**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет прикладної математики та інформатики

(повне найменування назва факультету)

Кафедра дискретного аналізу та інтелектуальних систем

(повна назва кафедри)

Дипломна робота

«Порівняльний аналіз методів класифікації даних»

Виконала: студентка IV курсу, групи ПМІ-43 спеціальності

122 Комп'ютерні науки та інформаційні технології

(шифр і назва спеціальності)

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Бень Х. Б. |  |
| Керівник | (підпис) | (прізвище та ініціали)  Квасниця Г. А. |  |
|  | (підпис) | (прізвище та ініціали) |  |

Львів – 2024

Зміст

[Вступ 3](#_Toc168834349)

[Розділ 1. Теоретичні основи машинного навчання 5](#_Toc168834350)

[1.1 Означення основних понять 5](#_Toc168834351)

[1.2 Види класифікації 9](#_Toc168834352)

[1.3 Процес класифікації 12](#_Toc168834353)

[1.4 Оцінювання класифікації 14](#_Toc168834354)

[Розділ 2. Метод дерева рішень 18](#_Toc168834355)

[2.1 Теоретичні відомості 18](#_Toc168834356)

[2.2 Алгоритм побудови ID3 21](#_Toc168834357)

[2.3 Алгоритм побудови C45 23](#_Toc168834358)

[2.4 Алгоритм побудови CART 24](#_Toc168834359)

[2.5 Алгоритм побудови Chaid 25](#_Toc168834360)

[2.6 Алгоритм побудови Random Forest 26](#_Toc168834361)

[Розділ 3. Опис результатів 27](#_Toc168834362)

[3.1 Вибірки даних 27](#_Toc168834363)

[3.2 Аналіз метрик 29](#_Toc168834364)

[3.3 Візуалізація за допомогою Confusion Matrix 32](#_Toc168834365)

[3.4 Візуалізація за допомогою POC - кривої 33](#_Toc168834366)

[Висновок 34](#_Toc168834367)

[Джерела 36](#_Toc168834368)

Вступ

У наш час наплив інформації завдяки можливостям її зберігання та збирання став, водночас, винятковою цінністю і великою складністю. Велика кількість інформації, можливість її доступного використання відкрили багато нових можливостей та дали поштовх для розвитку в різних сферах. Проте це створило таку проблему, як невміння користуватись даними правильно. Такі обсяги інформації часто видаються складними для інтерпретації. Щоб їх коректно застосовувати потрібно вміти систематизувати для подальшого аналізу. Тому, необхідно вивчати різні способи їхнього впорядкування.

Актуальність теми з практичної точки зору полягає в тому, щоби досліджувати можливості класифікації даних та застосовувати для різних цілей та прикладів.

Зважаючи на вказане, вище обрана тема є актуальною у практичному контексті

Питання класифікації у своїх роботах досліджувало багато вчених. Серед них:

* Френсіс Голтон (Francis Galton) - робота "Класифікація та вбудована асоціація" (Classification and Heredity)
* Рональд Фішер (Ronald Fisher) – праця "Статистичні методи для дослідження різниць та взаємозв'язків" (The Statistical Methods for Research Workers)
* Леонард Айзенберг (Leonard Eisenberg) – "Моделі класифікації та їх використання в розпізнаванні образів" (Classification Models and Their Use in Image Recognition)
* Том Мішель (Tom Mitchell): "Машинне навчання" (Machine Learning)

Оскільки не було знайдено комплексної праці із детальними поясненнями цієї теми, робота є актуальною з теоретичної точки зору.

Таким чином, метою цієї курсової роботи є дослідження загальних класифікаційних понять та особливостей на прикладі двох методів: логістичної регресії та дерева рішень.

Завдання полягає в тому, щоб виконати такі пункти:

* описати що таке класифікація
* визначити її різновиди
* дослідити алгоритми та оцінювання
* охарактеризувати методи
* вивчити метод логістичної регресії
* вивчити метод дерева рішень

Об’єктом дослідження стали методи класифікації, зокрема – логістична регресія та дерево рішень.

Предметом було визначено застосування цих методів для вступної кампанії та рейтингу успішності студентів.

Практичне застосування цих прикладів може бути використано в аналізі наступних вступних кампаній, поділі рейтингових списків або в інших прикладах.

# Розділ 1. Теоретичні основи машинного навчання

## 1.1 Означення основних понять

Для глибинного розуміння класифікаційних процесів необхідний детальний розбір принципів та теоретичних понять.

**Машинне навчання**

Машинне навчання (ML) — це галузь дослідження штучного інтелекту, пов’язана з розробкою та вивченням статистичних алгоритмів, які можуть навчатися на основі даних і узагальнювати невидимі дані, а отже виконувати завдання без чітких інструкцій.

Є 3 типи машинного навчання:

1. Навчання з учителем (Supervised Learning) – це процес створення моделі, що навчається на великій кількості прикладів для подальшого прогнозування результатів
2. Навчання без вчителя (Unsupervised Learning) – це тренування алгоритму пошуку закономірностей, що пізніше дає змогу поділяти категорії
3. Навчання з підкріпленням (Reinforcement Learning) – це процес навчання моделі через взаємодію з середовищем, і як наслідок отримує покарання за невдачі або нагороди за здобутки.

Найчастіше застосовують у таких галузях:

1) обробка мови

2) розпізнавання образів

3) аналіз даних

4) рекомендаційні системи

5) виявлення аномалій.

Найвідомішими алгоритмами МН вважаються: класифікація, кластеризація, регресія та нейронні мережі.

**Класифікація**

Класифікація – це процес розподілення елементів предметної області на конкретні підгрупи за певним алгоритмом, що формується визначальними характеристиками.

Завдання класифікації – це віднесення об’єкту до того чи іншого класу. Такий тип сортування об’єктів, що має наперед визначені класи та повинен розрізнити елементи між ними, відносять до «класифікації з учителем».

Основна мета класифікації полягає в тому, щоб створити модель, яка може точно призначити мітку або категорію новому спостереженню на основі його характеристик.

Методи умовно поділяють на дві групи.

1. Статистичні методи класифікації

* байєсівська класифікація;
* логістична регресія;
* дискримінантний аналіз

1. Методи машинного навчання для класифікації

* за допомогою дерев рішень;
* класифікація за допомогою штучних нейронних мереж;
* класифікація за допомогою алгоритмів покриття;
* класифікація методом опорних векторів;
* класифікація за допомогою методу k-найближчих сусідів;
* класифікація CBR-методом [2]

**Кластеризація**

Проте не завжди ми отримуємо сформовані класи, за якими сортувати.

В такому випадку ми отримуємо задачу «класифікації без учителя». Відповідно нам потрібно сформувати класи, розділивши заданий набір об’єктів на групи за певними критеріями, що означає «здійснити кластерний аналіз».

Кластеризація – це метод аналізу даних, який використовується для групування схожих об'єктів або прикладів у наборі даних.

Основна ідея полягає в тому, щоб об'єкти, які схожі один на одного, були розташовані в одному кластері, тоді як об'єкти, які відрізняються, були розташовані у різних кластерах.

Мета кластерного аналізу : виявити закономірності та структури в наборі даних, які можуть надати розуміння базових зв’язків і асоціацій, що стане основою поділу на кластери.

Алгоритми кластерного аналізу розрізняють на такі категорії:

* Ієрархічні алгоритми (Hierarchy Algorithms)
* K-means
* Алгоритми на основі щільності (Density-based methods)
* Модельно-орієнтовані алгоритми
* Нечітка кластеризація (Fuzzy Clustering)

Кластерний аналіз застосовують у різних сферах і галузях. Ось кілька типових прикладів:

1. Економіка - сегментація ринку - розділення клієнтів на групи на основі їх купівельної поведінки, демографічних чи інших характеристик
2. Біологія та медицина - групування пацієнтів із подібними клінічними характеристиками.
3. Менеджмент – групування персоналу для подальшого вибору стратегії управління
4. Соціологія – поділ респондентів на групи

**Регресія**

Регресія – це статистичний метод, який використовується для моделювання взаємозв’язків між залежною змінною (цільовою змінною) та однією чи декількома незалежними змінними (прознозованими змінні).

