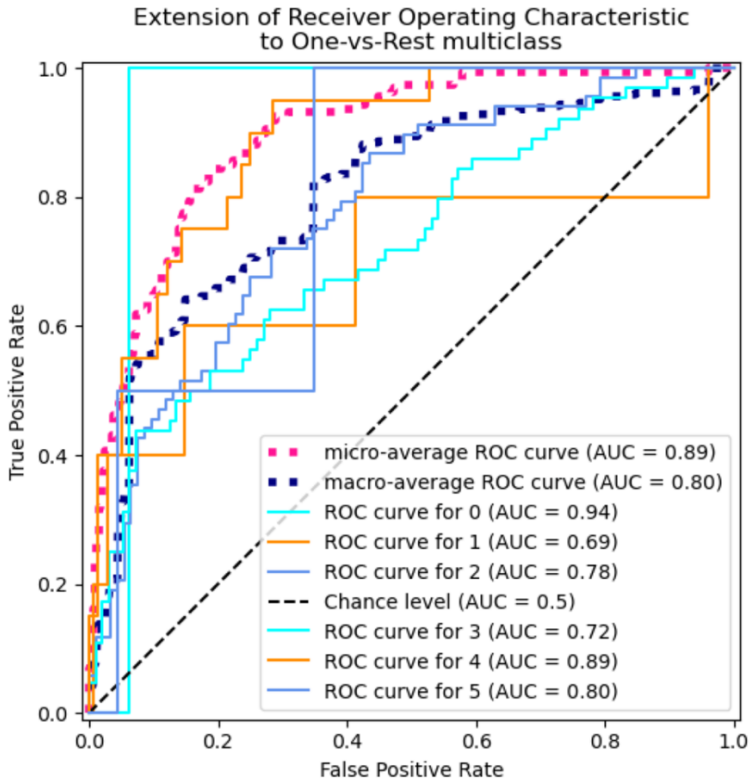

Рис. 2. *OvR ROC* для першої моделі

Для першої моделі концепція мікроусереднення як один із способів узагальнення інформації багатокласових кривих *ROC* демонструє значення *AUC* 0.89 проти значення *AUC* 0.80 для концепції макроусереднення.

Усі криві *OvR ROC* для другої моделі також були побудовані разом (рис. 3).

Концепція мікроусереднення інформації багатокласових кривих *ROC* демонструє значення *AUC* 0.91 проти значення *AUC* 0.83 для концепції макроусереднення.

Висновки

Мета роботи досягнута: розроблена нова модель *ML* та її метрики, що поширюють досягнення попередньої моделі бінарної класифікації на модель багатокласової класифікації та дають змогу порівнювати різні моделі щодо якості аналізу.

Бізнес-проблема аналізу властивостей речовини на основі її фізико-хімічних особливостей перетворена на *ML*-проблему. Отримано надійне визначення проблеми бізнесу та її важливості на основі набору даних, що містить числову інформацію про склад вина разом із його якістю з огляду на склад напою.

Розв'язання *ML*-проблеми розв'язує бізнес-проблему передбачення якості, а отже, і ціни. Більшість бізнес-проблем підпадають під одну з двох категорій: класифікація чи регресія. *ML*-проблему додано до категорії багатозначної класифікації. Ціль належить шістьом класам 0–5.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 1 (31)

