Науковий вісник НЛТУ України
Scientific Bulletin of UNFU
ISSN 1994-7836 (print)
ISSN 2519-2477 (online)
https://doi.org/10.36930/40330315
Article received 29.04.2023
Article accepted 25.05.2023
UDC 004.[8+048+622]
Correspondence author: V. V. Petryna volodymyr.v.petryna@lpnu.ua

В. В. Петрина, А. В. Дорошенко, Р. В. Сидоренко, В. М. Теслюк
Національний університет "Львівська політехніка", м. Львів, Україна

МОДЕЛЬ ТА ЗАСОБИ ЗБИРАННЯ ТА ОБРОБЛЕННЯ ДАНИХ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Досліджено вплив ітеративного методу зважування даних респондентів на підставі певних факторів на точність навчання моделі машинного навчання для вирішення завдань класифікації. Збір та оброблення даних є критичним етапом у процесі розроблення та використання моделей машинного навчання, оскільки якість та наочність даних безпосередньо впливають на точність та ефективність моделей. Проаналізовано математичне забезпечення алгоритмів моделей класифікації. Здійснено огляд літературних джерел, пов'язаних із тематикою статті. Проаналізовано набори даних, доступні у мережі для вирішення завдань класифікації. Розроблено програмне забезпечення для роботи із моделями машинного навчання. Проведено попередню підготовку вхідних даних для навчання та тестування вибраних моделей. Використано такі моделі класифікації, як наївний класифікатор Байєса, класифікатор випадкового лісу, наївний байєсів класифікатор Гауса, а також ітеративний метод зважування даних. Ці моделі інтегровано у програмне забезпечення, розроблене для оброблення, підготовки, зберігання даних. Досліджено обрані моделі із використанням попередньо підготовлених даних за допомогою програмного забезпечення відповідно до визначених сценаріїв. Згідно з результатами дослідження виявлено позитивний тренд на якість навчання моделей за коректної підготовки даних і вибору відповідних змінних для зважування даних респондентів. Показники ефективності, точності навчання алгоритму показують позитивну динаміку порівняно з результатами тестування моделей без використання зважування даних. Результатами дослідження підтверджується значущий вплив ітеративного методу зважування даних на результати навчання, тренування та тестування моделей машинного навчання, а саме мультиплікативного класифікатора Байєса.

Ключові слова: оброблення даних; алгоритми; наївний класифікатор Байєса; класифікатор випадкового лісу; наївний байєсів класифікатор Гауса; класифікація; аналіз даних; метрики оцінки моделей; зважування даних.

Вступ / Introduction

Тенденції останніх декількох років свідчать про надзвичайну цінність даних у сучасному суспільстві. Завдяки швидкому розвитку технологій ми стали свідками значного збільшення обсягу доступної інформації, що потребує адекватного збирання, оброблення та аналізу. Тому збиранню та обробленню даних з використанням машинного навчання належить центральне місце у сфері аналізу даних.

Машинне навчання (англ. *Machine Learning*, ML), як сучасний підхід до оброблення даних, забезпечує можливість автоматизованого отримання цінної інформації з великого обсягу даних. Це полегшує роботу аналітикам, дослідникам та бізнес-експертам, даючи змогу їм ефективніше виділяти залежності, виявляти патерни та робити передбачення.

Розроблено модель та засоби збирання й оброблення даних з використанням машинного навчання. Описано основні принципи машинного навчання, які лежать в основі цих процесів, а також проаналізовано інструменти та техніки і особливості алгоритмічного забезпечення, які допомагають ефективно збирати та обробляти дані. Окрім цього, наведено приклади застосування алгоритмів для машинного навчання для вирішення реальних проблем, що сприяють удосконаленню процесу збирання та оброблення даних.

Вивчення та розроблення моделей та засобів збирання та оброблення даних з використанням машинного навчання дає змогу нам краще розуміти сутність цих процесів та їх вплив на розвиток нашого суспільства.

Об'єкт дослідження – процес збирання та ефективного оброблення даних для застосування в алгоритмах машинного навчання.

Предмет дослідження – моделі та засоби попереднього оброблення даних для машинного навчання.

Інформація про авторів:
Петрина Володимир Васильович, аспірант, кафедра автоматизованих систем управління.
Email: volodymyr.v.petryna@lpnu.ua; https://orcid.org/0009-0009-1330-199X
Дорошенко Анастасія Володимирівна, канд. техн. наук, доцент, кафедра автоматизованих систем управління.
Email: anaastasiia.v.doroshenko@lpnu.ua; https://orcid.org/0000-0002-7214-5108
Сидоренко Роман Вікторович, асистент, кафедра автоматизованих систем управління.
Email: roman.v.sydorenko@lpnu.ua; https://orcid.org/0000-0002-9026-301X
Теслюк Василь Миколайович, д-р техн. наук, професор, завідувач кафедри автоматизованих систем управління.
Email: vasyl.m.teslyuk@lpnu.ua; https://orcid.org/0000-0002-5974-9310

Цитування за ДСТУ: Петрина В. В., Дорошенко А. В., Сидоренко Р. В., Теслюк В. М. Модель та засоби збирання та оброблення даних з використанням машинного навчання. Науковий вісник НЛТУ України. 2023, т. 33, № 3. С. 102–109.
Citation APA: Petryna, V. V., Doroshenko, A. V., Sydorenko, R. V., & Teslyuk, V. M. (2023). Model and means of data collection and processing using machine learning. *Scientific Bulletin of UNFU*, 33(3), 102–109. https://doi.org/10.36930/40330315

Мета роботи – розроблення моделі та засобів збирання та оброблення даних для вирішення завдань класифікації з використанням машинного навчання.