Метою використання варто виокремити прогнозування – передбачення значення безперервної змінної на основі отриманого набору даних.

Можна виділити такий поділ типів регресії:

* Лінійна регресія
* Логістична регресія
* Поліноміальна регресія
* Багатофакторна регресія
* Регресія з регуляризацією

Оцінювати коректність моделі можна за такими метриками:

1. Середньоквадратична помилка (MSE): середня абсолютна різниця між прогнозованим і фактичним значеннями цільової змінної.
2. Середня абсолютна помилка (MAE): середня квадратична різниця між прогнозованим і фактичним значеннями цільової змінної.
3. Коефіцієнт детермінації (R²): Показує, яка частка варіації цільової змінної пояснюється моделлю.

Застосовують регресію часто у таких випадках:

* Оцінювання цін на продукт
* Прогнозування продажів
* Моделювання залежностей
* Виявлення факторів ризику

## 1.2 Види класифікації

Можна розрізняти такі види класифікації:

1. *за ознаками*
2. **допоміжна (штучна) класифікація** – відбувається з використанням якоїсь зовнішньої ознаки, яка не є суттєвою ознакою, але оптимальна для застосування, щоб надати порядок предметній області. Наприклад: відсортовування учнів за алфавітом – ми можемо швидко відшукати необхідне прізвище, що для деяких завдань є зручно, але це не дає нам інформативного наповнення про вміння та якості особи;
3. **природня класифікація** – це складніший варіант способу впорядкування, оскільки потребує вивчення та аналіз властивостей об’єкту. Завданням є пошук закономірностей між елементами для виокремлення їхніх властивостей. Ґрунтується на природніх зв’язках та подібностях чи характерних особливостях. Також може використовуватися для передбачення, якої категорії будуть стосуватися нові дані. Наприклад: класифікація для формування видів тварин чи рослин у біології [1].
4. *за процедурою розподілу*
5. **проста** – використовує невелику кількість категорій або класів для розподілу об'єктів. Наприклад, може бути використана бінарна класифікація, де об'єкти поділяються на два взаємовиключні класи. Це простий підхід, який дає можливість зрозуміти основний аспект або властивість об'єктів, але може бути недостатнім для більш деталізованого аналізу або розпізнавання складних взаємозв'язків. Може бути ефективною, якщо основна інформація може бути висвітлена в межах обмеженого набору класів;
6. **складна** – використовує більшу кількість категорій або класів для розподілу об'єктів. Цей підхід дозволяє враховувати більшу різноманітність характеристик і відтворювати більш деталізовану структуру або відносини між об'єктами. Складна класифікація може включати багатокласову класифікацію з численними категоріями або природну ієрархічну класифікацію, де класи організовані у вигляді дерева або ієрархії з різними рівнями. Може бути необхідною, коли варто враховувати багато різноманітних характеристик або з'ясувати багатоаспектну залежність між об'єктами.
7. *за кількістю ознак*
8. **одновимірна** – використовує лише одну ознаку або змінну для розподілу об'єктів на класи. Цей підхід простий та прямолінійний, оскільки базується на аналізі лише одного аспекту об'єктів. Наприклад, можна класифікувати студентів на основі їхнього віку, де єдиним критерієм є числове значення віку. Одновимірна класифікація може бути корисною, коли важлива лише одна особливість або коли інші ознаки не мають значення для вирішення конкретного завдання. Тобто є ефективнішою для наборів даних із простим сценарієм;
9. **багатовимірна** – використовує кілька ознак або змінних для розподілу об'єктів на класи. Цей підхід дозволяє враховувати більш широкий набір характеристик об'єктів, що надає більше гнучкості у випадках, коли важливо врахувати багато аспектів. Наприклад, можна класифікувати студентів за віком, середнім балом, кількістю годин, витрачених на вивчення і т. д. Кожна з цих ознак може бути використаною як вимір для розподілу об'єктів у багатовимірному просторі. Багатовимірна класифікація дозволяє отримати більш повну інформацію про зв'язки та структуру даних, а також може покращити точність класифікаційної моделі для оптимальнішого аналізу даних.
10. *за метою:*
11. **прогнозування** – застосовується для передбачення результату від надходження нових даних. Наприклад, прогнозування приросту попиту для певних товарів;
12. **ідентифікація** – вирізнення, якої з категорій стосується конкретний елемент за його характеристиками. Прикладом може бути ідентифікація осіб за їхніми біометричними даними;
13. **систематизація –** використовується для організації та впорядкування даних, дає можливість групувати схожі об'єкти разом, що полегшує подальше оброблення. Наприклад, формування баз даних і створення спрощеної системи пошуку елементів;
14. **виявлення аномалій –** відокремлення елементів від нормального розподілу.Може застосовуватися, щобивідокремити нормальну активність від підозрілої активності в мережі або для виявлення шахрайських операцій у банківських рахунках;
15. **пріоритезація –** необхідна для ранжування об’єктів за їхньою релевантністю. Може використовуватись для формування порядку рекомендацій веб-сторінок після запиту у пошукових системах.

## 1.3 Процес класифікації

Види наборів вхідних даних поділяють на дві підмножини: навчальну і тестову.

**Навчальна множина (або training set) –** це набір даних, який використовується для навчання моделі. Вона містить вхідні дані (приклади) та відповідні вихідні значення (цільові значення). Вихідні значення використовуються для налаштування параметрів моделі так, щоб вона могла розпізнавати патерни та робити передбачення.

**Тестова множина (або test set)** також включає вхідні дані (приклади) та відповідні вихідні значення. Вихідні значення використовуються для оцінки працездатності моделі на нових, невидимих даних.

Варто описати в якому порядку відбувається класифікація. Процес складається з двох етапів: конструювання моделі та її використання.

1. **Конструювання моделі.** На цьому етапі відбуваються підготовчі процеси для формування математичної формули чи тренувальної моделі, до яких пізніше застосовується конкретний алгоритм. Тобто ми описуємо множину визначених класів.

1. ***Збір та підготовка даних***: необхідно зібрати відповідні дані, які будуть використовуватись для класифікації. Це можуть бути числові дані, текстові описи, зображення, аудіо або будь-який інший тип даних. Потім дані підлягають попередньому обробленню, включаючи очищення, перевірку на коректність та масштабування.
2. ***Визначення та пріоритезація ознак***: необхідно вибрати набір ознак, що описують об'єкти, та визначити, які мають найбільше значення. Вибір може відбуватися або вручну, або враховуючи результати обчислення цінності інформації, що несе кожна ознака.
3. ***Вибір алгоритму*:** вибір алгоритму залежить від типу даних та завдання класифікації. Якщо нам необхідно робити вибір, враховуючи велику кількість аспектів, то краще вибирати алгоритми машинного навчання, а якщо ми будемо працювати з математичними обчисленнями – статистичного.
4. ***Тренування моделі***. Після визначення алгоритму необхідно навчити модель на тренувальному наборі даних. Тренування моделі включає пошук оптимальних параметрів алгоритму, які найкраще адаптуються до навчальних даних. Це може включати процес навчання з учителем, де дані мають мітки класів, або навчання без учителя, де дані не мають міток, і модель виявляє внутрішні закономірності або групи в даних.
5. ***Валідація моделі***. Після тренування моделі необхідно перевірити її ефективність та точність на валідаційному наборі даних. Це допомагає оцінити якість класифікації та зробити необхідні налаштування для поліпшення моделі

2. **Використання моделі**. Етап, на якому відбувається власне поділ об’єктів предметної області. Робота відбувається з новими або тестовими даними, до яких ми застосовуємо нашу натреновану модель. Рівень точності – відсоток правильно класифікованих прикладів у тестовій множині.

1. ***Тестування моделі***. Після успішної валідації модель може бути протестована на тестовому наборі даних, які модель раніше не бачила. Це дозволяє оцінити загальну здатність моделі до класифікації нових об'єктів та отримати бажаний результат.
2. ***Оцінка правильності***. Нам необхідно отримати підтвердження коректності результатів. Для цього розраховується рівень точності.
3. ***Застосування моделі***: Після успішного тестування модель може бути застосована для класифікації нових невідомих об'єктів або даних в реальному часі.