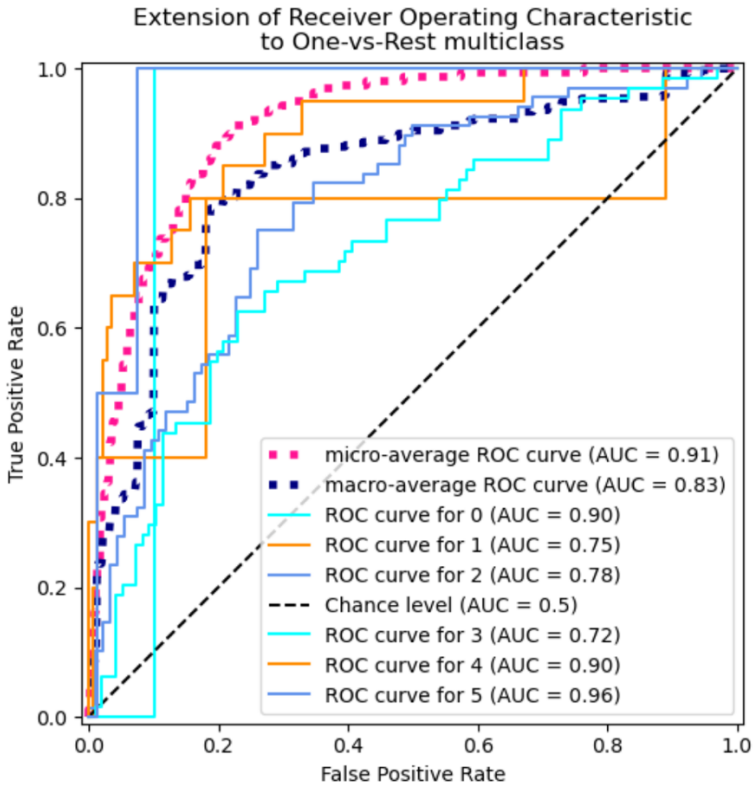

Рис. 3. *OvR ROC* для другої моделі

Для розв'язання проблем *ML* було отримано дані, необхідні для навчання моделі. Хоча в *AWS* доступні керовані інструменти для маніпулювання даними, сценарій був записаний у блокнот *Jupyter* для оброблення інформації. Сценарій вилучення та завантаження містить три розділи *ETL*: імпорт та змінні (бібліотека *boto3*); завантаження та вилучення (вебзапит та збереження байтів з URL в архів ZIP); завантаження на *Amazon S*3.

За допомогою *Pandas*, бібліотеки *Python* з відкритим кодом дані переформатовано з формату *CSV* в табличне подання в рядках і стовпцях. Отримано атрибут *shape DataFrame* (1599,12). Завдяки доменним знанням розв'язано проблеми з типом даних. Діаграма розсіювання допомогла візуалізувати зв'язок між сульфатами та алкоголем. Крім того, діаграма розсіювання допомогла виявити два спеціальні регіони, пов'язані з якісними й неякісними винами. Кореляційна матриця показала високу кореляцію, лінійний зв'язок між алкоголем, сульфатами та якістю напою, коефіцієнти кореляції, відповідно, 0.47 та 0.25. На тепловій карті має місце деяка кореляція 0.67 між лимонною кислотою та фіксованою кислотністю, незначна кореляція –0.68 між фіксованою кислотністю та pH.

Інженерія функцій, яка передбачає вибір або вилучення найкращих функцій для навчання машин, не розглядається, оскільки вона описана в попередніх роботах.

Для перевірки точності моделей дані розділені на набори навчання, перевірки та тестування в такій пропорції: 80 %, 10 % та 10 % відповідно. Використано алгоритм *XGBoost* навчання з учителем і навчальні завдання *AWS SageMaker*.

Навчені моделі розгорнуті за допомогою *AWS SageMaker* для оброблення викликів *API* з програм та для аналізу за допомогою пакетного перетворення, яке використовувалось для отримання характеристики якості передбачення властивостей речовини на основі її фізико-хімічних особливостей, тобто якості передбачень для розв'язання проблеми бізнесу. Застосовувались кінцеві точки однієї моделі.

Для оцінювання якості моделей використано дані, які модель не передбачила через набір утримання. Застосовано матриці невідповідності та *AUC-ROC*. Матриці невідповідності не дали змогу порівняти дві моделі щодо якості передбачення. Однак аналіз матриць сприяв вибору напряму майбутніх досліджень – розробленню моделей для незбалансованих класів. Значення *AUC* 0.89 та 0.91

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

для концепції мікроусереднення і 0.80 та 0.83 для концепції макроусереднення дали змогу зробити однозначний вибір на користь другої моделі.

Тюнінг моделі важливий для того, щоб знайти найкращий розв'язок бізнес-проблеми.