Для досягнення зазначеної мети визначено такі основні *завдання дослідження*:

1. Розроблення системи збирання, оброблення та зберігання даних із використанням сучасних технологій, що дасть змогу покращити підготовку даних для завдань машинного навчання.
2. Аналіз та порівняння моделей та базових алгоритмів машинного навчання для вирішення завдань класифікації об'єктів.
3. Застосування методу зважування даних для підвищення точності класифікації об'єктів.

Аналіз останніх досліджень та публікацій. Для вирішення завдань класифікації використовуються такі алгоритми, як наївний класифікатор Байєса, класифікатор випадкового лісу, наївний байєсів класифікатор Гауса та RIM-weighting алгоритми [16]. Наївний класифікатор Байєса використовує теорему Байєса для визначення ймовірності приналежності спостереження (елемента вибірки) до одного з класів при припущенні (наївному) незалежності змінних. Тобто, якщо на підставі значень змінних можна однозначно визначити, до якого класу належить спостереження, класифікатор Байєса повідомить ймовірність приналежності до цього класу.

Байєсівська система класифікації широко використовується в багатьох галузях, проте існують труднощі коваріаційного матричного аналізу, зазвичай складно надійно оцінити. Для вирішення проблеми розроблено багато наївних підходів Байєса (NB) із доброю продуктивністю. Однак припущення про умовну незалежність між атрибутами в NB рідко виконується насправді. Для вирішення цієї проблеми розроблено різні схеми зважування атрибутів. Серед них нещодавно досягнуто хороших результатів за допомогою класифікаційного зворотного зв'язку для оптимізації ваги атрибутів кожного класу [18].

У багатьох реальних завданнях класифікації не всі класи даних представлені однаково. Ця проблема, відома також як "прокляття" незбалансованого класу в наборах даних, має потенційний вплив на процедуру навчання класифікатора у спосіб вивчення моделі, яка буде упередженою на користь класу більшості. У роботі [2] автори пропонують підхід із недостатньою вибіркою з використанням простого класифікатора Байєса, щоб вибрати найбільш інформативні екземпляри з доступного навчального набору на підставі випадкового початкового набору.

Окрім цього, в роботі [19] запропоновано багато підходів до зважування атрибутів для покращання моделей, наприклад класифікатора Байєса способом пом'якшення припущення про умовну незалежність. У реальних навчальних програмах різні атрибути відіграють різні ролі для проблем класифікації, тому зважування атрибутів може потенційно покращити ефективність класифікації завдяки призначенню різних ваг різним атрибутам. Зважування атрибутів має на меті зважити кожен атрибут відповідно до його прогнозної здатності класифікації. Він збільшує ваги атрибутів із високим ступенем прогнозування та знижує ваги атрибутів із слабким прогнозуванням.

Продуктивність моделі машинного навчання можна підвищити [9], якщо використовувати збалансований набір даних для навчання та тестування моделі. Окрім цього, можливості прогнозування моделі можна покращити, використовуючи належні та пов'язані характеристики даних. Тому балансування даних і вибір функцій дуже важливі для покращання продуктивності моделі.

Результати дослідження та їх обговорення / Research results and their discussion

Математичне забезпечення засобів збирання та оброблення даних з використанням машинного навчання. Наївний класифікатор Байєса є одним із найпоширеніших алгоритмів машинного навчання для задачі класифікації. Він ґрунтується на теоремі Байєса та використовує припущення незалежності між ознаками (відтак назва "наївний") [6].

Загалом формула для наївного класифікатора Байєса має такий вигляд:

$$P(y|x_1,x_2,...,x_n)=\frac{P(x_1,x_2,...,x_n|y)\cdot P(y)}{P(x_1,x_2,...,x_n)}, \quad (1)$$

де: $P(y|x_1,x_2,...,x_n)$ – умовна ймовірність того, що об'єкт з ознаками $x_1,x_2,...,x_n$ належить до класу y; $P(x_1,x_2,...,x_n|y)$ представляє умовну ймовірність того, що об'єкт з класу y має ознаки $x_1,x_2,...,x_n$; $P(y)$ є апріорною ймовірністю класу y, тобто імовірністю, що випадково вибраний об'єкт належить до класу y; $P(y|x_1,x_2,...,x_n)$ є маргінальною ймовірністю ознак $x_1,x_2,...,x_n$.

Застосування наївного класифікатора Байєса полягає у визначенні ймовірностей $P(y|x_1,x_2,...,x_n)$ для кожного класу у і виборі класу з найбільшою ймовірністю. Наївний класифікатор Байєса є ефективним та широко використовується у багатьох областях, таких як тексти, медицина, розпізнавання образів тощо. Його привабливість полягає у простоті реалізації та швидкості роботи.

Випадковий ліс – це ансамбль дерев рішень, який використовується для задач класифікації. Кожне дерево рішень у випадковому лісі прогнозує клас об'єкта, і остаточний результат визначається шляхом голосування більшості дерев [3].

Формулу для класифікатора випадкового лісу можна подати так:

$$C(x)=\text{mode}(C_1(x),C_2(x),...C_n(x)), \quad (2)$$

де: $C(x)$ – клас, передбачений випадковим лісом для об'єкта з ознаками x; $C_1(x),C_2(x),...C_n(x)$ – класи, передбачені кожним окремим деревом випадкового лісу для об'єкта з ознаками x; $\text{mode}(\cdot)$ – функція, яка повертає найчастіший клас серед переданих значень.

Побудова випадкового лісу передбачає виконання таких кроків [16]:

Крок 1. Випадковий вибір підвибірки даних із навчального набору.

Крок 2. Випадковий вибір підмножини ознак для кожного дерева.