Процес класифікації може бути ітеративним, і вимагати налаштування параметрів моделі та оптимізацію для досягнення кращої точності класифікації.

## ***1.4 Оцінювання класифікації***

Є чотири типи оцінювання якості моделі класифікації:

1. **Крос-перевірка (Cross-Validation) –** використовує один початковий набір даних, який поділяється на кілька підмножин. Модель навчається на одних підмножинах, а оцінка її ефективності проводиться на інших. Цей процес повторюється кілька разів з різними комбінаціями. В результаті отримується усереднена метрика ефективності моделі. Цей спосіб варто використовувати для вибірок з великою кількістю даних, щоби отримати різні комбінації навчальної та тестової вибірок [5].
2. **Тестова множина (Test Set)** – це оцінювання, яка на початку отримує два чітко поділені набори даних: Тестова вибірка – це окремий набір даних, який не використовується під час навчання моделі для обчислення точності. Модель, яка була навчена на тренувальному наборі, використовується для класифікації об'єктів у тестовій множині.
3. **Таблиця спряженості (Confusion Matrix)** – це інструмент для оцінки точності моделі класифікації. Її елементами у головній діагоналі є кількість правильно класифікованих об'єктів для кожного класу та неправильно класифікованих – на побічній.

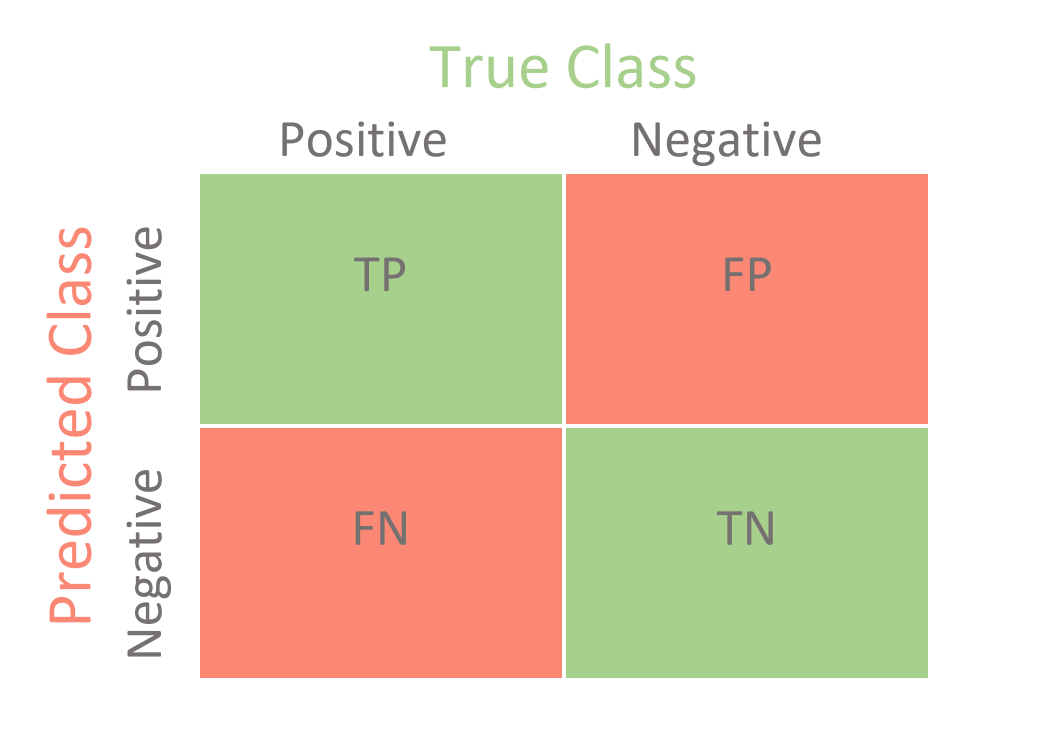


Рис. 1. Таблиця спряженості

* TP (True Positive) – кількість передбачень, де класифікатор правильно прогнозує позитивний клас (істинно-позитивні випадки);
* TN (True Negative) — кількість передбачень, що мають правильно визначений негативний клас (істинно-негативні випадки);
* FN (False Negative) – кількість передбачень, де помилково визначений негативний клас як позитивний. Це помилка 1-го роду (хибно-негативні зразки);
* FP (False Positive) – кількість передбачень, де помилково визначений позитивний клас як негативний. Це помилка 2-го роду (хибно-позитивні випадки) [англ. матриця].

Як наслідок, з цієї таблиці можна вивести такі показники, як:

* **Точність (accuracy)** – дає загальну точність моделі, тобто частку правильно класифікованих від загальної кількості зразків. Для розрахунку використовують таку формулу:

AC = (TP+TN)/(TP+TN+FP+FN)

* **Коефіцієнт неправильної класифікації (Misclassification Rate)**: повідомляє, яка частка прогнозів була неправильною. Також відомий, як помилка класифікації. Можна обчислити його за допомогою

MR = (FP+FN)/(TP+TN+FP+FN) або (1-AC)

* **Правильність (precision)**: несе в собі інформацію, яка частка прогнозів позитивного класу була насправді позитивною. Щоб обчислити точність, використовують наступну формулу:

PR = TP/(TP+FP)

* **Чутливість (recall/sensitivity)**: дає інформацію про те, яку частку всіх позитивних зразків класифікатор правильно передбачив як позитивні. Він також відомий як **істинний позитивний коефіцієнт (TPR)**. Для обчислення використовують таку формулу:

SE=TPR= TP/(TP+FN)

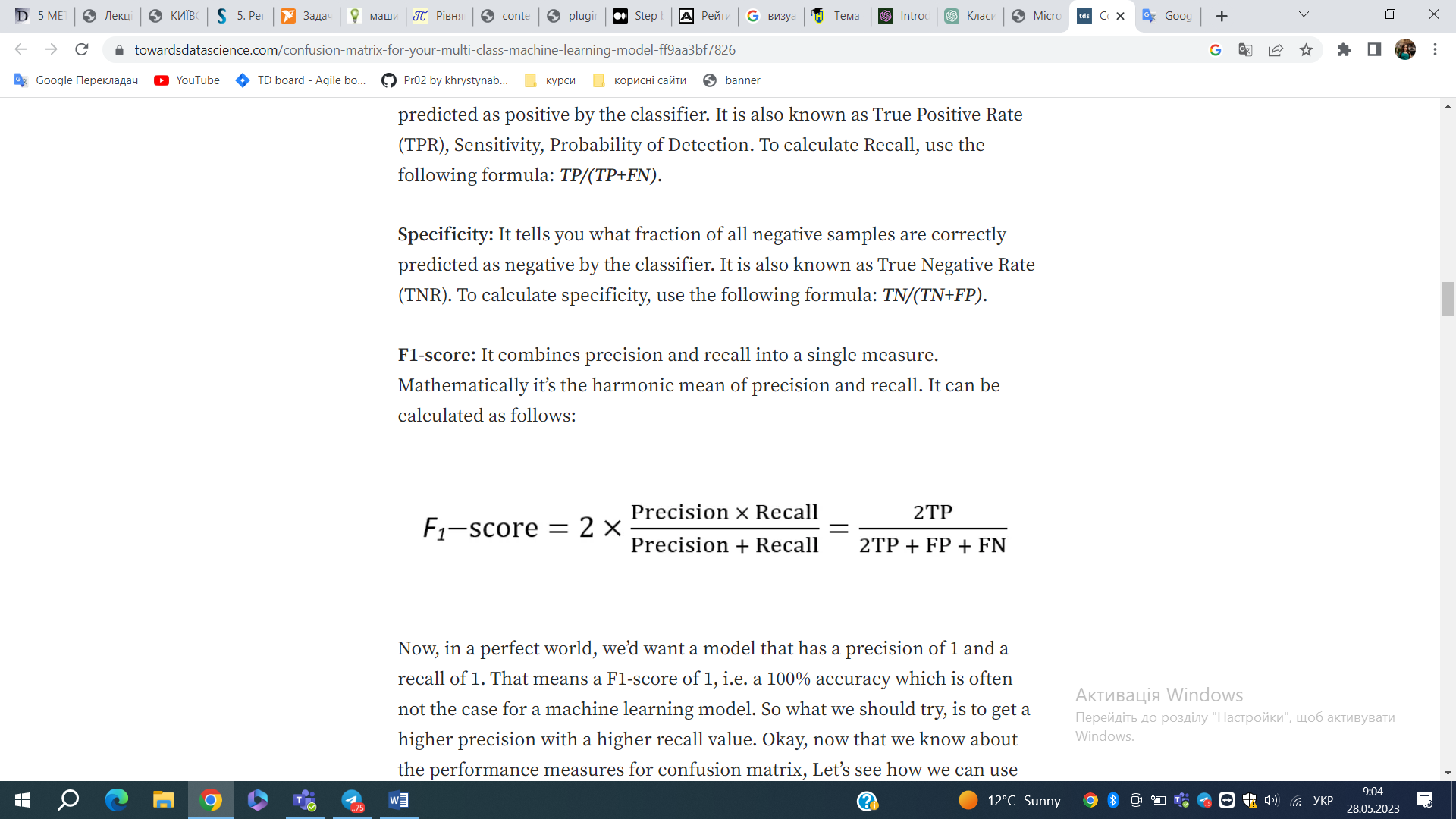
* **Хибний позитивний коефіцієнт (FPR):** частка, яка обчислюється такою формулою:

FPR = FP/(TN+FP)

* **Специфічність (specificity)** - це частка всіх негативних зразків, які класифікатор правильно передбачив як негативні. Він також має назву – справжній **негативний коефіцієнт (TNR).** Для його розрахунку маємо формулу:

SP =TNR=TN/(TN+FP)

* **F1-міра (F1-score)**: поєднує в собі точність і запам’ятовування. Математично – це середнє гармонійне значення точності та запам’ятовування. Його можна розрахувати наступним чином:



1. **ROC-аналіз (Receiver Operating Characteristic)**: відбувається завдяки аналізу результатів внаслідок побудови ROC-кривої. Він показує залежність між чутливістю (доля правильно визначених позитивних об'єктів) та специфічністю (доля правильно визначених негативних об'єктів) за зміни порогу. Крива ROC може бути використана для порівняння різних моделей та вибору оптимального порогу відсічення для заданої задачі класифікації.

ROC-крива будується як графік, де на осі X відображається специфічність, а на осі Y – чутливість. Для кожного порогового значення обчислені метрики відображаються на графіку, і відрізок з'єднує ці точки.

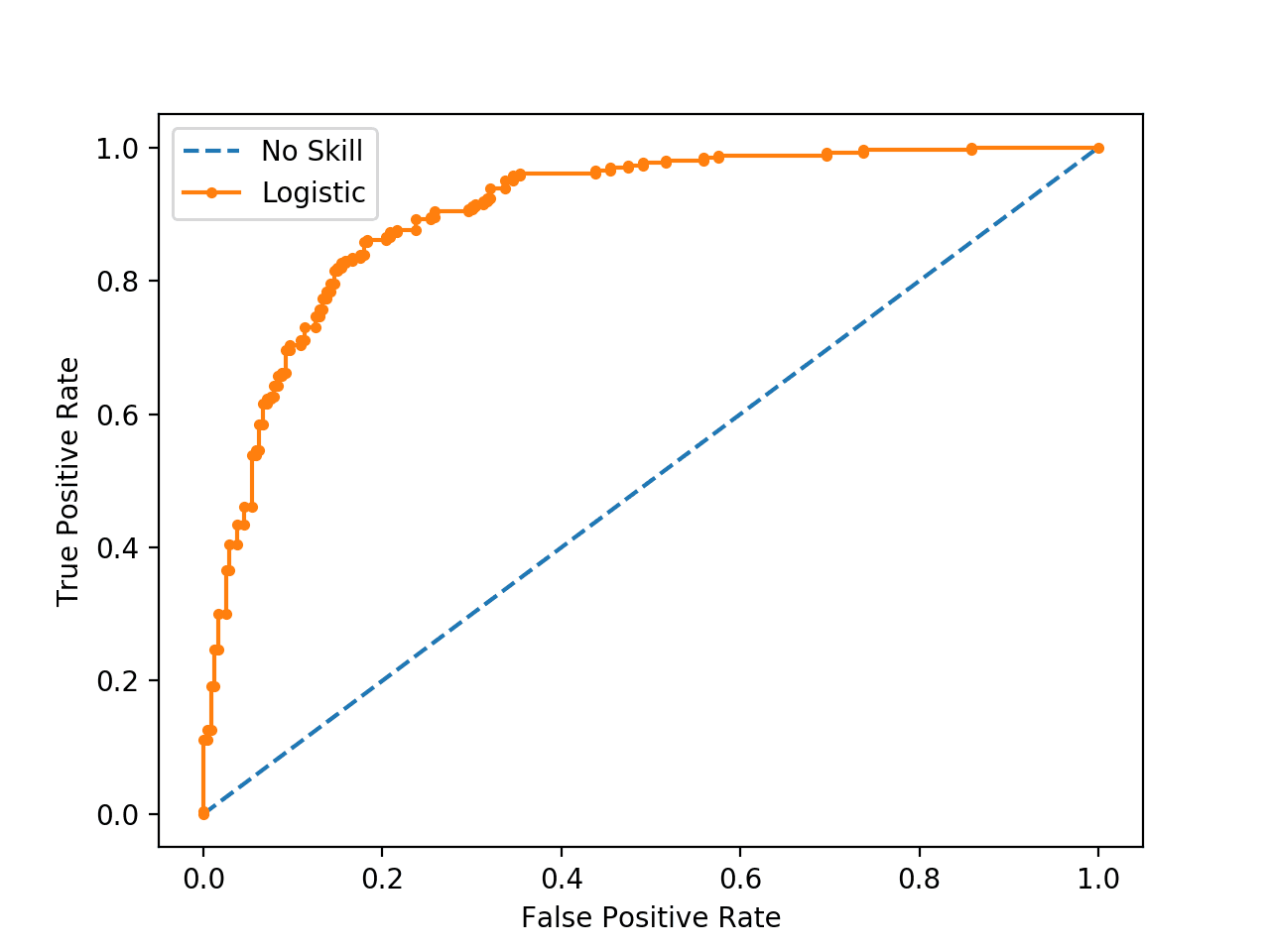
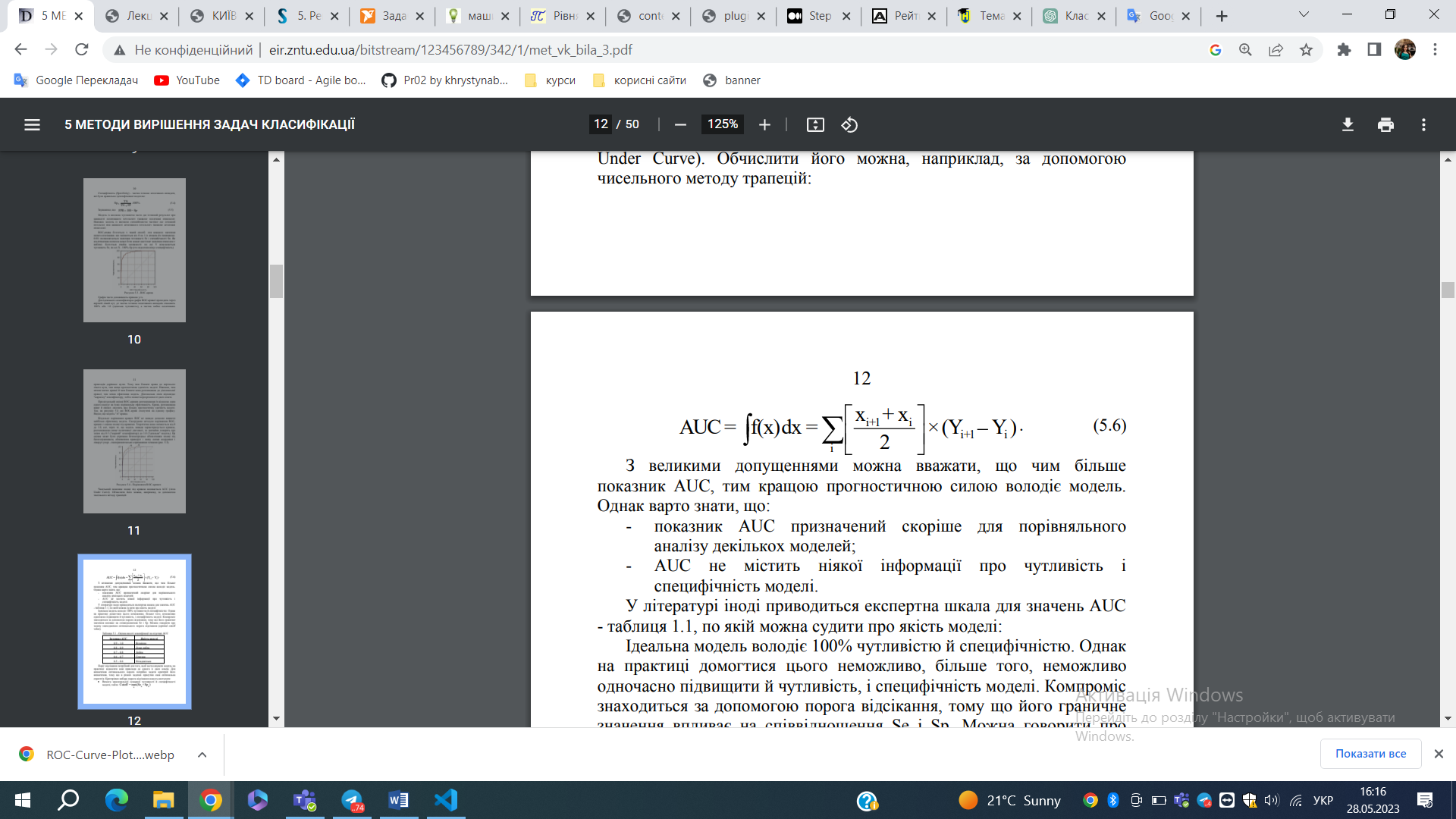