Список літератури

1. Кирсанов О., Кривенко С. Конструювання ознак для застосування навчання машин при обробці клінічних даних. *ICTEE.*2024. Т. 4. № 2. 2024. С. 162–171. DOI: https://doi.org/10.23939/ictee2024.02.162
2. Dua D., Graff C. UCI Machine Learning Repository. *University of California, School of Information and Computer Science*, Irvine, CA. 2019. URL: http://archive.ics.uci.edu/ml (accessed: 07.11.2024)
3. Cortez P., Cerdeira A., Almeida F., Matos T., Reis J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems.* Vol. 47. No. 4. 2009. P. 547–553. DOI: 10.1016/j.dss.2009.05.016
4. Bhatlawande S., Srivastava S., Shilaskar S. Information extraction from digital receipts and bank transactions using machine learning. *2023 International Conference on Integration of Computational Intelligent System (ICICIS).* Pune, India. 2023. P. 1–5. DOI: 10.1109/ICICIS56802.2023.10430289
5. Wu Z., Pulido V. Statistical analyses for fantasy sports. *2022 IEEE Integrated STEM Education Conference (ISEC)*. Princeton, NJ, USA. 2022. P. 269–269. DOI: 10.1109/ISEC54952.2022.10025111
6. Fu W., Widagdo T. E. SQL interface development for spatial data retrieval on Cassandra. *2022 International Conference on Data and Software Engineering (ICoDSE).* Denpasar, Indonesia. 2022. P. 138–143. DOI: 10.1109/ICoDSE56892.2022.9971942
7. Chatziantoniou D., Kantere V., Antoniou N., Gantzia A. Data virtual machines: simplifying data sharing, exploration & querying in big data environments. *2022 IEEE International Conference on Big Data (Big Data).* Osaka, Japan. 2022. P. 373–380. DOI: 10.1109/BigData55660.2022.10020508
8. Geetha V., Sujatha N. An overview of descriptive analytics and data visualization. *2024 5th International Conference on Smart Electronics and Communication (ICOSEC).* Trichy, India. 2024. P. 1158–1163. DOI: 10.1109/ICOSEC61587.2024.10722273
9. Du B., Deng F. The method of network intrusion detection based on descriptive statistics model and Logistic model. *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE).* Guilin, China. 2022. P. 160–163. DOI: 10.1109/MLKE55170.2022.00037
10. Cuartero A., Paoletti M. E., García-Rodríguez P., Haut J. M. PyCircularStats: a Python-based tool for remote sensing circular statistics and graphical analysis. *IGARSS 2022 -- 2022 IEEE International Geoscience and Remote Sensing Symposium.* Kuala Lumpur, Malaysia. 2022. P. 2876–2879. DOI: 10.1109/IGARSS46834.2022.9884758
11. Camacho J., Wasielewska K., Bro R., Kotz D. Interpretable feature learning in multivariate big data analysis for network monitoring. *IEEE Transactions on Network and Service Management*. Vol. 21. No. 3. 2024. P. 2926–2943. DOI: 10.1109/TNSM.2024.3368501
12. Wang Q., Mazor T., Harbig T. A., Cerami E., Gehlenborg N. ThreadStates: state-based visual analysis of disease progression. *IEEE Transactions on Visualization and Computer Graphics.* Vol. 28. No. 1. 2022. P. 238–247. DOI: 10.1109/TVCG.2021.3114840
13. Baes M., Herrera C., Neufeld A., Ruyssen P. Low-rank plus sparse decomposition of covariance matrices using neural network parametrization. *IEEE Transactions on Neural Networks and Learning Systems.* Vol. 34. No. 1. 2023. P. 171–185. DOI: 10.1109/TNNLS.2021.3091598
14. Burgueño-Romero A. M., Benítez-Hidalgo A., Barba-González C., Aldana-Montes J. F. Towards an open-source MLOps architecture. *IEEE Software.* 2024. DOI: 10.1109/MS.2024.3421675
15. Muralikrishna B. S. R. Clinical diagnosis of Alzheimer's disease employing support vector machine. 2*022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE).* Ballari, India. 2022. P. 1–5. DOI: 10.1109/ICDCECE53908.2022.9792897
16. Ghorbani R., Ghousi R., Makui A., Atashi A. A new hybrid predictive model to predict the early mortality risk in intensive care units on a highly imbalanced dataset. *IEEE Access.* Vol. 8. 2020. P. 141066–141079. DOI: 10.1109/ACCESS.2020.3013320
17. Charitha C., Devi Chaitrasree A., Varma P. C., Lakshmi C. Type-II diabetes prediction using machine learning algorithms. *2022 International Conference on Computer Communication and Informatics (ICCCI)*. Coimbatore, India. 2022. P. 1–5. DOI: 10.1109/ICCCI54379.2022.9740844
18. Bezruk V. M., Krivenko S. A., Kyrsanov O. O., Kryvenko S. S., Kryvenko L. S. Training the machine learning model for clinical IoT data and device interoperability. *2023 12th Mediterranean Conference on Embedded Computing (MECO)*. Budva, Montenegro. 2023. P. 1–6. DOI: 10.1109/MECO58584.2023.10154963
19. Smith M. L., Kwembe T. A. Application of machine learning classifiers interfacing Google Colab and Sklearn to intrusion detection CSE-CIC-IDS2018 dataset. *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*. Las Vegas, NV, USA. 2023. P. 1884–1890. DOI: 10.1109/CSCE60160.2023.00311
20. McClish D. K. Analyzing a portion of the ROC curve. *Medical Decision Making*. Vol. 9. No. 3. 1989. P. 190–195. DOI: 10.1177/0272989X8900900307