Крок 3. Побудова дерева рішень на підвибірці даних із використанням обраної підмножини ознак.

Крок 4. Повторення кроків 1-3 для кількості дерев, визначеної параметром моделі.

Крок 5. Класифікація нового об'єкта способом голосування більшості дерев.

Випадковий ліс є потужним і надійним алгоритмом, який добре працює з великими наборами даних та може уникнути перенавчання. Він широко використовується у багатьох галузях, таких як медицина, фінанси, комп'ютерний зір тощо [6].

Класифікатор Гауса, також відомий як нормальний класифікатор Байєса, є статистичним методом, який використовується для задач класифікації. Він ґрунтується на припущенні, що розподіл ознак кожного класу є Гаусівським (нормальним) [3].

Формула для класифікатора Гауса має такий вигляд:

$$P(x|y) = \frac{P(x|y) \cdot P(y)}{P(x)}, \quad (3)$$

де: $P(x|y)$ є умовною ймовірністю того, що об'єкт з ознаками x належить до класу y; $P(x|y)$ є умовною ймовірністю ознак x за умови, що об'єкт належить до класу y; $P(y)$ є апріорною ймовірністю класу y, тобто ймовірністю, що випадково вибраний об'єкт належить до класу y; $P(x)$ є маргінальною ймовірністю ознак x.

Для класифікатора Гауса припускається, що розподіл ознак x за умови, що об'єкт належить до класу y, є розподілом Гауса з певними параметрами (середнє значення та дисперсія). Ці параметри можуть бути визначені з тренувальних даних для кожного класу [5].

Під час класифікації нового об'єкта класифікатор Гауса використовує наведену вище формулу, щоб визначити ймовірність належності об'єкта до кожного класу, а потім вибирає клас із найбільшою ймовірністю.

Алгоритм зважування – це метод, що використовує ваги для призначення різних рівнів важливості ознак або даних. Цей алгоритм застосовується в багатьох сферах, таких як машинне навчання, статистика, оптимізація та ін.

Вираз для зваженого алгоритму можна записати так:

$$y = w_1x_1 + w_2x_2 + ... + w_nx_n, \quad (4)$$

де: y – вихідний результат алгоритму; $x_1, x_2, \ldots, x_n$ – вхідні дані або ознаки; $w_1, w_2, \ldots, w_n$ – ваги, які відображають важливість кожної ознаки.

Вагові коефіцієнти $w_1, w_2, \ldots, w_n$ можна встановити заздалегідь на підставі досвіду або експертних знань, або ж їх можна визначити в процесі навчання моделі. Алгоритм зважування дає змогу керувати впливом кожної ознаки на остаточний результат. Способом зміни ваг можна надавати більшої вагомості важливим ознакам та меншої – менш важливим. Це дає змогу адаптувати алгоритм до конкретних вимог й отримати кращі результати в різних ситуаціях.

Застосування зваженого алгоритму може бути корисним для вирішення проблем з нерівномірним розподілом важливості ознак або в ситуаціях, коли деякі ознаки мають більший вплив на остаточний результат, ніж інші.

На підставі виразів (1)-(4) розроблено моделі засобів збирання та оброблення даних із використанням методів машинного навчання.

Розроблення структурної схеми. Модулі системи збирання та ефективного оброблення даних і застосування алгоритмів машинного навчання відіграють важливу роль у процесі розроблення моделей машинного навчання. Основна мета розроблених моделей та модулів полягає в підготовці даних для подальшого застосування алгоритмів машинного навчання, забезпечуючи при цьому оптимальні умови для навчання моделей та досягнення кращих результатів передбачення. У роботі розроблено схему програмного забезпечення, структура якої містить такі модулі:

1. Модуль збирання даних: система дає змогу завантажувати файли різних форматів (CSV-файли, avro, parquet).
2. Модуль попереднього оброблення даних: дані передаються на крок попереднього оброблення, де здійснюються їх очищення, нормалізація та перетворення на формат, зручний для подальшого аналізу. Цей процес може містити видалення аномальних даних, заповнення пропущених значень, зміну формату та інші операції.
3. Модуль розділення навчальної та тестової вибірки: дані розділяються на навчальну та тестову вибірки. Навчальна вибірка використовується для навчання моделі, тоді як тестова вибірка – для перевірки точності моделі.
4. Модуль використання алгоритмів машинного навчання: з використанням алгоритмів машинного навчання розробляються та навчаються моделі, що передбачають залежність між вхідними та вихідними даними.
5. Засоби тестування моделей: навчені моделі перевіряються на тестовій вибірці для оцінювання їх точності та ефективності.

Розроблена структурна схема ґрунтується на модульному принципі, що дає змогу швидко вдосконалювати програмну систему.

Особливості алгоритмічного забезпечення. Алгоритмічне забезпечення системи містить сукупність алгоритмів, які використовуються для вирішення різних завдань у комп'ютерній системі. Упродовж останніх років величезної популярності набувають системи штучного інтелекту. Одним із основних факторів у роботі із такими системами є підбір коректних моделей відповідно до поставленої задачі і коректне навчання моделей. Для вирішення завдання, поставленого під час дослідження, обрано декілька основних алгоритмів класифікації, таких як наївний байєсів класифікатор Гауса, мультиплікативний наївний класифікатор Байєса, Random Forest.

Для послідовності виконання операцій у роботі із алгоритмами наведемо блок-схеми кожного із алгоритмів. Блок-схему наївного класифікатора Байєса зображено на рис. 1.