Рис. 2. ROC-крива

Точність може бути оцінена за допомогою площі під кривою (Area Under the Curve - AUC). Чим більше AUC, тим краще модель може розрізнити класи [7].



пояснення

# 

# Розділ 2. Метод дерева рішень

## 2.1 Теоретичні відомості

Дерево рішень вважається одним із найпопулярнішим алгоритмом для розв’язання задач класифікації. Його також використовують і для прогнозування – передбачення майбутніх результатів. Відмінність полягає у значенні цільової змінної, якщо вона є дискретною, то метод застосовується для задачі класифікації, якщо ж змінна безперервна, тоді для прогнозування.

Дерево рішень — це структура даних, яка використовується для прийняття рішень, базуючись на результатах моделі, сформованої внаслідок попередньо опрацьованої сукупності даних.

Основні компоненти дерева рішень:

* Вузол (Node): демонструє умову та розділення внаслідок неї.
* Кореневий вузол (Root Node): перший вузол дерева, з якого починається розділення даних.
* Листок (Leaf): знаходяться внизу дерева, представляє кінцеві рішення або прогнози.
* Гілка: вказують на можливі значення ознак.

Дерево рішень зберігає ієрархічну структуру, яка створюється набором правил, що представляють логічні конструкції «якщо: … то: …».

Одним із часткових випадків дерева рішень є бінарне дерево, що є прикладом простої структури, оскільки відповіді можуть бути лише «так», «ні».

Як результат побудови схеми у вигляді дерева можна отримати аналіз даних та спосіб пошуку відповідей на конкретні запитання. Формулювати схеми можна за допомогою дуже різних допоміжних даних, що мають прямий стосунок до поставленого запитання. Чим більше надати інформації, умов, атрибутів, тим точніше можна отримати результат, оскільки натренована модель буде мати в собі ширший обсяг опрацьованих обставин.

Процес побудови та формування структури, а також процес знаходження результатів буде описано у наступних підрозділах детально для кожного досліджуваного алгоритму.

У підсумку варто описати переваги та недоліки використання даного алгоритму:

*Переваги дерева рішень:*

1. Легко зрозуміти і інтерпретувати: Дерева рішень надають простий і зрозумілий спосіб прийняття рішень. Вони можуть бути інтерпретовані і пояснені людям без спеціальних знань в галузі машинного навчання.
2. Використовується для класифікації та регресії: Дерева рішень можуть використовуватись як для задач класифікації, так і для задач регресії. Вони можуть прогнозувати категоріальні мітки або числові значення.
3. Може працювати зі змішаними типами даних: Дерева рішень можуть обробляти набори даних, які містять як категоріальні, так і числові ознаки. Вони можуть автоматично обробляти такі типи даних без потреби в додаткових попередніх перетвореннях.

*Недоліки дерева рішень:*

1. Схильність до перенавчання: Дерева рішень можуть стати складними і перенавчатися на тренувальних даних, особливо якщо глибина дерева занадто велика або використовуються інші параметри, які сприяють складності моделі. Це може призводити до поганої універсальності і низької здатності до узагальнення на нові дані.
2. Вразливість до шуму та змін в даних: Дерева рішень можуть бути чутливими до шуму та випадкових змін в даних. Навіть невеликі зміни в тренувальних даних можуть призводити до істотних змін у структурі та правилах прийняття рішень, що може негативно впливати на результат.

Для розв’язування дерева рішення використовується багато алгоритмів, таких як:

* ID3,
* CART,
* C4.5,
* C5.0,
* NewId,
* ITrule,
* CHAID,
* CN2
* Cart
* Random forest

Спектр застосувань цього алгоритму також дуже широкий, оскільки необхідність роботи з даними зараз присутня у більшості сфер.

Відомими прикладами застовування можна назвати такі напрямки:

1. Банківська справа: сформована сукупність даних клієнтської бази, що потребує автоматизованого аналізу про надання чи відхилення кредиту
2. Медицина: діагностика захворювань різних складностей, допомога у пошуку діагнозів.
3. Біологія: пошук взаємозв’язків між біологічними ознаками для формування структури класифікації видів

Це лише частина таких можливостей, які вже використовують. Проте цей список буде суттєво розширюватись, зокрема у цій роботі буде описано як ще можна корисно застосувати такі методи.

## 2.2 Алгоритм побудови ID3

ID3 (Iterative Dichotomizer 3) – це один із найбільш вживаних алгоритмів побудови дерева рішень. Назва походить від опису процесу, оскільки алгоритм ітеративно (повторно) дихотомізує (поділяє) ознаки на дві або більше групи на кожному етапі [2].

Алгоритм був винайдено Россом Квінланом, австралійським науковцем, наприкінці 1970-х і на початку 1980-х років. Квінлан відомий вагомим внеском у сфері машинного навчання, зокрема внаслідок серії методів класифікації.

Ідея алгоритму полягає в тому, щоб знайти атрибут, який на даному етапі є найбільш корисним для класифікації набору даних. Пошук найкориснішого атрибута відбувається завдяки обрахунку приросту інформації, що залежить від значення ентропії.

Ентропія (Entropy) – це міра невпорядкованості або невизначеності в наборі даних. Називають ще «ступенем хаосу».

Це означає, що ентропія використовується для пошуку найінформативнішого атрибута, що сприяє зменшенню невизначеності. Чим менша ентропія, тим менший показник невизначеності, тобто метою є мінімізувати ентропію. Якщо ентропія дорівнює нулю, то всі елементи належать до одного класу, і ми маємо «зрозумілий» результат, якщо – одиниці, то елементи все ще хаотично розділені між різними класами.[1]

Розрахунок ентропії відбувається за такою формулою:

, де

* n = кількість класів
* A – приклад значення категорії одного із атрибутів
* = це показник ймовірності, що обраховується як частка від кількості елементів класу та загальної кількість рядків обраної категорії

Приріст інформації (Information Gain) – це показник, що відповідає за вибір атрибута, за яким буде відбуватись поділ даних. Він шукає найвагоміший атрибут, що найкраще зменшує невизначеність та класифікує цільові змінні [2].