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2025. No. 1 (31)*

References

1. Kyrsanov, O., Kryvenko, S. (2024), "Feature construction for machine learning application in clinical data processing" ["Konstrujuvannja oznak dlja zastosuvannja navchannja mashyn pry obrobci klinichnyx danux"], *ICTEE.2024*, Vol. 4, No. 2, P. 162–171. DOI: https://doi.org/10.23939/ictee2024.02.162
2. Dua, D., Graff, C. (2019), "*UCI Machine Learning Repository*, University of California, School of Information and Computer Science", Irvine, CA, available at: http://archive.ics.uci.edu/ml (last accessed 07.11.2024)
3. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J. (2009), "Modeling wine preferences by data mining from physicochemical properties", *Decision Support Systems*, Vol. 47, No. 4, P. 547–553. DOI: 10.1016/j.dss.2009.05.016
4. Bhatlawande, S., Srivastava, S., Shilaskar, S. (2023), "Information extraction from digital receipts and bank transactions using machine learning", *2023 International Conference on Integration of Computational Intelligent System (ICICIS)*, Pune, India, P. 1–5. DOI: 10.1109/ICICIS56802.2023.10430289
5. Wu, Z., Pulido, V. (2022), "Statistical analyses for fantasy sports", *2022 IEEE Integrated STEM Education Conference (ISEC)*, Princeton, NJ, USA, P. 269–269. DOI: 10.1109/ISEC54952.2022.10025111
6. Fu, W., Widagdo, T. E. (2022), "SQL interface development for spatial data retrieval on Cassandra", *2022 International Conference on Data and Software Engineering (ICoDSE)*, Denpasar, Indonesia, P. 138–143. DOI: 10.1109/ICoDSE56892.2022.9971942
7. Chatziantoniou, D., Kantere, V., Antoniou, N., Gantzia, A. (2022), "Data virtual machines: simplifying data sharing, exploration & querying in big data environments", *2022 IEEE International Conference on Big Data (Big Data)*, Osaka, Japan, P. 373–380. DOI: 10.1109/BigData55660.2022.10020508
8. Geetha, V., Sujatha, N. (2024), "An overview of descriptive analytics and data visualization", *2024 5th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, P. 1158–1163. DOI: 10.1109/ICOSEC61587.2024.10722273
9. Du, B., Deng, F. (2022), "The method of network intrusion detection based on descriptive statistics model and Logistic model", *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE)*, Guilin, China, P. 160–163. DOI: 10.1109/MLKE55170.2022.00037
10. Cuartero, A., Paoletti, M. E., García-Rodríguez, P., Haut, J. M. (2022), "PyCircularStats: a Python-based tool for remote sensing circular statistics and graphical analysis", *IGARSS 2022 – 2022 IEEE International Geoscience and Remote Sensing Symposium*, Kuala Lumpur, Malaysia, P. 2876–2879. DOI: 10.1109/IGARSS46834.2022.9884758
11. Camacho, J., Wasielewska, K., Bro, R., Kotz, D. (2024), "Interpretable feature learning in multivariate big data analysis for network monitoring", *IEEE Transactions on Network and Service Management*, Vol. 21, No. 3, P. 2926–2943. DOI: 10.1109/TNSM.2024.3368501
12. Wang, Q., Mazor, T., Harbig, T. A., Cerami, E., Gehlenborg, N. (2022), "ThreadStates: state-based visual analysis of disease progression", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 28, No. 1, P. 238–247. DOI: 10.1109/TVCG.2021.3114840
13. Baes, M., Herrera, C., Neufeld, A., Ruyssen, P. (2023), "Low-rank plus sparse decomposition of covariance matrices using neural network parametrization", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 34, No. 1, P. 171–185. DOI: 10.1109/TNNLS.2021.3091598
14. Burgueño-Romero, A. M., Benítez-Hidalgo, A., Barba-González, C., Aldana-Montes, J. F. (2024), "Towards an open-source MLOps architecture", *IEEE Software*, DOI: 10.1109/MS.2024.3421675
15. Muralikrishna, B. S. R. (2022), "Clinical diagnosis of Alzheimer's disease employing support vector machine", *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballari, India, P. 1–5. DOI: 10.1109/ICDCECE53908.2022.9792897
16. Ghorbani, R., Ghousi, R., Makui, A., Atashi, A. (2020), "A new hybrid predictive model to predict the early mortality risk in intensive care units on a highly imbalanced dataset", *IEEE Access*, Vol. 8, P. 141066–141079. DOI: 10.1109/ACCESS.2020.3013320
17. Charitha, C., Devi Chaitrasree, A., Varma, P. C., Lakshmi, C. (2022), "Type-II diabetes prediction using machine learning algorithms", *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, P. 1–5. DOI: 10.1109/ICCCI54379.2022.9740844
18. Bezruk, V. M., Krivenko, S. A., Kyrsanov, O. O., Kryvenko, S. S., Kryvenko, L. S. (2023), "Training the machine learning model for clinical IoT data and device interoperability", *2023 12th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, P. 1–6. DOI: 10.1109/MECO58584.2023.10154963
19. Smith, M. L., Kwembe, T. A. (2023), "Application of machine learning classifiers interfacing Google Colab and Sklearn to intrusion detection CSE-CIC-IDS2018 dataset", *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, Las Vegas, NV, USA, P. 1884–1890. DOI: 10.1109/CSCE60160.2023.00311
20. McClish, D. K. (1989), "Analyzing a portion of the ROC curve", *Medical Decision Making*, Vol. 9, No. 3, P. 190–195. DOI: 10.1177/0272989X8900900307

Надійшла (Received) 01.01.2025

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 1 (31)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

Відомості про авторів / About the Authors

Кирсанов Олександр Олександрович – Харківський національний університет радіоелектроніки, аспірант кафедри інформаційно-мережної інженерії, Харків, Україна; e-mail: oleksandr.kyrsanov@nure.ua; ORCID ID: https://orcid.org/ 0009-0001-5987-4728

Кривенко Станіслав Анатолійович – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, доцент кафедри інформаційно-мережної інженерії, Харків, Україна; e-mail: stanislav.kryvenko@nure; ORCID ID: https://orcid.org/ 0000-0002-5244-1276

Kyrsanov Oleksandr – Kharkiv National University of Radio Electronics, Postgraduate Student at the Department of Information and Network Engineering, Kharkiv, Ukraine.

Kryvenko Stanislav – PhD (Engineering Sciences), Associate Professor, Kharkiv National University of Radio Electronics, Associate Professor at the Department of Information and Network Engineering, Kharkiv, Ukraine.

MACHINE LEARNING MODEL FOR PREDICTING SUBSTANCE PROPERTIES BASED ON ITS PHYSICOCHEMICAL PROPERTIES

Subject matter. The article focuses on extending previous binary classification results to multi-class classification using an ML model to analyze substance properties based on physicochemical characteristics. **Goal**. The primary objective is to develop a new ML model and metrics to compare different models' analysis quality, particularly in predicting wine quality from its composition. **Tasks** are data preparation, model development, training, tuning, evaluation, deployment, and monitoring. **Methods**. The study uses AWS SageMaker for data preparation, model development, training, tuning, evaluation, deployment, and monitoring, with data processed using Jupyter notebooks and pandas. **Results**. Data Analysis: The analysis includes descriptive statistics, correlation matrices, and visualizations like histograms and scatter plots to understand data relationships and quality. Model Training and Evaluation: The models were trained using XGBoost, with data split into training, validation, and testing sets, and evaluated using confusion matrices and AUC-ROC metrics. Confusion Matrix Analysis: Confusion matrices for two models showed mixed results, highlighting the challenge of comparing model performance and the need for further research on unbalanced classes. Hyperparameter Tuning: Amazon SageMaker's automatic hyperparameter tuning was used to optimize model performance, employing Bayesian optimization and Gaussian process regression. ROC-AUC Metrics: The study utilized ROC-AUC metrics to evaluate model performance, with micro-averaging and macro-averaging approaches showing different AUC values for the two models. Key Findings: The second model showed slightly better performance based on AUC metrics, but confusion matrix analysis suggested the need for models tailored to unbalanced classes. **Conclusions**. The research successfully developed a new ML model for multi-class classification, demonstrating its potential for improving wine quality prediction and suggesting future research directions.

Keywords: ML model; cloud; Confusion Matrix Analysis; predicted quality; actual quality; ROC-AUC; data; parameters; averaging.

Бібліографічні описи / Bibliographic descriptions

Кирсанов О. О., Кривенко С. А. Модель ML для аналізу властивостей речовини на основі її фізико-хімічних особливостей. *Сучасний стан наукових досліджень та технологій в промисловості*. 2025. № 1 (31). С. 151–165. DOI: https://doi.org/10.30837/2522-9818.2025.1.151

Kyrsanov, O., Kryvenko, S. (2025), "Machine learning model for predicting substance properties based on its physicochemical properties", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (31), P. 151–165. DOI: https://doi.org/10.30837/2522-9818.2025.1.151