Рис. 1. Блок-схема алгоритму наївного класифікатора Байєса / Block diagram of the naive Bayesian classifier algorithm

Ця блок-схема описує процес тренування та тестування наївного класифікатора Байєса для завдання класифікації. Під час тренування класифікатор обчислює апріорну та умовну ймовірності для кожної унікальної мітки класу та кожного ознакового стовпця. Під час тестування класифікатор обчислює апостеріорні ймовірності для кожної унікальної мітки класу для кожного запису в тестовому наборі та зараховує його до класу з найбільшою апостеріорною ймовірністю.

Блок-схему класифікатора випадкового лісу зображено на рис. 2.

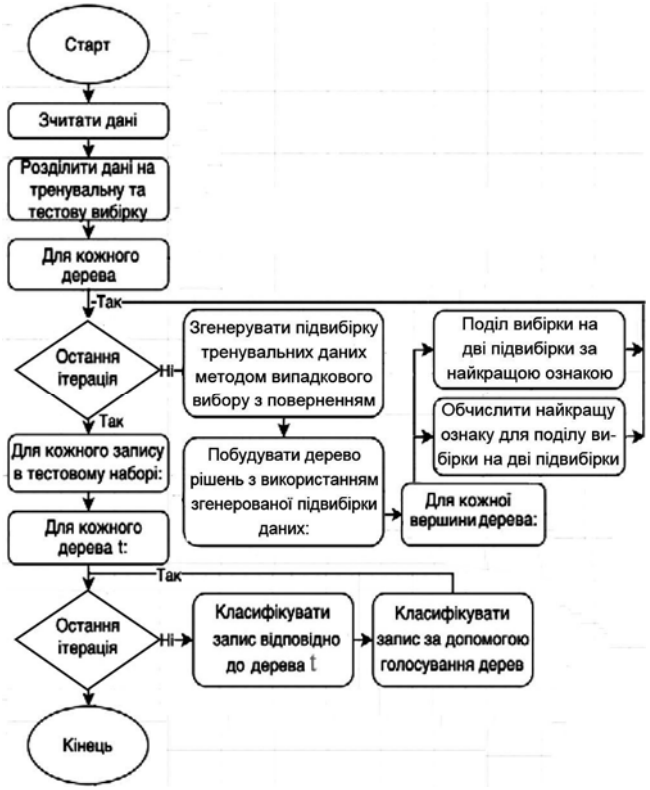

Рис. 2. Блок-схема класифікатора випадкового лісу / Block diagram of the random forest classifier

Ця блок-схема описує процес тренування та тестування випадкового лісу для задачі класифікації. Під час тренування для кожного дерева в лісі випадково вибирається підвибірка тренувальних даних із поверненням, – на цій підвибірці побудовано дерево рішень. Під час тестування кожен запис із тестового набору класифікується кожним деревом у лісі, а потім відбувається голосування дерев. Унаслідок, запис належить до класу, який набрав найбільшу кількість голосів.

На рис. 3 наведемо блок-схему наївного байєсового класифікатора Гауса.

Ця блок-схема описує процес тренування та тестування наївного байєсового класифікатора Гауса для задачі класифікації. Під час тренування для кожного класу обчислюються середнє значення та коваріаційна матриця для кожного атрибута. Під час тестування для кожного запису у тестовому наборі обчислюється ймовірність того, що запис належить до кожного класу, використовуючи формулу Гауса-Ньютона. Запис класифікується в клас з найбільшою ймовірністю.

Отже, на підставі аналізу алгоритмів для виконання дослідження вибираємо наївний класифікатор Байєса. Під час експерименту вирішується проблема класифікації: чи підпишеться клієнт на терміновий депозит на підставі його демографічних і маркетингових факторів. Для навчання обрано основними параметрами вік, наявність позик, поточний фінансовий баланс респондента, а також фактор зважування даних параметрів. Наївний класифікатор Байєса визначає ймовірність відношення елемента за допомогою значень змінних якомога точніше, що найбільше відповідає умові задачі.

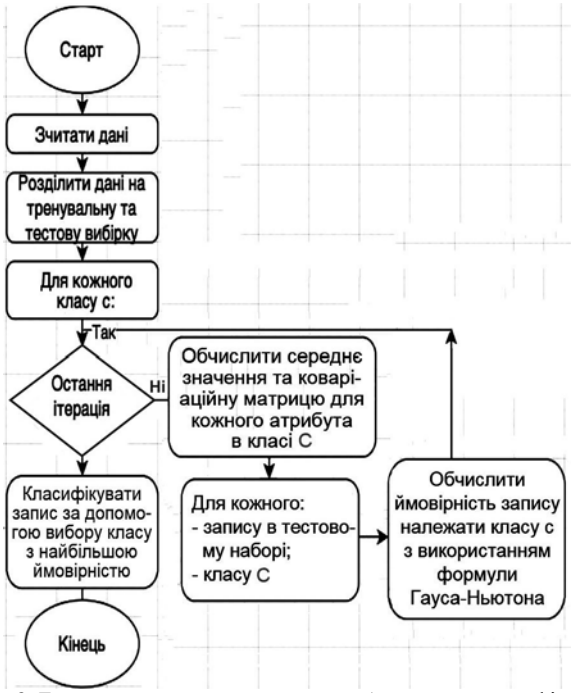

Рис. 3. Блок-схема алгоритму наївного байєсового класифікатора Гауса / Block diagram of the naive Bayesian Gauss classifier algorithm

Розроблення програмного та інформаційного забезпечення. Перед початком розроблення проаналізовано засоби розроблення інтелектуальної компоненти. Мова програмування Python [12], що певною мірою через використання інтерпретатора поступається за швидкістю компільованим мовам програмування, таким як Java, водночас, має низку переваг для розв'язання задач машинного навчання. Основні переваги, які стали вирішальними для вибору мови програмування, такі:

- велика кількість бібліотек, у яких уже реалізовано всі необхідні алгоритми та інтерфейси;
- розвинена спільнота розробників у галузі машинного навчання, що підтримують бібліотеки згідно з останніми розробками;
- велика кількість інформації і можливих вирішень проблем, що доступні у мережі;
- GIL – global interpreter lock, система, що гарантує однопоточність виконання програми мовою Python. Це дає змогу під час операцій розрахунків гарантувати однаковий результат й уникати можливих проблем із спільною пам'яттю.

Для веб-сервісу обрано фреймворк Fast API [17]. Перевагами цього фреймворку є швидкість розроблення програмного забезпечення, проста документація, велика кількість інформації у відкритому доступі, використання останніх технологій, інтегроване використання конкурентності. Також одним із найбільших плюсів є впровадження автодокументації коду, що дає змогу ділитись із програмою, яка містить усю необхідну документацію для використання. Для збереження бекенду використано базу даних Postgresql [11]. Для розгортання аплікації застосовано Docker [4], що дає змогу контейнеризувати різні компоненти і дає можливість розроблення модульним способом. Для розроблення візуальної частини програмного забезпечення частини об-

рано фреймворк Angular [1]. Для розроблення алгоритмів ML-аналізу використано Python бібліотеки Pandas та Scikit-learn [14].

Для моделі було обрано наївний байєсів класифікатор Гауса, мультиплікативний наївний класифікатор Байєса, Random Forest. Діаграму класів програмного забезпечення зображено на рис. 4. Призначення кожного з класів наведено у табл. 1.

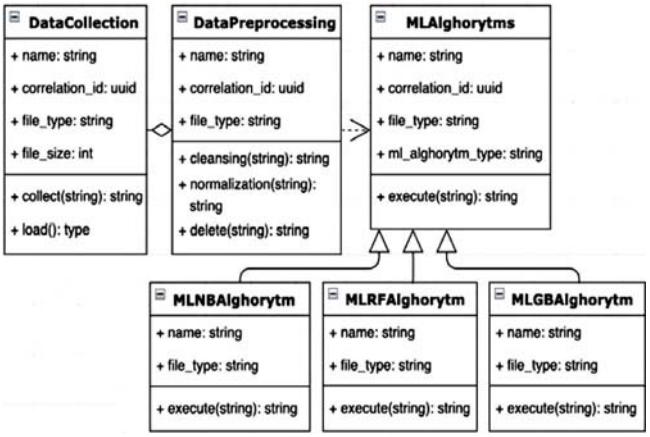

Рис. 4. Структурна схема діаграми класів / Structural diagram of the class diagram

Табл. 1. Опис ролі класів / Description of the role of classes

№	Назва класу	Призначення
1	Data Collection	Клас, відповідальний за завантаження даних різних форматів
2	Data Preprocessing	Клас, відповідальний за передобробку даних, враховуючи очищення, нормалізацію, видалення дублікатів, заповнення пропущених значень та інші операції для забезпечення якості та консистентності даних
3	Machine Learning Algorithms	Клас, який містить реалізації різних алгоритмів машинного навчання, таких як класифікаційні, регресійні, кластеризаційні, ансамблеві алгоритми та інші
4	Model Training	Клас, відповідальний за навчання моделей машинного навчання на попереднього оброблених даних з використанням вибраних алгоритмів
5	Model Evaluation	Клас, відповідальний за оцінку ефективності навчених моделей

Розроблене програмне забезпечення має такі переваги, а саме:

- швидке розгортання системи. Система побудована із використанням програми Docker, що дає змогу контейнеризувати окремо компоненти, будувати залежності між системами;
- використання останніх на цей момент версій бібліотек і технологій. Дає змогу будувати систему згідно з останніми програмними досягненнями;
- використання передових бібліотек із розроблення даних. Використано алгоритми із максимальними оптимізаціями, що дає змогу проводити різні експерименти і досягати максимальних результатів;
- модульність. Кожен компонент системи є незалежним, застосовані підходи SOLID, DRY, REST. Є можливість відлагодження системи незалежності від інших компонентів.

Результати експериментів та їх обговорення. Для проведення експерименту обрано маркетинговий набір банківських даних [7]: цей набір даних містить інформацію про маркетингові кампанії португальської банківської установи, зокрема вік, стать, регіон, роботу, сімейний стан і канали зв'язку клієнта, які використовуються для маркетингу. Його можна використовувати, щоб передбачити, чи підпишеться клієнт на терміновий депозит на підставі його демографічних і маркетингових факторів. Під навчання моделей, для порівняння результатів передбачення застосовано також ітеративний метод зважування даних. Отже, з'являється можливість порівняти вплив зважування основних факторів на результати передбачення.

Першим і, водночас, одним із основних етапів експерименту є підготовка і попереднє оброблення даних. Використовуючи попередньо розроблену систему і можливості, що вона надає, необхідно завантажити файл із набором даних для тренування чи тестування моделей та їх оброблення. Аплікація надає можливість аналізувати й обробляти дані за допомогою статистичних показників із діаграм, а також наведеного прикладу даних. Далі користувач має можливість здійснювати фільтрування даних згідно з параметрами залежно від потреби.

Система надає можливість фільтрувати за класами, вказувати певну кількість конкретних класів, а також задану кількість випадкових рядків із набору даних. Обравши критерій фільтрації, система надає можливість застосувати фільтрацію і сформувати у такий спосіб набір даних, що є результатом, або ж зберегти цей фільтр для подальшого використання [10]. На наступному кроці система після однієї із двох операцій відправляє на вкладку Data, де можна переглянути результати фільтрації та усіх попередніх активностей системи. На цій вкладці відображаються усі завантажені файли клієнтом, створені фільтри, а також набори даних, що є результатами. Також є окремо вкладка Filters, де можна провести фільтрацію, обравши файл із попередньо завантажених. Далі в аплікацію додано докер контейнер із jupyter notebook, куди клієнт може перейти для написання та тестування машинного навчання та іншого коду. Це вкладка ML Jupyter Notebook.

Для експерименту визначено 5 різних сценаріїв повного циклу операцій. Вхідними даними обрано набір даних із сайту Kaggle [7]. Основною моделлю згідно з аналізом обрано мультиплікативний наївний класифікатор Байєса.

Вхідні дані: набір даних = 5172 рядків, кількість респондентів, які взяли депозит = 3672, кількість респондентів, які не взяли депозит = 1500.

Учителем або ж лейблом у цьому наборі даних є колонка "Prediction", де "Prediction" 0 не взяв депозит, "Prediction" 1 взяв депозит. Набір даних на вхід уже підготовлений, токени визначені. Додатково проведено зважування даних на підставі колонок "вік", "наявність позики", а також "фінансовий баланс". Ці показники обрано експериментально, для визначення таких, що мають найвагоміший вплив на навчання алгоритму.

Під час проведення експерименту за підготовки даних для машинного навчання було протестовано такі сценарії:

1) Із вхідного набору даних необхідно профільтрувати 1500 повідомлень, де колонка "Prediction" = 0 і аналогічна кількість колонки "Prediction" = 1. Цей набір даних застосовано як тренувальний, а також як перший для тестування. Для тренування потрібно розбити вхідний набір даних на 75 % тренувальних даних, 25 % тестових. Тренувальна вибірка повинна містити приблизно однакову кількість позитивних і негативних кейсів для тестування та тренування.
2) Із вхідного набору даних необхідно профільтрувати 750 повідомлень, де колонка "Prediction" = 0 і 1500 де "Prediction" = 1.

3) Із вхідного набору даних необхідно отримати тільки тих респондентів, які не взяли депозит.
4) Із вхідного набору даних необхідно отримати випадкове число даних, використано 4300.
5) Із вхідного набору даних необхідно отримати тільки тих респондентів, які взяли депозит.

Для оцінювання результатів задач класифікації використовуються різні метрики, які вимірюють різні аспекти точності та ефективності класифікатора. Основні метрики, які часто використовуються і обрані для експерименту, такі [3]:

1. Точність (Accuracy) є одним з найпоширеніших метрик оцінювання ефективності моделей класифікації. Показує відношення кількості правильно класифікованих прикладів до загальної кількості прикладів у наборі даних.
2. Влучність (Precision) є метрикою оцінювання ефективності моделей класифікації. Показує відношення кількості правильно класифікованих позитивних прикладів до загальної кількості прикладів, які модель визначила як позитивні.
3. Повнота (Recall) є метрикою оцінювання ефективності моделей класифікації. Показує відношення кількості правильно класифікованих позитивних прикладів до загальної кількості реальних позитивних прикладів у наборі даних.
4. F-міра (F-measure) є гармонічним середнім двох показників: точності (Precision) і повноти (Recall). Вона є метрикою оцінювання ефективності моделей класифікації, яка комбінує обидва ці показники для отримання загальної міри ефективності.

Також для відображення результатів класифікації обрано матрицю невідповідностей. Матриця невідповідностей, водночас, представляє співвідношення помилок до коректно спрогнозованих результатів. Нижче наведено приклад експерименту.

Рис. 5. Матриця невідповідностей сценарію 1 із зважуванням / Scenario 1 confusion matrix with weighting

Сценарій 1. Із основного набору даних після застосовування фільтрації вибрано 3000 рядків. Набір даних складається із 1500 рядків, у яких респондент обрав депозит і, водночас, 1500 рядків, де респондент не обрав депозит. Далі за допомогою методів бібліотеки pandas цей набір даних поділено на тренувальну вибірку, яка містить 75 % від загального набору даних, а саме 2250 рядків і 750 рядків для відповідного тестування. Набори даних для тестування містять приблизно однакову кількість рядків де обрано і не обрано депозит. Додатково на одному і тому самому наборі даних проведено тренування із використанням зважування факторів та без нього. У табл. 2 наведено результати метрик за сценарієм 1 із використанням зважування.

Табл. 2. Метрики оцінювання класифікації сценарію 1 із застосуванням зважування / Evaluation metrics for scenario 1 classification using weighting

Метрика	Результат
Точність	0,945
Влучність	0,935
Повнота	0,961
F-міра	0,948

Результати показників, зазначених у табл. 1, вимірюються у шкалі від 0 до 1, тому одиницею їх виміру є частка, а саме часткове співвідношення. Матрицю невідповідностей для цього сценарію наведено на рис. 5.

У табл. 3 і на рис. 6 зображено результати тренування цієї самої моделі, але без зважування факторів.

Табл. 3. Метрики оцінювання класифікації сценарію 1 без застосуванням зважування / Evaluation metrics for scenario 1 classification without weighting

Метрика	Результат
Точність	0,911
Влучність	0,962
Повнота	0,86
F-міра	0,908

Рис. 6. Матриця невідповідностей сценарію 1 без зважування / Confusion matrix of scenario 1 without weighting

1) Із вхідного набору даних необхідно профільтрувати 750 повідомлень, де колонка "Preditction" = 0 і 1500 де "Preditction" = 1.
2) Із вхідного набору даних необхідно отримати тільки тих респондентів, які не взяли депозит.
3) Із вхідного набору даних необхідно отримати випадкове число даних, використано 4300.
4) Із вхідного набору даних необхідно отримати тільки тих респондентів, які взяли депозит.

Згідно з результатами обчислення метрик, можна виділити, що застосування зважування факторів неістотно впливає на результати тренування даних у позитивний бік. Відповідно до цього можна зробити висновок, що за коректного зважування даних є можливість підвищити точність навчання моделі наївного класифікатора Байєса.

Наступні експерименти проведено аналогічно до першого сценарію щодо підготовки тренувальних і тестових наборів даних. Для тестування сценарію 2 із вибірки профільтровано 750 повідомлень, де респондент не підписався на депозит і 1500, де респонденти підписалися. За сценарієм 3 обрано вибірку із респондентів, що не взяли депозит. У сценарії 4 обрано випадкове число даних із більш-менш рівномірним розподілом даних щодо підписки на депозит у кількості 4300. Під завершення сценарію 5 обрано тільки тих респондентів,

що взяли депозит. Згенеровані тестові набори даних для кожного сценарію використано для тестування моделі.

У табл. 4 наведено результати наступних чотирьох експериментів і порівняно їхні результати із застосуванням зважування та без зважування даних. Згідно з даними цієї таблиці, простежується тренд впливу застосування зважування факторів із тенденцією до покращання результатів тренування даних, які згодом буде використано під час процесу їх навчання, та використано для допасовування параметрів (зокрема, вагових коефіцієнтів) для класифікатора. Більшість підходів, які здійснюють пошук емпіричних взаємозв'язків у тренувальних даних, мають схильність до перенавчання цих даних, тобто, можуть виявляти та використовувати видимі взаємозв'язки в тренувальних даних, які в загальному випадку не є дійсними.

Табл. 4. Метрики оцінювання класифікації сценаріїв 2-5 / Evaluation metrics for the classification of scenarios 2-5

№ сценарію	Результат із зважуванням				Результат без зважування			
	Точність	Влучність	Повнота	F-міра	Точність	Влучність	Повнота	F-міра
2	0,947	0,946	0,947	0,947	0,908	0,916	0,908	0,906
3	0,91	0,846	0,91	0,882	0,902	0,834	0,909	0,873
4	0,948	0,952	0,948	0,947	0,944	0,95	0,938	0,937
5	0,954	0,91	0,954	0,932	0,942	0,901	0,948	0,927

Обговорення результатів дослідження. Багато дослідників приділяли увагу вирішенню проблем класифікації і підвищенню точності навчання та тестування моделей машинного навчання. Згідно з даними роботи [8], незважаючи на те, що наївний класифікатор Байєса доволі ефективний у різних завданнях інтелектуального аналізу даних, він показує невтішний результат під час автоматичної класифікації тексту. Ґрунтуючись на результатах використання наївного класифікатора Байєса для аналізу тексту природною мовою, виявлено серйозну проблему оцінювання параметрів, яке спричиняє погані результати в області класифікації тексту.

У роботі [15] досліджено ефективність інструментів машинного навчання, які можуть вирішити проблему оцінювання рівня адаптації студента до онлайн-навчання. Ці засоби містять інтелектуальні методи та моделі, враховуючи методи класифікації та нейронні мережі. Найкращими виявились модель випадкового лісу і послідовна нейронна мережа з результатами 88 і 91 % точності. У роботі [13] автори розглянули проблему класифікації ймовірності банкрутства компанії на підставі економічних і фінансових показників. У роботі [10] описано розв'язання задачі інтелектуального аналізу даних за допомогою нейронних методів структури моделі послідовних геометричних перетворень (NLS SGTM).

Отже, за результатами виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – набула подальшого розвитку модель класифікації даних, яка ґрунтується на використанні ітеративного методу зважування, на тренуванні мультиплікативного наївного класифікатора Байєса та найвагоміших показників із вхідного набору даних, що дало змогу підвищити її точність.

Практична значущість результатів дослідження – розроблено алгоритми, програмні засоби, які можна застосувати в аналізі даних будь-яких підприємств чи установ для фінансових працівників, а також відділів аналізу даних, особливо інженерів з машинного навчання для підвищення продуктивності їх роботи і вироблення консолідованої точки збору, що потребує попереднього машинного оброблення.

Висновки / Conclusions

У проведеному дослідженні здійснено порівняння ефективності застосування моделі мультиплікативного наївного класифікатора Байєса, а також класифікатора випадкового лісу і наївного байєсового класифікатора Гауса для розв'язання задачі класифікації. За результатами дослідження визначено, що для цієї задачі найбільш перспективним для застосування є мультиплікативний наївний класифікатор Байєса. На основі наведених вище досліджень набула подальшого розвитку модель класифікації даних.

Розроблено структурну схему та вказано основні модулі системи. Розроблено алгоритмічне забезпечення, яке використано у цій системі, а також розроблено програмне та інформаційне забезпечення.

В експериментальній частині проведено ручне тестування різних тест-сценаріїв, які використовують різні алгоритми формування навчальної вибірки для обраного алгоритму.

Проведено навчання розробленої моделі на прикладі розв'язання задачі класифікації даних респондентів на підставі певного предиката. Цим предикатом у наборі даних є колонка, яка ідентифікує, чи взяв респондент депозит на підставі певних критеріїв. Основними критеріями обрано вік, наявність позики, фінансовий баланс із набору даних.

Також, як засіб підвищення точності класифікації, застосовано процедуру зважування даних з використанням RIM-алгоритму для виділення тих показників, які мають вищий пріоритет для навчання моделі. Наведені результати експериментів доводять ефективність застосування запропонованого методу зважування для цієї задачі, оскільки дали змогу підвищити точність класифікації на 1-4 % для різних сценаріїв.

References

1. Angular. (2023). The web development framework for building the future. URL: https://angular.io/
2. Aridas, C. K., Karlos, S., Kanas, V. G., Fazakis, N., & Kotsiantis, S. B. (2020). Uncertainty Based Under-Sampling for Learning Naive Bayes Classifiers Under Imbalanced Data Sets. IEEE *Access*, 8, 2122–2133. https://doi.org/10.1109/ACCESS.2019.2961784
3. Bishop, C. (2006). Pattern recognition and machine learning. Dialektika-Williams, 124.
4. Docker. (2023). URL: https://www.docker.com/
5. Harrington, P. (2012). Machine learning in action. Manning Publications, 78.
6. Jia, L., Wang, Z., Lv, S., & Xu, Z. (2022). PE_DIM: An Efficient Probabilistic Ensemble Classification Algorithm for Diabetes Handling Class Imbalance Missing Values. IEEE *Access,* 10, 107459–107476. https://doi.org/10.1109/ACCESS.2022.3212067
7. Kaggle. (2023). URL: https://kaggle.com/datasets/rouseguy/bankbalanced.

8. Kim, S., Han, K., Rim, H., & Myaeng, S. H. (2006). Some Effective Techniques for Naive Bayes Text Classification. *Transactions on Knowledge and Data Engineering*, 18(11), 1457–1466. https://doi.org/10.1109/TKDE.2006.180
9. Li, J. P., Haq, A. U., Din, S. U., Khan, J., Khan, A., & Saboor, A. (2020). Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare. IEEE *Access*, 8, 107562–107582. https://doi.org/10.1109/ACCESS.2020.3001149
10. Luengo, D., Subbotin, S. (Eds.), & Doroshenko, A. (2019). Application of global optimization methods to increase the accuracy of classification in the data mining tasks. *Computer Modeling and Intelligent Systems*. Proc. 2-nd Int. Conf. CMIS-2019, Vol. 2353: Main Conference Zaporizhzhia, Ukraine, 98–109. CEUR-WS.org. URL: https://ceur-ws.org/Vol-2353/
11. PostgreSQL. (2023). The World's Most Advanced Open Source Relational Database. 25th May 2023: PostgreSQL 16 Beta 1 Released! URL: https://www.postgresql.org/
12. Python. (2023). Python is a programming language that lets you work quickly and integrate systems more effectively. URL: https://www.python.org/
13. Savchuk, D., & Doroshenko, A. (2021). "Investigation of machine learning classification methods effectiveness." In: 2021 IEEE 16th I*nternational Conference on Computer Sciences and Information Technologies* (CSIT), LVIV, Ukraine, 33–37, https://doi.org/10.1109/CSIT52700.2021.9648582
14. Scikit-learn. (2023). Machine Learning in Python. URL: https://scikit-learn.org/stable/
15. Teslyuk, V., Doroshenko, A., & Savchuk, D. (2023). Intelligent Methods and Models for Assessing Level of Student Adaptation to Online Learning. CEUR *Workshop Proceedings*, Vol. 3387, Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems, Vol. I: Machine Learning Workshop, Kharkiv, Ukraine, 331–343.
16. Theobald, O. (2017). Machine Learning for absolute beginners: Python for Data Science, Book 3. Scatterplot Press LTD, 168. URL: https://www.amazon.com/Machine-Learning-Absolute-Beginners-Science/dp/B09QBPJZ9D
17. Tiangolo / FastAPI. (2023). FastAPI framework, high performance, easy to learn, fast to code, ready for production. URL: https://fastapi.tiangolo.com/lo/
18. Wang, S., Ren, J., & Bai, R. (2020). A Regularized Attribute Weighting Framework for Naive Bayes. IEEE *Access*, 8, 225639–225649. https://doi.org/10.1109/ACCESS.2020.3044946
19. Yu, L., Gan, S., Chen, Y., & He, M. (2020). Correlation-Based Weight Adjusted Naive Bayes. IEEE *Access*, 8, 51377–51387. https://doi.org/10.1109/ACCESS.2020.2973331

V. V. Petryna, A. V. Doroshenko, R. V. Sydorenko, V. M. Teslyuk

Lviv Polytechnic National University, Lviv, Ukraine

MODEL AND MEANS OF DATA COLLECTION AND PROCESSING USING MACHINE LEARNING

The influence of the iterative method of weighting the respondents' data based on certain factors on the accuracy of the machine learning model in solving classification tasks was studied. Data collection and processing is a critical step in the process of developing and using machine learning models, after the quality and visibility of the data have a direct impact on the accuracy and effectiveness of the models. Software for working with machine learning models is developed. An analysis of the mathematical support of algorithms of classification models was carried out. A review of literary sources and related to the topic of the article was carried out. Data sets available on the network for solving classification problems were analysed. Classification models such as naive Bayesian classifier, random forest classifier, Gaussian naive Bayesian classifier, as well as an iterative method of data weighting were used. These models are integrated into the software developed for data processing, preparation, and storage. Preliminary preparation of input data for training and testing of selected models performed. Research was carried out using pre-prepared data using software according to the defined scenarios. According to the results of the research, a positive trend was found for the quality of model training with correct data preparation and selection of appropriate variables for weighting the respondents' data. Indicators of efficiency and accuracy of algorithm training show positive dynamics and are comparable with the results of testing models without using data weighting. The results of the study confirm the significant impact of the iterative data weighting method on the results of learning, training, and testing machine learning models, namely the multiplicative Bayesian classifier.

Keywords: data processing; algorithms; naive Bayesian classifier; random forest classifier; Gaussian naive Bayesian classifier; classification; data analysis; model evaluation metrics; data weighting.