Обрахунок відбувається за такою формулою:

, де

* m – кількість категорій
* S – атрибут, тобто риса, яку характеризують дані у стовпці

, де

* H(D) – ентропія всього датасету.

Хід роботи алгоритму:

1) обчислення загальної ентропії

2) обчислення ентропії для кожного атрибуту

3) обчислення приросту інформації

4) створення вузла дерева

5) процес рекурсії, створенння піддерев у вигляді внутрішніх вузлів

6) перевірка зупинки

7) отримання дерева із кінцевими листковими вузлами

Для передбачення результатів завдяки створеному дереву необхідно зробити такі кроки:

1. Починанаємо із коренового вузла
2. Перевірка значення атрибута
3. Рухаємось по відповідній гілці
4. Повторюємо кроки рекурсивно
5. Алгоритм закінчується, коли доходимо до конкретного результуючого класу

Отже, у даному розділі було описано особливості та роботу алгоритму ID3.

Результати, переваги, недоліки будуть описані у наступному розділі.

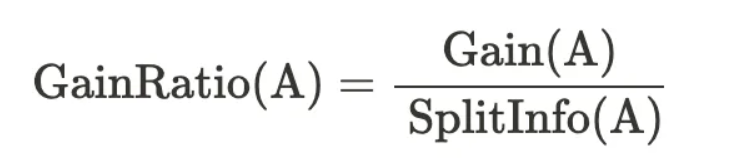
## 2.3 Алгоритм побудови C45

C45 – це покращена версія алгоритму ID3, автором якої є також Росс Квінлан.

Оскільки алгоритм базується на ID3, то зберігаються критерії поділу та такі поняття як ентропія та приріст інформації.

Проте є декілька важливих функцій, що суттєво покращують роботу цього алгоритму:

1. Обробка неперервних атрибутів – це означає, що алгоритм C45 може обробляти не тільки дискретні значення, але й неперервні. Це відбувається завдяки утворенню порогових значень після обчислення середніх значень між парами сусідів. Після чого для порогів обраховується ентропія та приріст інформації.
2. Обробка відсутніх значень – після знайдення відсутнього значення відбувається обрахунок ймовірності для заміни найдоречнішим в даному контексті значенням.
3. Коефіцієнт співвідношення приросту функції (Gain Ratio) – покращений показник для збалансування інформаційного приросту, щоб не надавати перевагу атрибутам із більшості кількістю рис, оскільки вони мають меншу ентропію.

****

1. Обрізка дерева – це покращення для уникнення перенавчання дерева, що відбувається внаслідок видалення розгалужень із несуттєвим впливом на результат, перевіряючи чи підходять умови відхилення.

Цікавий факт, що існують нові покращені версії цієї серії алгоритмів. Наприклад, відомо про C5.0, проте його деталі є не доступні публічно, оскільки Квінлан його використовує комерційно.

## 2.4 Алгоритм побудови CART

Алгоритм CART (Classification and Regression Trees) - це метод машинного навчання, який використовується як для задач класифікації, так і для регресії [4].

Алгоритм був заснований у 1984 році Лео Бріманом, Джером Фрідманом та Карл Дж. Стоуном.

Загальний принцип роботи алгоритму є такий як і у попередніх алгоритмах, проте суттєвою відмінністю є критерій поділу. Для задач класифікації це індекс Джині, а для задач регресії – MSE (Mean Squared Error).

Індекс Джині () – це показник, що демонструє наскільки добре атрибут розділяє тренувальні дані за класами. Тобто, це метрика для пошуку найкращої однорідності для підвузлів [5]. Цей показник зберігає значення суми квадратів ймовірностей для кожного класу. Нижче наведемо формулу для обчислення:

де – це ймовірність віднесення об’єкту до певного класу,

n – кількість підкласів

S – вузол, для якого відбувається обрахунок

Хід роботи:

1. Обрахунок індексу Джині для кожного з порогових значень
2. Вибір найкращого розбиття (той що мінімізує Джині Індекс)
3. Процес рекурсії
4. Зупинка при отриманні листкових вузлів
5. Обрізка дерева

## 2.5 Алгоритм побудови Chaid

CHAID (Chi-squared Automatic Interaction Detector) – це алгоритм для побудови дерева рішень, основної властивістю якого є обчислення хі-квадрату.

Значення хі-квадрат – це різниця між стандартним очікуваним сценарієм і фактичними результатами, які спостерігаються у ваших даних.

Максимальне значення хі-квадрат є найбільш статистично значущим результатом у дереві рішень CHAID. Іншими словами, це найсильніший зв’язок між двома змінними зі знайдених значень хі-квадрат. Зазвичай алгоритм використовують для категоріальних даних.

Важливою особливістю є можливість множинних розбиттів, тобто дерево може мати більше двох вузлів.

Як обчислюється хі-квадрат?

* *-* спостережена частота (фактичні значення) для категорії
* - очікувана частота для категорії
* Σ — сума по всіх категоріях

Хід алгоритму

1. Обчислюємо кількість екземплярів у атрибуті для кожного класу
2. Обчислюємо очікувані частоти
3. Шукаємо хі-квадрат
4. Порівнюємо значення χ² з критичним значенням з таблиці χ² розподілу
5. Якщо отримане значення χ² більше за критичне, ми відхиляємо нульову гіпотезу, що свідчить про наявність статистично значущої залежності між змінними.

## 2.6 Алгоритм побудови Random Forest

Random Forest (Рандомний ліс) – це алгоритм машинного навчання, який в процесі роботи формує сукупність дерев, а не одне дерево. Це і є його основною рисою та відмінністю. Назва має пряме відношення, оскільки ліс – це також множина хаотично створених дерев.

Алгоритм рандомного лісу поєднує результати кількох (випадково створених) дерев рішень для отримання кінцевого результату. Такий процес ще називають **Ensemble Learning** [6]. Зазвичай кожне дерево формується на базі алгоритму CART. Проте можливо використовувати й інші, залежно від потреб та даних, з якими відбувається робота.

Кожне дерево в ансамблі працює із окремою підмножиною елементів. Формування цих підмножин відбувається завдяки методу беггінгу.

Беггінг (Bagging) – це процес, що означає формування випадкових підмножин із початкового тренувального набору. Пізніше відбувається зіставлення результатів кожного дерева для вибору найоптимальнішого та найточнішого результату. Це можна порівняти із «голосуванням», коли рішення вибирається на користь того, що отримав найбільше голосів. Важливо зазначити, що елементи у підмножинах можуть бути обрані декілька разів, або можуть бути взагалі пропущені.

Хід роботи:

1) Застосовуння методу беггінгу – формування підмножин

2) Будуємо ансамбль дерев рішень для кожної підмножини

3) Застосовуємо метод передбачення для даних тестової підмножини

4) Визначаємо, яке найчастіше рішення було серед множини дерев

5) Отримуємо фінальне рішення – віднесення до певного класу.

# Розділ 3. Опис результатів

## 3.1 Вибірки даних

Для перевірки роботи алгоритмів, аналізу їхніх показників було використано вибірки із різних сфер, з різною кількість ознак та різною кількістю елементів.

Метою цього дослідження був пошук найоптимальніших застовувань алгоритмів класифікації, а саме: ID3, C4.5, CART, CHAID, Random Forest, для даних різного типу, щоб забезпечити різноманітні випадки, для точніших висновків.

Було обрано п'ять різних наборів даних, кожен з яких представляє характерний приклад застосування методів машинного навчання. Вибірки обирались із різних сфер, що дає можливість продемонструвати актуальність теми для використання у широких колах. Нижче наведено детальний опис кожного з використаних наборів даних:

1. Вибір, чи грати в теніс, на основі погодних умов

Дана вибірка містить інформацію про погодні умови та рішення щодо гри в теніс. Основна мета цього набору даних – передбачити, чи буде проведено гру в теніс на основі зовнішніх факторів.

* Атрибути:
* Outlook (погода): Sunny, Overcast, Rainy
* Temperature (температура): High, Mild, Cool
* Humidity (вологість): High, Normal
* Wind (вітер): Weak, Strong
* Цільова змінна:
* PlayTennis: Yes, No

Цей набір даних є класичним прикладом використання алгоритмів класифікації для прийняття рішень на основі простих факторів. Він використовується для перевірки чи правильно будується модель.

1. Присвоєння стипендії студенту, зважаючи на його бали, активність, відвідуваність

У цих даних міститься інформація про студентів та їхні академічні показники, активність та відвідуваність. Метою є запропонувати такий програмний спосіб оцінювання чи варто надавати студентові стипендію.

* Атрибути:
* Marks (бали): від 0 до 100, у вигляді A – F системи для трьох предметів
* Activity (активність): оцінка від 0 до 5 – показник додактових балів
* Attendance\_grade (відвідуваність): yes/no – допуск по відвідуваності
* Цільова змінна:
* Scholarship: Yes, No

Такий підхід допомагає зрозуміти, як різні фактори впливають на рішення щодо присвоєння стипендій. Також це дає можливість будувати рейтингові списки враховуючи більшу кількість показників, а не тільки середній бал.

1. Прогноз розлучення на основі питань про шлюбне життя

Зазначена вибірка містить інформацію про шлюбне життя пар, включаючи відповіді на низку запитань, що стосуються їхніх стосунків. Метою є передбачити, чи відбудеться розлучення. Інформація була опублікована на платформі Kaggle [!]

* Атрибути:
* 54 питання, які стосуються подружжя та їхніх взаємин, на яке дають відповіді у вигляді:

(0=Never, 1=Seldom, 2=Averagely, 3=Frequently, 4=Always).

* Цільова змінна:
* Divorce: Yes, No

Цей набір даних надає можливість вивчити взаємозв'язок між різними аспектами шлюбних стосунків і їхнім кінцевим результатом та виокремити, які аспекти стають найвагомішими. Таке дослідження може застосовуватись для психологів та їхніх спостережень.

1. Прогноз майбутнього у компанії співробітників залежно від отриманої інформації.

Це сукупність даних, яка надано HR-відділом про співробітників. Вона містить деталі про співробітників компанії, включаючи їхню освіту, історію роботи, демографічні дані та фактори, пов’язані з роботою. Метою є передбачити, чи залишиться співробітник в компанії.

* Атрибути (визначено 8 різних характеристик):
* Education (освіта): Bachelor's, Master's, PhD
* WorkHistory (історія роботи): Joining Year (Рік приєднання), Experience in Current Domain (Досвід в поточній галузі)
* Demographics (демографічні дані): Age (вік), Gender (стать), City (місто)
* JobFactors (фактори, пов'язані з роботою): Payment Tier (Рівень оплати), Ever Benched (Чи був на бенчі – тимчасово без зайнятості на проекті)
* Цільова змінна:
* Leave or Not (Чи покинув): Це цільовий стовпчик, який вказує, чи залишив співробітник компанію або ні.

Така інформація допомагає зрозуміти, які фактори впливають на утримання співробітників у компанії. Також це можливість до оптимізація роботи HR-відділу.

1. Прогноз задоволеності клієнтів авіакомпанії на основі різних параметрів

Дана сукупність даних містить інформацію про рівень задоволеності клієнтів у певній (не зазначеній) авіакомпанії. Метою є передбачити, чи будуть майбутні клієнти задоволені на основі різних параметрів, включених у вибірку.

* Атрибути (Загальна кількість ознак – 22 штуки):
* Деталі про клієнта (Тип клієнта, Вік)
* Інформація про подорож (Тип подорожі, Відстань, Клас, Затримка у відправленні чи прибутті)
* Перелік пунктів для оцінки послуг (Розташування, складений розклад, харчування та інші додаткові послуги)
* Перелік пунктів для оцінки зручностей (Онлайн-підтримка, обслуговування, користування сервісом бронювання, багаж, посадка та інші)
* Цільова змінна:
* CustomerSatisfaction (Задоволення якістю): Satisfied, Dissatisfied

Це дає нам краще розуміння про фактори, що впливають на задоволеність клієнтів авіакомпанії, та допомагає розробляти стратегії для покращення обслуговування.

## Аналіз метрик

У даному розділі було проведено порівняльний аналіз ефективності алгоритмів класифікації ID3, C4.5, CART, CHAID та Random Forest на різних вибірках даних. Для кожної вибірки було обчислено такі основні метрики:

* час виконання,
* точність,
* чутливість,
* правильність,
* специфічність та
* F1-міра.

Кожна метрика дозволяє оцінити різні аспекти роботи алгоритмів, такі як здатність правильно класифікувати дані, обробляти великі обсяги даних швидко та ефективно, а також виявляти різні типи помилок класифікації.

Вибірки даних, використані в аналізі, охоплюють різні області застосування, що дозволяє оцінити універсальність та надійність алгоритмів у різних контекстах. Зокрема, було досліджено вибірки, пов'язані з прийняттям рішень на основі погодних умов, присвоєнням стипендій студентам, прогнозуванням розлучень, утриманням співробітників у компаніях та оцінкою задоволеності клієнтів авіакомпаній (детальніше – див розділ 3.1).

Нижче наведені таблиці з результатами обчислень для кожної вибірки, що ілюструють ефективність кожного алгоритму за вказаними метриками. Ці результати допоможуть зробити висновки щодо доцільності використання кожного з алгоритмів для різних типів задач класифікації та прогнозування.

Першим показником було взято час роботи алгоритму (а саме побудови дерева), одиниця виміру якого – секунда. Цей показник є дуже суттєвим, оскільки часто це прямо пов’язано із ресурсами необхідними для роботи, що впливає на витрати для апаратного забезпечення.

У лівому стовпці подано інформацію про кількість елементів у вибірці у форматі: тренувальна вибірка/тестова вибірка. Зверху таблиці перераховано алгоритми, на яких відбувалось тестування. Нижче наведено таблицю із вказаним часом роботи для кожного алгоритму:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID3 | C45 | CART | CHAID | Forest |
| 20/5 | 0.508006 | 0.01686978 | 0.01834154 | 0.028433561 | 0.10077023 |
| 100/20 | 0.1613023 | 0.024705171 | 0.03152513 | 0.512581586 | 0.03034782 |
| 800/100 | 0.567224 | 0.04337501 | 0.43833518 | 2.766915082 | 0.15512204 |
| 4000/600 | 12.505325 | 1.21067571 | 77.9678659 | 21.0148625 | 7.14400577 |
| 20к/4к | 104.65811 | 193.297098 | 5759.99492 | 963.22899 | 1276.63264 |

Час, секунда

(табл. 1)

Проаналізувавши отримані результати, можна відзначити, що на маленьких вибірках даних найшвидше працює CART, проте на великих вибірках його швидкодія є найгіршою. Суттєво швидшими є C45 та ID3.

Наступним етапом був обрахунок точності (посилання на ф-лу). ,це найбільш узагальнений показник, який дає інформацію про коректність роботи побудованої моделі. Результати подано нижче:

Точність

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID3 | C45 | CART | CHAID | Forest |
| 20/5 | 1.0 | 1.0 | 1.0 | 0.75 | 1.0 |
| 100/20 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 800/100 | 0.75 | 0.85 | 0.85 | 1.0 | 0.9 |
| 4000/600 | 0.5770992 | 0.796946564 | 0.8045802 | 0.8167938 | 0.8174809 |
| 20к/4к | 0.44775 | 0.9035 | 0.923 | 0.77725 | 0.9375 |

(табл. 2)

Кожен із алгоритмів на вибірках із малою кількістю даних демонструє однозначну точність. Проте зі зростанням даних результати розбігаються. Random Forest має найвищу точність. Трошки менший, але все ще високий відсоток є у CART, Chaid та C45.

Під час перевірки роботи алгоритмів було обчислено ще такі показники:

* Чутливість - важлива в ситуаціях, коли важливо знайти всі позитивні випадки
* Правильність – необхідна, коли треба мінімізувати хибно позитивні випадки
* Специфічність - суттєва, коли важливо правильно виявляти негативні випадки
* F1 – оцінка - корисна, коли потрібно знайти баланс між чутливістю і правильністю. Вона особливо корисна у випадках з незбалансованими даними.

**150/20**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID3 | C45 | CART | CHAID | Forest |
| Чутливість |  |  |  |  |  |
| Правильність |  |  |  |  |  |
| Специфічність |  |  |  |  |  |
| F1-оцінка |  |  |  |  |  |

(табл. 3)

…

**4000/600 –** вибірка про працевлаштування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID3 | C45 | CART | CHAID | Forest |
| Чутливість | 0.52941176 | 0.638009049 | 0.660633484 | 0.59728506 | 0.660633484 |
| Правильність | 0.78 | 0.726804123 | 0.733668341 | 0.80981595 | 0.753478260 |
| Специфічність | 0.92396313 | 0.877880184 | 0.877880184 | 0.9285714 | 0.882442396 |
| F1-оцінка | 0.63072776 | 0.679518072 | 0.695238095 | 0.68750001 | 0.700987654 |

(табл. 4)

Для даної вибірки даних можна зробити такі висновки: чутливість та F1-оцінка мають найбільший показник у CART та Forest. Правильність та специфічність є найвищими Chaid. Зважаючи на інтерпретацію вибірки варто звернути увагу на такі показники як чутливість та специфічність. Оскільки важливо вчасно зауважити намір покинути компанію від цінного співробітника.

**20 000/4 000 –** вибірка про оцінку якості роботи в авіакомпанії

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID3 | C45 | CART | CHAID | Forest |
| Чутливість | 0.4878048 | 0.91302346 | 0.660633484 | 0.75563736 | 0.945697 |
| Правильність | 0.80181543 | 0.9096744 | 0.733668341 | 0.82017982 | 0.9487534 |
| Специфічність | 0.85659551 | 0.89217296 | 0.877880184 | 0.80295566 | 0.9392446 |
| F1-оцінка | 0.60658082 | 0.9113458 | 0.695238095 | 0.78658682 | 0.9472228 |

(табл. 5)

**…..**

**….**

**….**

**….**

**Різні розмірності для набору даних із оцінкою якості роботи авіакомпанії**

Ще одним предметом дослідження було пошук залежності між часом та точністю до кількості даних. На прикладі однієї вибірки. Результат такого спостереження дасть інформацію про масштабованість алгоритму.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID3 | C45 | CART | CHAID | Forest |
| 50/10 | 0.21571 | 0.08803 | 0.028800 | 5.1147391 | 0.0901050 |
| 500/100 | 1.722869 | 0.971092 | 2.762029 | 44.3249108 | 3.0164046 |
| 5000/1000 | 37.05520 | 22.156417 | 313.622 | 302.17766 | 130.0449 |
| 50000/1000 | 348.01483 | 722.56463 | 15год+ | 15+ | 10680.057 |

Час

(табл. 6)

У кожному алгоритмі відбувається різке зростання у часі виконання. Проте неможливо визначити узагальнену пропорційну залежність.

Точність

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID3 | C45 | CART | CHAID | Forest |
| 50/10 | 0.0 | 0.7 | 0.8 | 0.4 | 0.7 |
| 500/100 | 0.06 | 0.8 | 0.78 | 0.52 | 0.83 |
| 5000/1000 | 0.259 | 0.873 | 0.883 | 0.551 | 0.918 |
| 50000/1000 | 0.5021 | 0.921 | - | - | 0.946 |

(табл. 7)

Внаслідок отриманих результатів можна легко переконатись, що чим більше даних дається для тренування вибірки, тим точніший результат ми отримуємо на перевірці роботи моделі на тестових даних.

## Візуалізація за допомогою Confusion Matrix

Для наочної оцінки продуктивності класифікаційних моделей, було створено матриці спряженості для кожного алгоритму на різних наборах даних. Матриця спряженості наочно демонструє кількість правильних і неправильних передбачень, що дозволяє оцінити якість моделі та ідентифікувати типи помилок. Нижче наведено скріншоти результатів програми у вигляді матриць спряженості.

|  |  |
| --- | --- |
| (рис.3 – матриця ID3) | (рис.4 – матриця С45) |
| (рис.5 – матриця CART) | (рис.6 – матриця Chaid) |
| (рис.7 – матриця Random Forest) | |

## Візуалізація за допомогою ROC – кривої

Для додаткової оцінки продуктивності класифікаційних моделей також була побудована крива ROC (Receiver Operating Characteristic). Ця крива відображає співвідношення між чутливістю та специфічністю моделі при різних порогових значеннях. Вона є важливим інструментом для визначення оптимального порогу класифікації та порівняння різних моделей за їхньою здатністю правильно класифікувати дані. Нижче наведені скріншоти результатів програми у вигляді ROC кривої

|  |  |
| --- | --- |
| (рис.8 – ROC-крива ID3) | (рис.9 – ROC-крива C45) |
| (рис.10 – ROC-крива Cart) | (рис.11 – ROC-крива Chaid) |
| (рис.12 – ROC-крива Random Forest) | |

Результати ROC кривої особливі тим, що вони дозволяють оцінити ефективність класифікаційної моделі при різних порогових значеннях. Ця крива відображає залежність між чутливістю (true positive rate) та специфічністю (true negative rate) моделі. Чим ближче крива до верхнього лівого кута графіка (точка (0,1)), тим краще модель може розрізняти між позитивними і негативними класами при різних порогових значеннях.

Особливість результату полягає в тому, що крива ROC дає можливість визначити оптимальний поріг класифікації для моделі, що максимізує чутливість і специфічність одночасно.

# **Висновок**

У цій роботі було проведено дослідження та порівняння методів класифікації: логістичної регресії і дерев рішень.

Дерева рішень, з іншого боку, є гнучкими моделями, які використовують набір правил для прийняття рішень. Вони можуть обробляти як числові, так і категоріальні ознаки і мають високу інтерпретованість. Дерева рішень добре підходять для вирішення задач класифікації зі складною структурою даних і можуть бути ефективними для виявлення важливих ознак. Однак, вони можуть схильні до перенавчання, особливо при побудові глибоких дерев, і вони можуть бути вразливі до шуму та змін в даних.

Якщо завданням є прогнозування для класифікації нових даних, то дерево рішень буде більш оптимальним, оскільки приймає різні характеристики за типами даних та за суттю. Також варто зазначити, що дерево рішень має в порівнянні більшу точність у прогнозуванні.

Нижче наведено результати дослідження точності двох методів:

|  |  |  |
| --- | --- | --- |
|  | Логістична регресія | Дерево рішень |
| Предметна область про успішність | 0.5 | 1.0 |
| Предметна область про вступну кампанію | 0.67 | 1.0 |

Варто враховувати обмеження та переваги кожного методу при виборі найкращого підходу для конкретного завдання. Крім того, комбінування різних методів може привести до покращення результатів класифікації шляхом використання їх комбінованих переваг.

# Джерела

1. Задачі Data Mining. Класифікація та кластеризація [Електронний ресурс] Режим доступу до ресурсу: https://moodle.znu.edu.ua/pluginfile.php /486129/mod\_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%206.pdf
2. Інформаційні системи та технології в управлінні. [Електронний ресурс] Режим доступу до ресурсу: http://eir.zntu.edu.ua/bitstream/123456789/342/ 1/met\_vk\_bila\_3.pdf
3. Класифікація алгоритмів машинного навчання: лінійна регресія, класифікація та кластеризація [Електронний ресурс] Режим доступу до ресурсу: https://bloginnovazione.it/uk/machine-learning/3716/
4. How to Use ROC Curves and Precision-Recall Curves for Classification in Python [Електронний ресурс] Режим доступу до ресурсу: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/
5. Confusion Matrix for Your Multi-Class Machine Learning Model [Електронний ресурс] Режим доступу до ресурсу: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826
6. Методи дерев рішень, класифікації та прогнозування [Електронний ресурс] Режим доступу до ресурсу: https://moodle.znu.edu.ua/pluginfile.php?file=/486136 /mod\_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%209.pdf
7. Дерева рішень і алгоритми їх побудови [Електронний ресурс] Режим доступу до ресурсу: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/590b3734-7e5c-48cc-80ec-f1704589696b/content

Додатки: