**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**З В І Т**

**Лабораторна робота №10  
з дисципліни**

**«Комп’ютерні системи**

**штучного інтелекту»**

Виконавець:

студент групи КІ-22м Косей М.П.

Керівник:

викладач Саяпін В.Г.

2023

**Лабораторна робота №10**

**Тема:** Методи та засоби відображення розв’язків оптимізаційних задач.

**Мета:** Одержати знання роботи методів за допомогою яких можна

автоматизовано приймати рішення.

**ХІД РОБОТИ**

1. **Ознайомитись з теоретичними відомостями до лабораторної роботи**

Метод повного перебору варіантів (**brute-force**) і дерева рішень (**Decision Trees**) є двома різними підходами до вирішення оптимізаційних задач.

У таблиці 1 дано порівняння цих двох методів.

**Таблиця 1.** Порівняння методу повного перебору варіантів і дерева рішень.

| **Аспекти** | **Дерево рішень** | **Метод повного перебору варіантів** |
| --- | --- | --- |
| **Складність обчислень** | Ефективний при великій кількості можливих варіантів | Експоненціальний ріст складності зі збільшенням розміру задачі |
| **Масштабованість** | Підходить для великих задач, структуроване зменшення простору пошуку | Непрактичний для великих задач, експоненціальний ріст кількості варіантів |
| **Інтерпретованість** | Інтерпретований, можливість візуалізації та розуміння | Немає структурованої інтерпретації |
| **Ефективність** | Ефективний для швидкого прийняття рішень, особливо для великих задач | Точний результат, але обчислювально витратний для великих задач |

Дерева рішень використовується для прийняття рішень, що застосо­вуються в статистиці, аналізі даних та машинному навчанні.

Дерева рішень найбільш успішно застосовуються в наступних галузях:

* **Банківська справа:** оцінка кредитоспроможності клієнтів банку при наданні кредитів.
* **Промисловість:** контроль якості продукції (виявлення дефектів у готових товарах), безперебійні випробування (наприклад, перевірка якості зварки) та інші подібні завдання.
* **Медицина:** діагностика різних за складністю захворювань.
* **Молекулярна біологія:** аналіз структури амінокислот.
* **Торгівля:** класифікація клієнтів та товарів.

Цей інструмент допомагає вирішувати наступні задачі:

* + - 1. **Класифікація:** віднесення об'єктів до одного з наперед відомих класів. Цільова змінна повинна мати дискретні значення.
      2. **Регресія:** прогнозування числового значення незалежної змінної для заданого вхідного вектора.
      3. **Опис об'єктів:** набір правил у дереві рішень дозволяє компактно описувати об'єкти. Тому замість складних структур, що використовуються для опису об'єктів, можна зберігати дерева рішень.

Дерева рішень відносяться до типу навчання з вчителем.

У процесі навчання дерева рішень використовується набір навчальних даних, в якому кожному вхідному прикладу відповідає правильна вихідна мітка або класифікація. Модель дерева рішень навчається на цих парах вхід-вихід з метою прогнозування відповідей для нових вхідних даних.

Дерева рішень(**Decision Tree**) - це прості моделі, які мають гілки(**Branch**), вузли(**Node**) та листки(**Leaf Node**) і розбивають набір даних(**dataset**) на менші підмножини, що містять екземпляри з подібними значеннями. (див. рисунок 1).

Елементів дерева рішень:

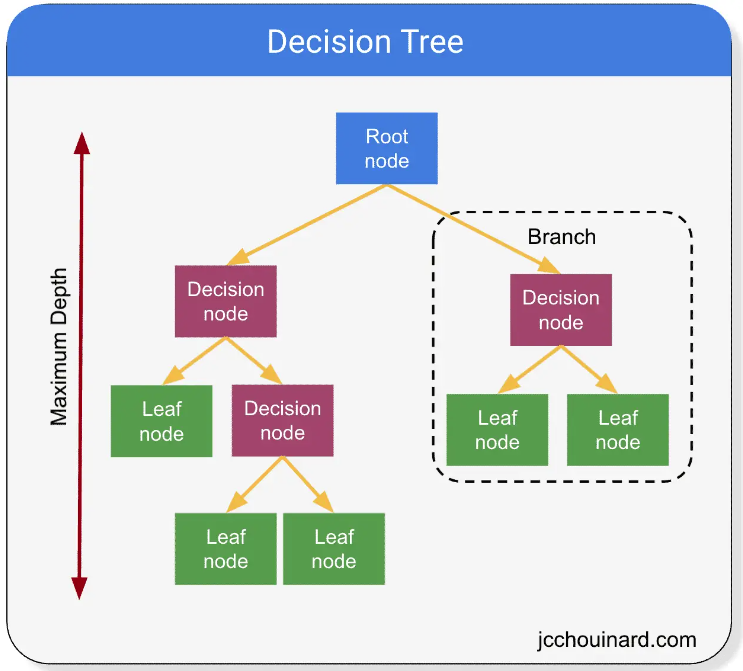
**Кореневий вузол (Root Node):** Перший вузол на шляху, з якого почина­ються всі рішення. Він не має батьківського вузла і має 2 дочірні вузли.

**Вузли рішень (Decision nodes):** Вузли, які мають 1 батьківський вузол і розгалужуються на дочірні вузли (рішення або листки).

**Листові вузли(Leaf nodes):** Вузли, які мають 1 батьківський вузол, але не поділяються далі (також відомі як термінальні вузли). Вони є вузлами, які дають прогноз або класифікацію.

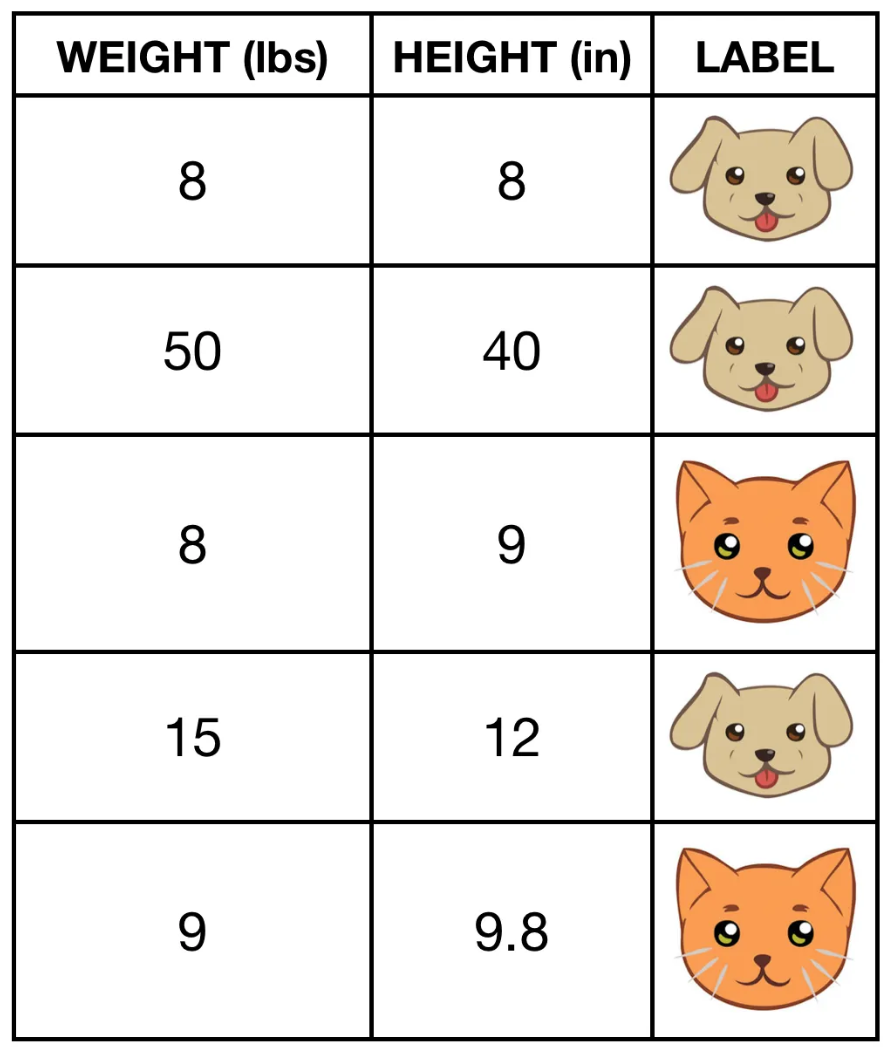
**Гілки(Branches):** Частина дерева, також відома, як піддерево.

**Максимальна глибина(Maximum depth):** Максимальна глибина дерева визначається як найбільша кількість рівнів. Це найбільша кількість гілок або рішень, які можуть бути пройдені від кореневого вузла до будь-якого листового вузла в дереві.

****

**Рис. 1** Схема та визначення елементів дерева рішень.

Розберемо які властивості мають вузли дерев та алгоритм побудови дерева рішень на прикладі дерева для класифікації тварин на два класи: коти та собаки(див. рисунок 2, <https://towardsdatascience.com/decision-tree-an-algorithm-that-works-like-the-human-brain-8bc0652f1fc6>).



**Рис. 2** [Навчальний датасет.](https://towardsdatascience.com/decision-tree-an-algorithm-that-works-like-the-human-brain-8bc0652f1fc6)

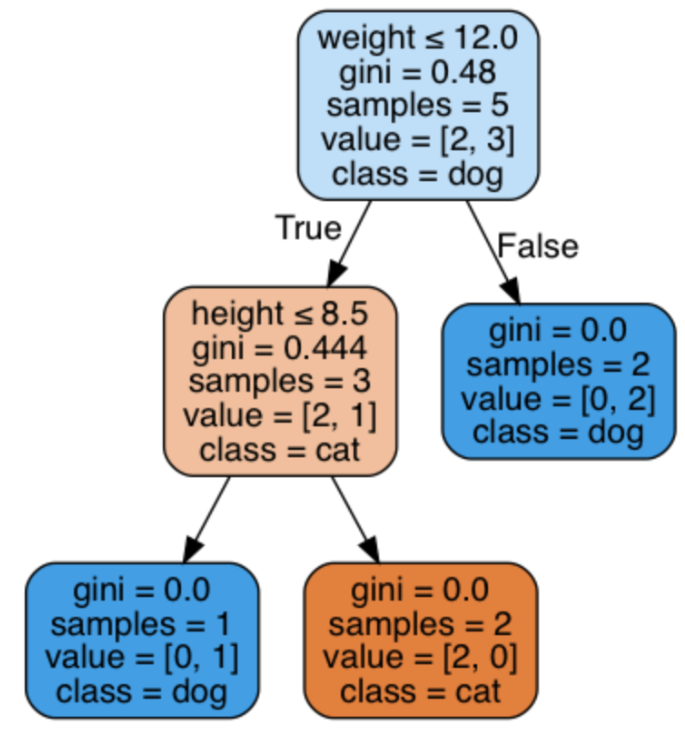
Вузол в дереві рішень має наступні властивості : (див. рисунок 3)

**Предикат:** Всі вузли, окрім листових вузлів мають асоційований предикат або умову, яка визначає, як розбити дані на підмножини. Предикат може бути умовою порівняння атрибуту з певним значенням або іншими логічними умовами (**weight <= 12, heigh<=8.5**).

**Елементи датасету:** Вузол також містить підмножину даних, які відповідають певній умові, встановленій предикатом. Ці елементи датасету використовуються для подальшої рекурсивної побудови піддерев (**samples = 5, …; values = [2,3]** **…..**).

Листовий вузол в дереві рішень має наступні властивості:

**Прогнозоване значення:** У випадку задачі класифікації це може бути конкретна мітка класу, до якого належать вхідні дані. У випадку задачі регресії це може бути числове значення, яке прогнозується моделлю **(class = dog, class = dog).**



**Рис. 3** [Приклад дерева для визначення класу тварин: кіт або собака](https://towardsdatascience.com/decision-tree-an-algorithm-that-works-like-the-human-brain-8bc0652f1fc6).

Алгоритм побудови дерева рішень може варіюватись в залежності від використовуваного методу (див. рисунки 4).

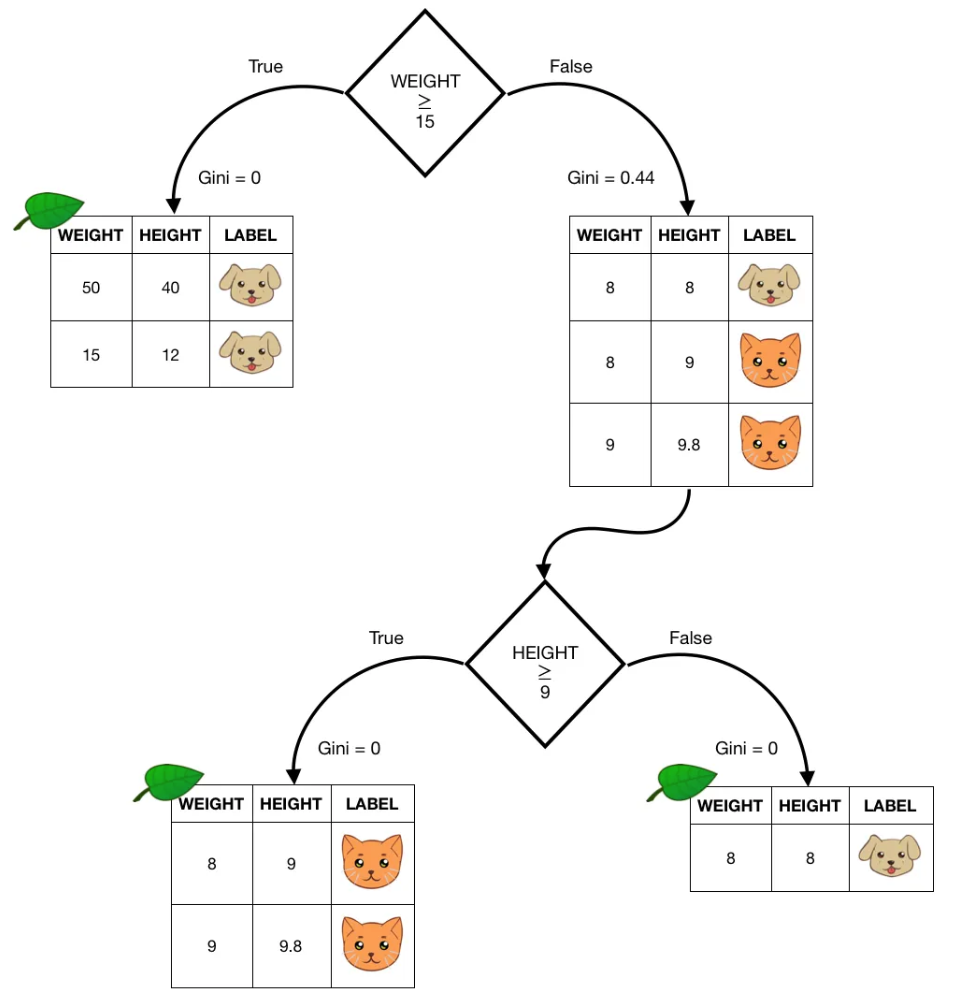
Один з найпоширеніших алгоритмів побудови дерева рішень - це алгоритм **ID3 (Iterative Dichotomiser 3)** або його вдосконалена версія, така як **C4.5** або **CART (Classification and Regression Trees).**

Основні кроки алгоритму побудови дерева наступні:

1. **Вибір кореневого вузла:** Визначення, який атрибут (признак) найкраще розділяє навчальні дані на підмножини. Використовується принцип жадібної максимізації приросту інформації. На кожному кроці алгоритм вибирає атрибут, за яким відбувається розбиття, таким чином, щоб приріст інформації(зменшення ентропії) був максимальним (індекс **Gini** для задач класифікації). Далі процедура рекурсивно повторюється, поки ентропія не стане рівною нулю або якійсь невеликій величині (якщо дерево не підлаштовується ідеально під навчальну вибірку, щоб уникнути перенавчання).
2. **Розділення навчальних даних:** Розділення навчальних даних на підмножини на основі значення вибраного атрибута. Кожна підмножина відповідає одній гілці дерева.
3. **Рекурсивна побудова піддерев:** Повторення кроків 1 і 2 для кожної підмножини даних (гілки), створених на попередньому кроці. Цей процес продовжується досягнення заданої умови зупинки, такої як досягнення максимальної глибини, вичерпання атрибутів або інших критеріїв зупинки.
4. **Призначення значень листовим вузлам:** Визначення значень листових вузлів дерева, які відповідають прогнозованим класам або числовим значенням відповідно до задачі (класифікація або регресія).

Алгоритм побудови дерева може мати варіації та додаткові етапи в залежності від конкретної реалізації та використовуваних методів.

Принципово важливим є розділення даних на основі атрибутів та визначення оптимального розбиття для кожного вузла дерева.



**Рис. 4** [Побудова дерева для визначення класу тварин: кіт або собака](https://towardsdatascience.com/decision-tree-an-algorithm-that-works-like-the-human-brain-8bc0652f1fc6).

1. **Використовуючи будь-яку мову програмування розробити додаток який буде моделювати дерево перебирання для булевого виразу.**

В попередній лабораторній роботі № 8 за варіантом 8 необхідно було запрограмувати штучну нейрону мережу з двома виходами, щоб отримати значення:

* першої функції стрілка Пірса (NOR ↓) на першому виході;
* другої функції штрих Шеффера (NAND ↑) на другому виході.

Зробимо теж саме за допомогою дерева рішень.

Аналізуючи таблицю істинності функції (**таблиця 2**) і використовуючи алгоритм побудови дерев рішень, можна зводити задачу до задачі класифікації, де дерево рішень визначає правила класифікації об'єктів на основі їх характеристик **А, В**.

Кількість класів три: **11, 01, 00.**

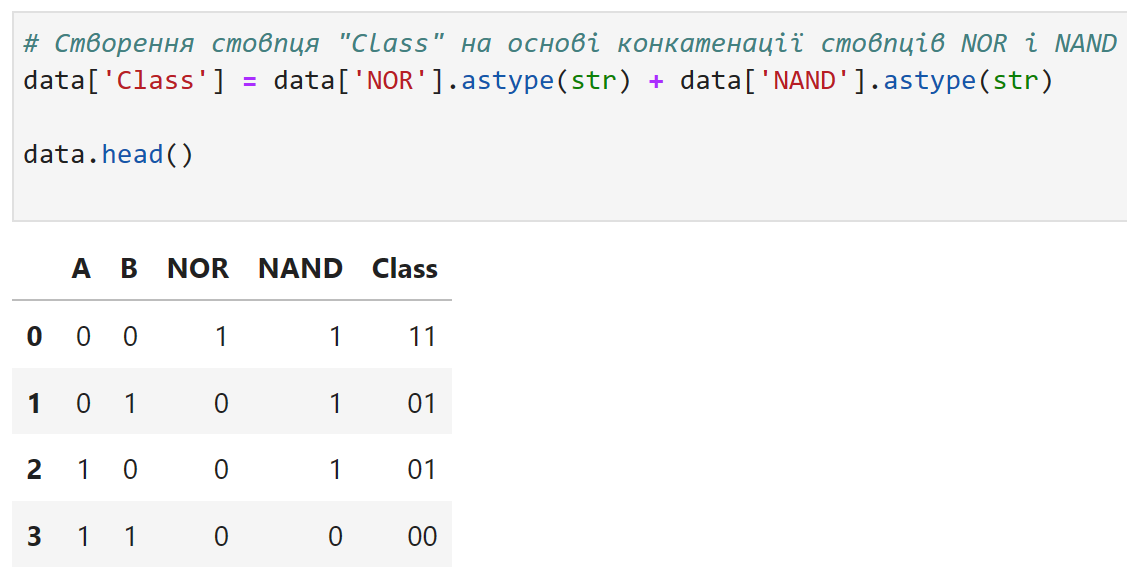
**Таблиця 2.** Таблиця істинності для булевої логічної функції (навчальний датасет).

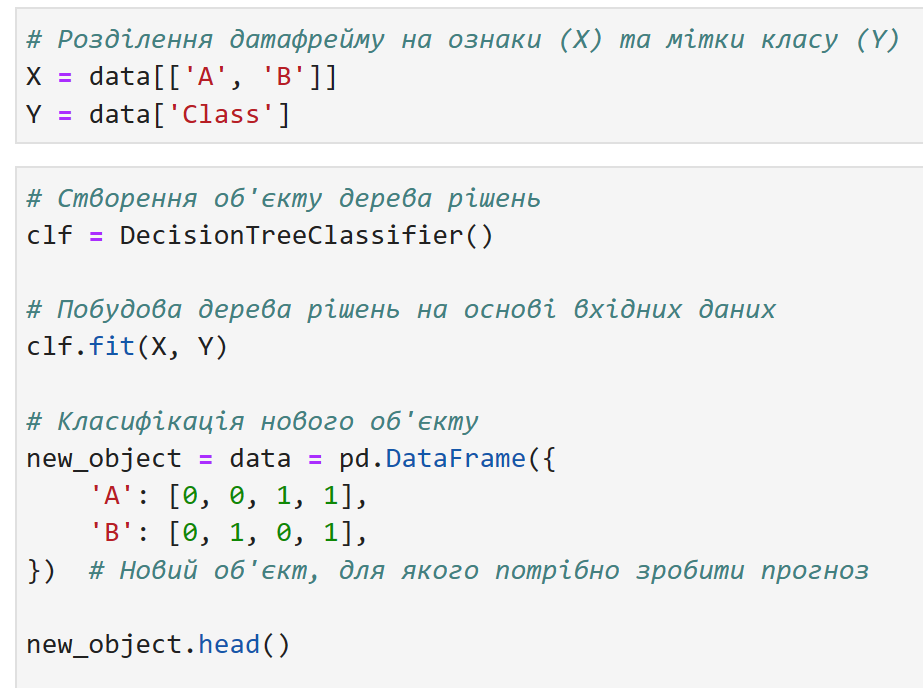
|  |  |  |  |
| --- | --- | --- | --- |
| **Вхідні змінні** | | **Вихідні змінні** | |
| **A** | **B** | **NOR ↓** | **NAND ↑** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

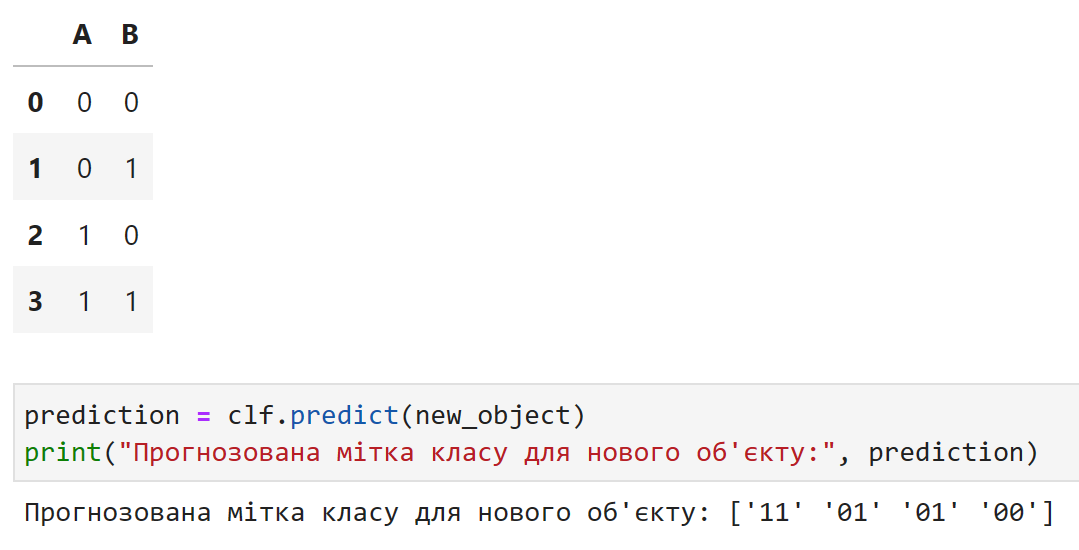
Використовуємо мову програмування **Python** та проект [**JupyterLab**](https://jupyter.org/).

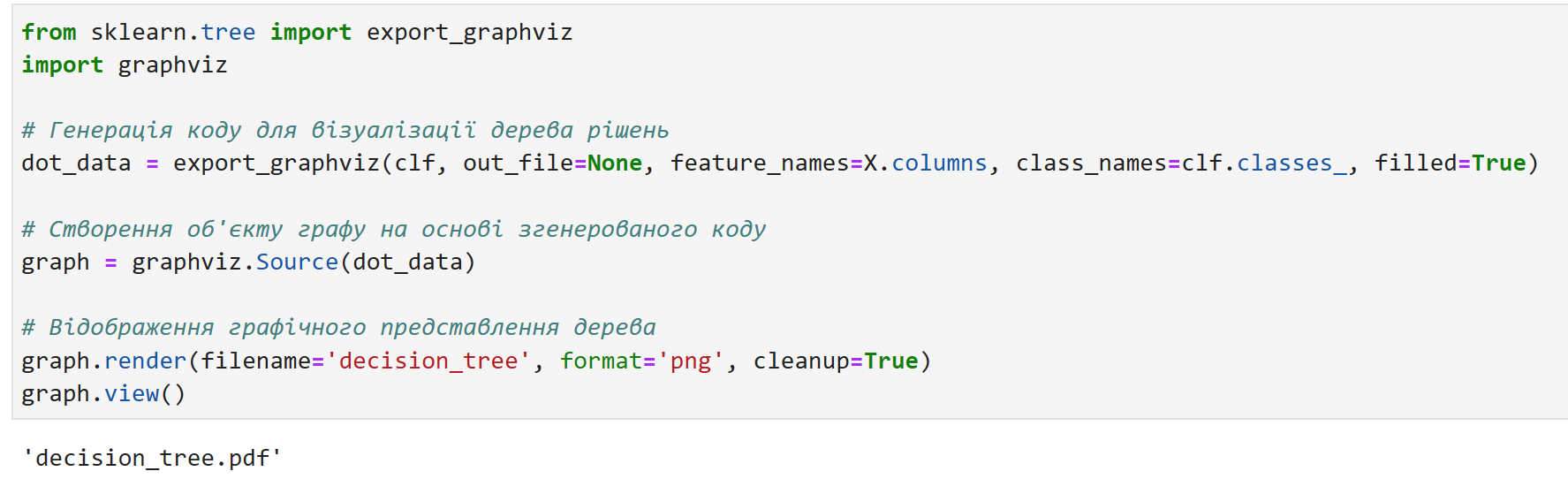


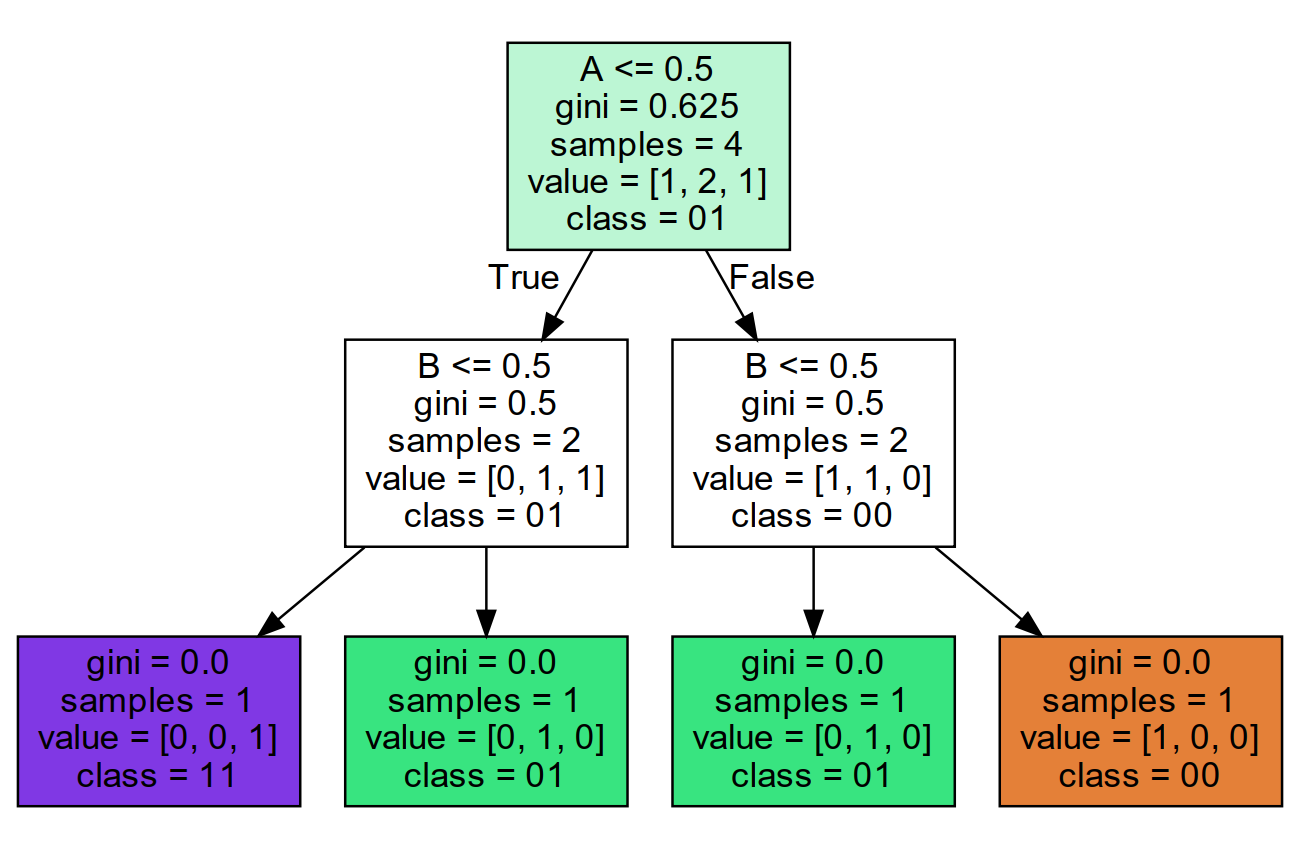






****

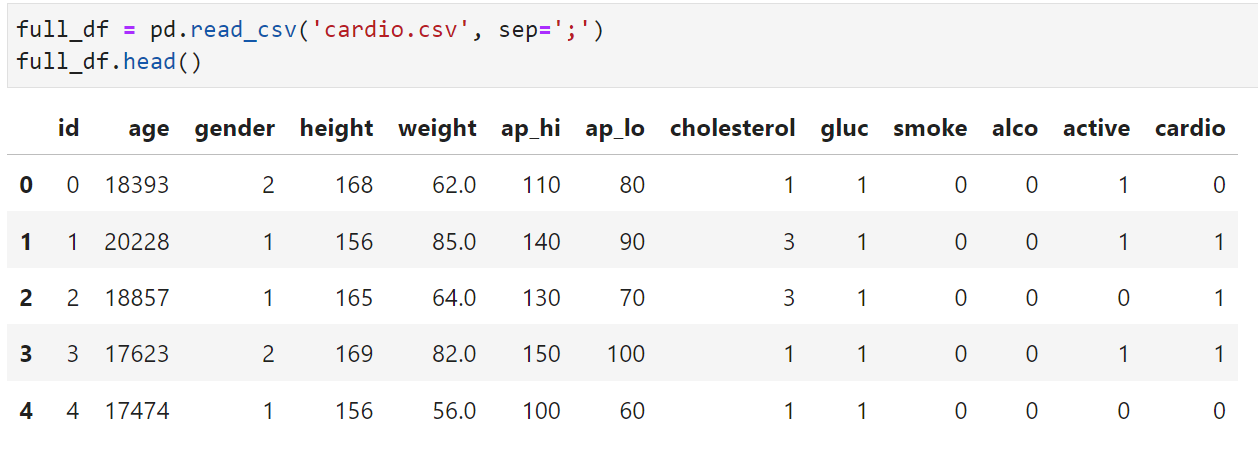




1. **Розглянемо більш складний приклад та побудуємо дерево рішень.**

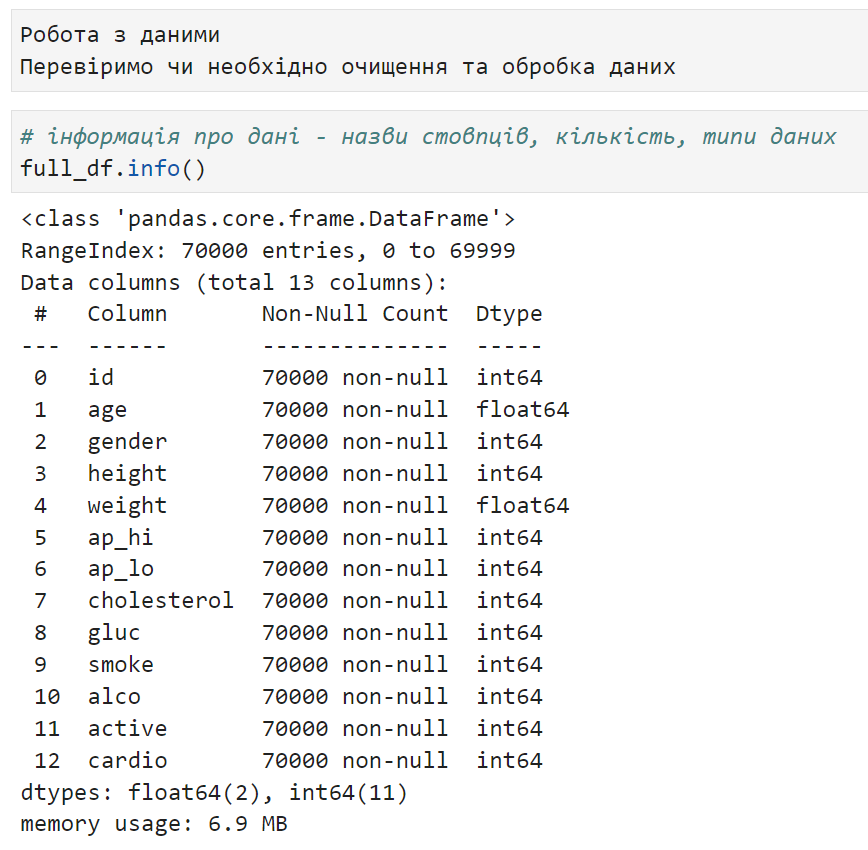
Будемо працювати з набором даних для задачі класифікації - дані про серцево-судинні захворювання [(https://www.kaggle.com/datasets/sulianova/ cardiovascular-disease-dataset?resource=download).]((https://www.kaggle.com/datasets/sulianova/%20cardiovascular-disease-dataset?resource=download).)

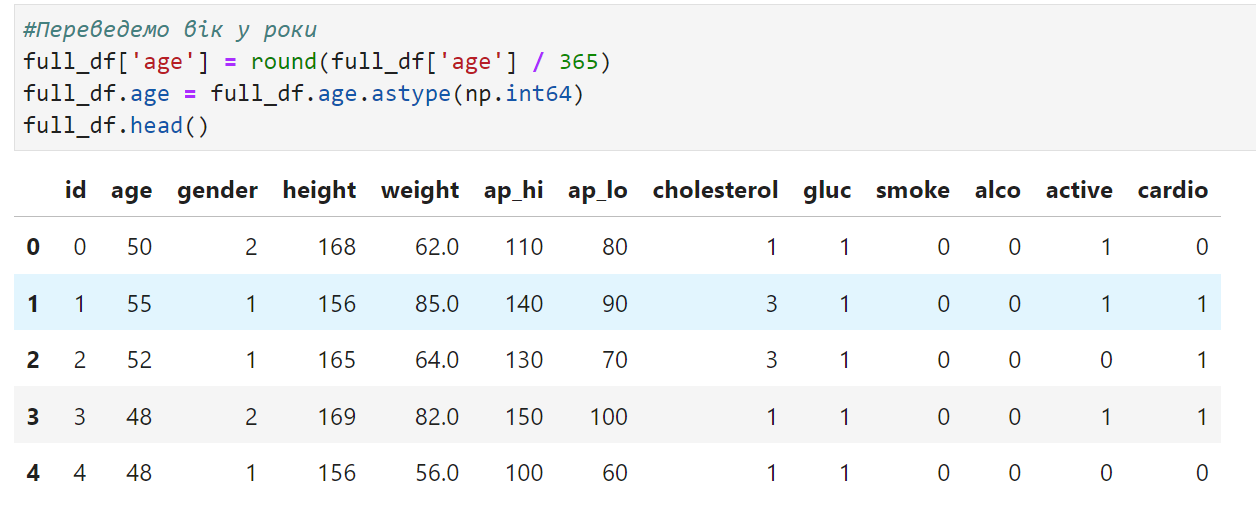
У цій задачі пропонується передбачити наявність серцево-судинних захворювань на основі результатів класичного медичного огляду.

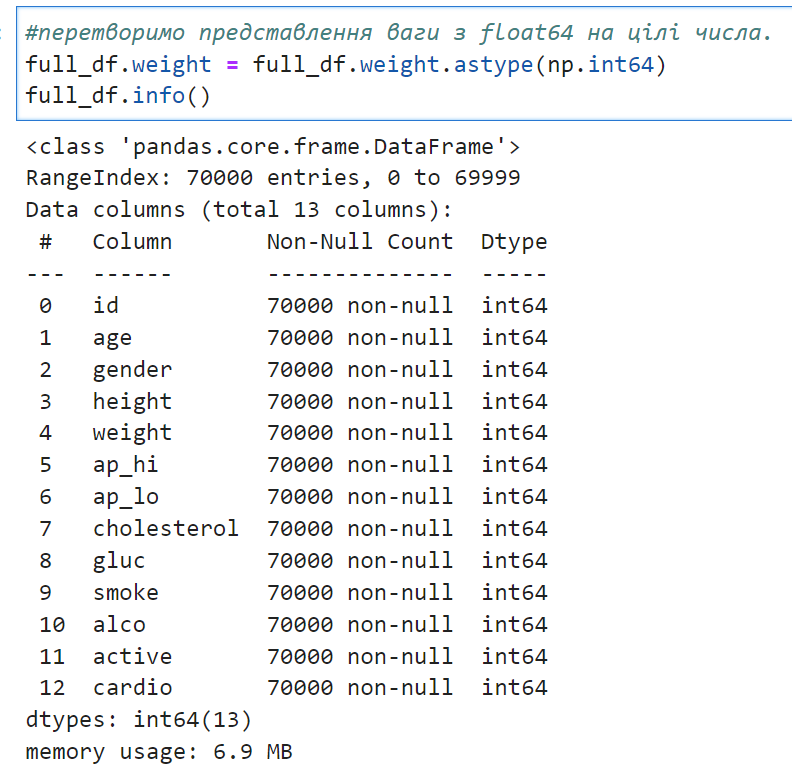


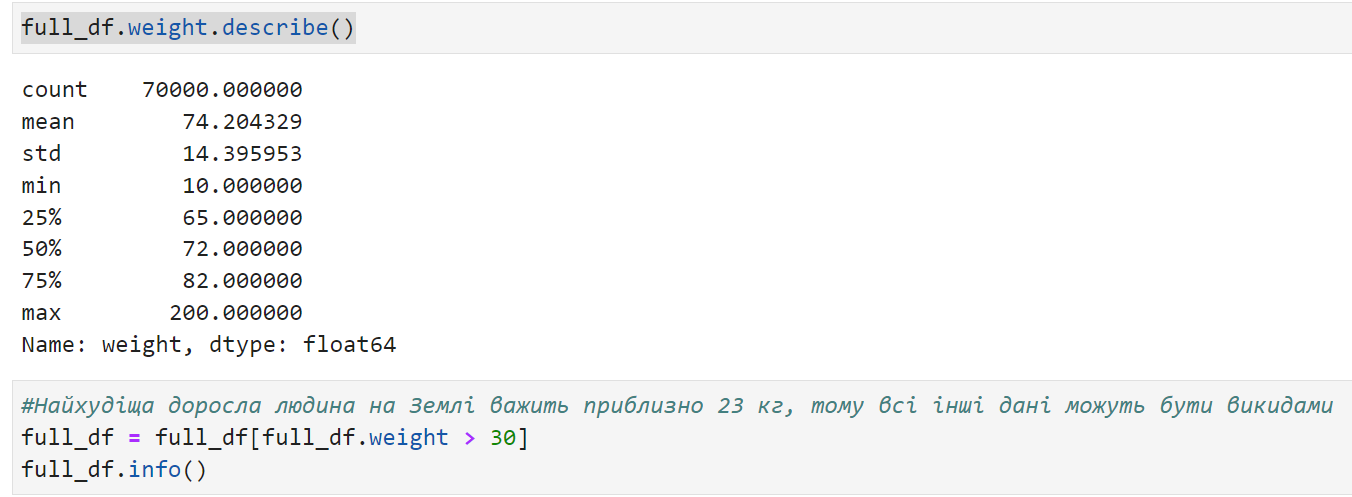
Датасет містить три групи ознак:

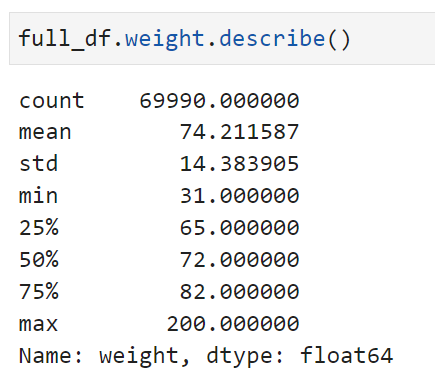
* Об'єктивні ознаки:
  + Вік (у днях);
  + Зріст;
  + Вага;
  + Стать;
* Результати вимірювання:
  + Верхній та нижній артеріальний тиск;
  + Рівень холестерину (три групи: норма, вище норми, значно вище норми);
  + Рівень глюкози (три групи: норма, вище норми, значно вище норми);
* Суб'єктивні ознаки (бінарні):
  + Куріння;
  + Споживання алкоголю;
  + Фізична активність.

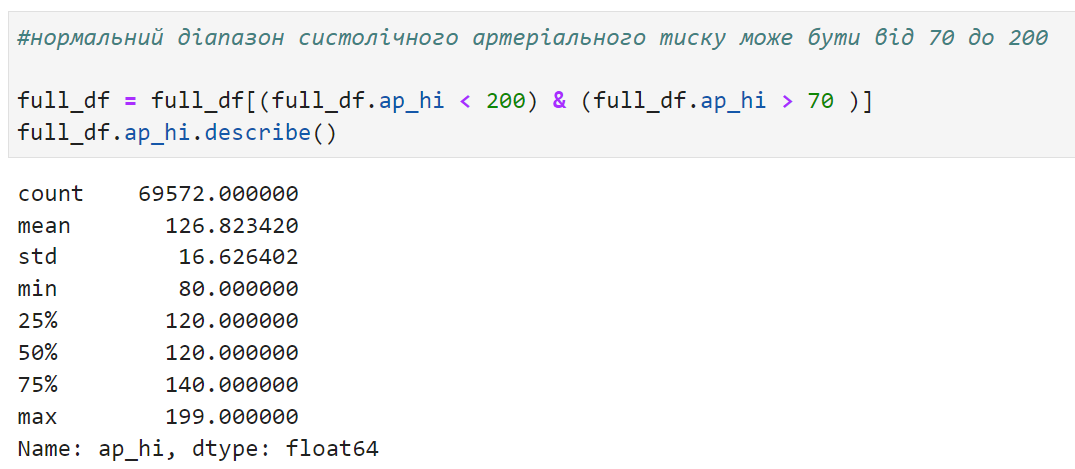


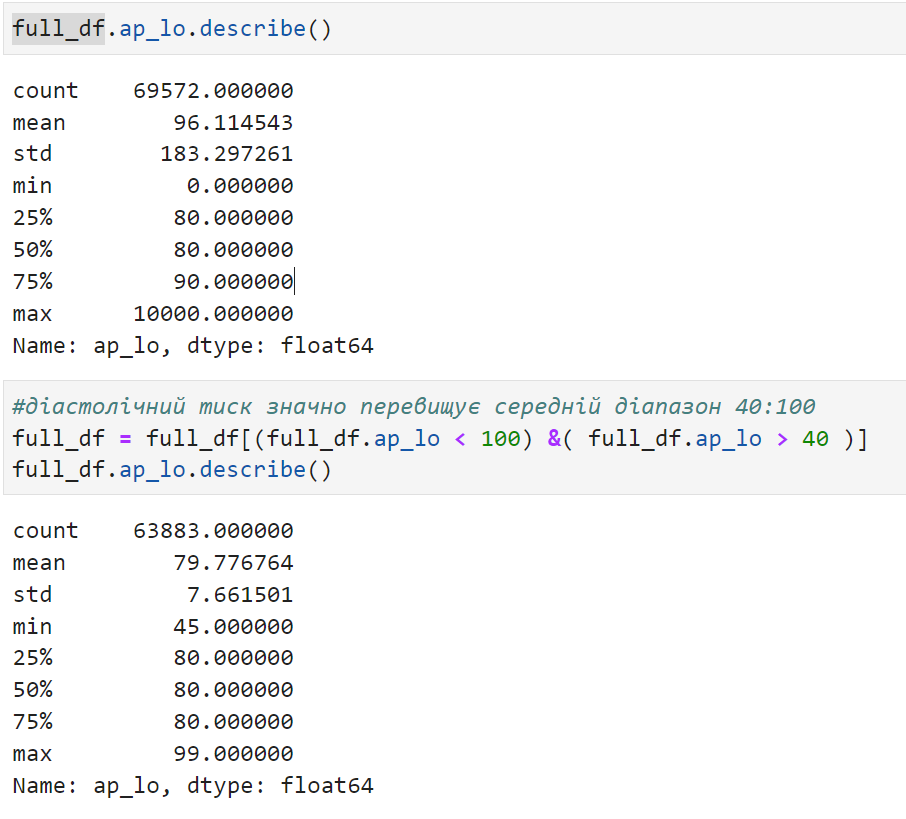




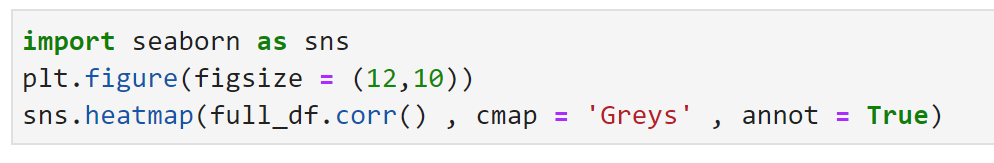


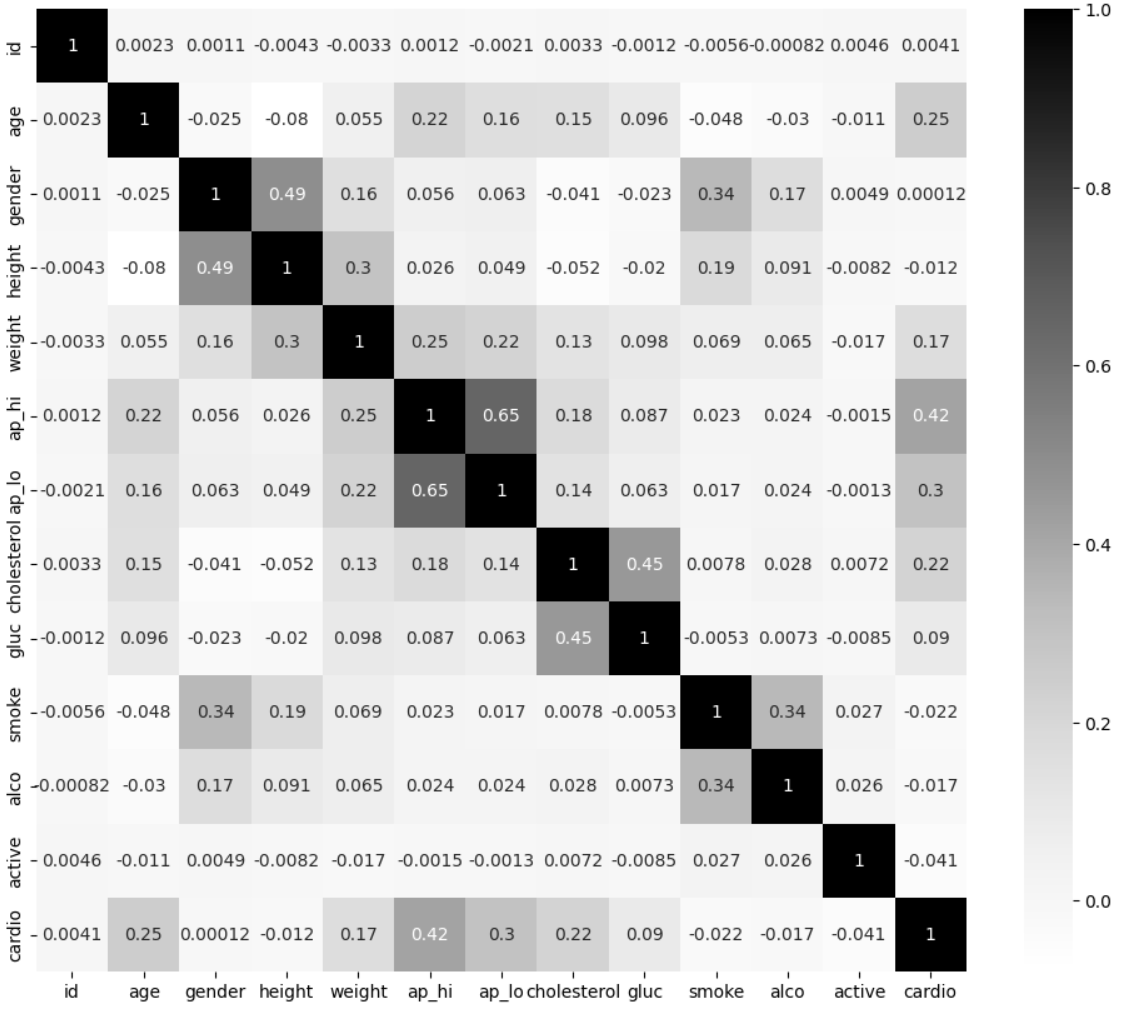


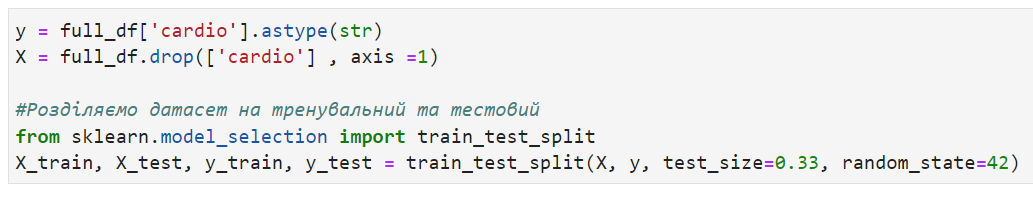


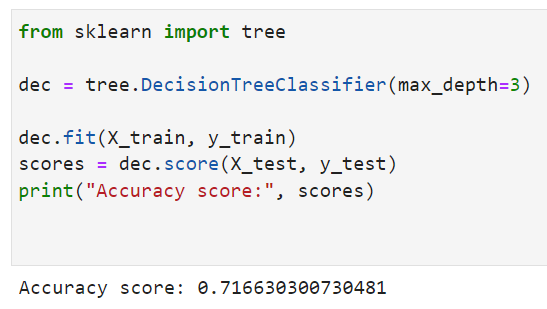


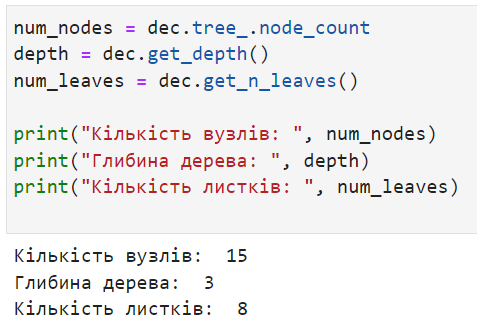
Подивимось кореляцію між ознаками

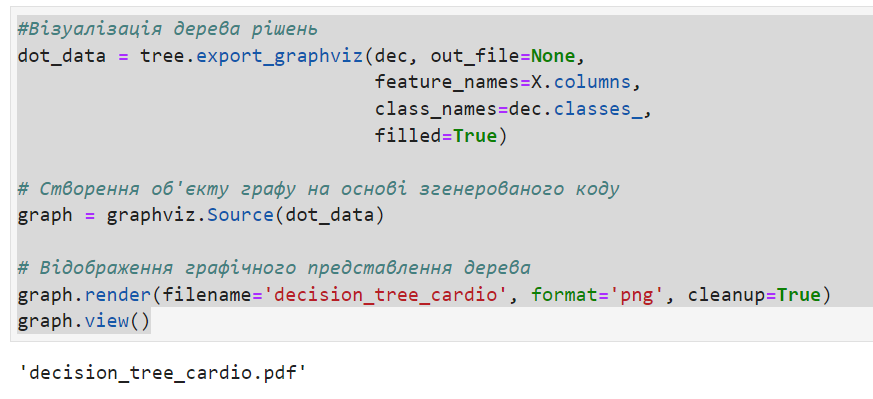


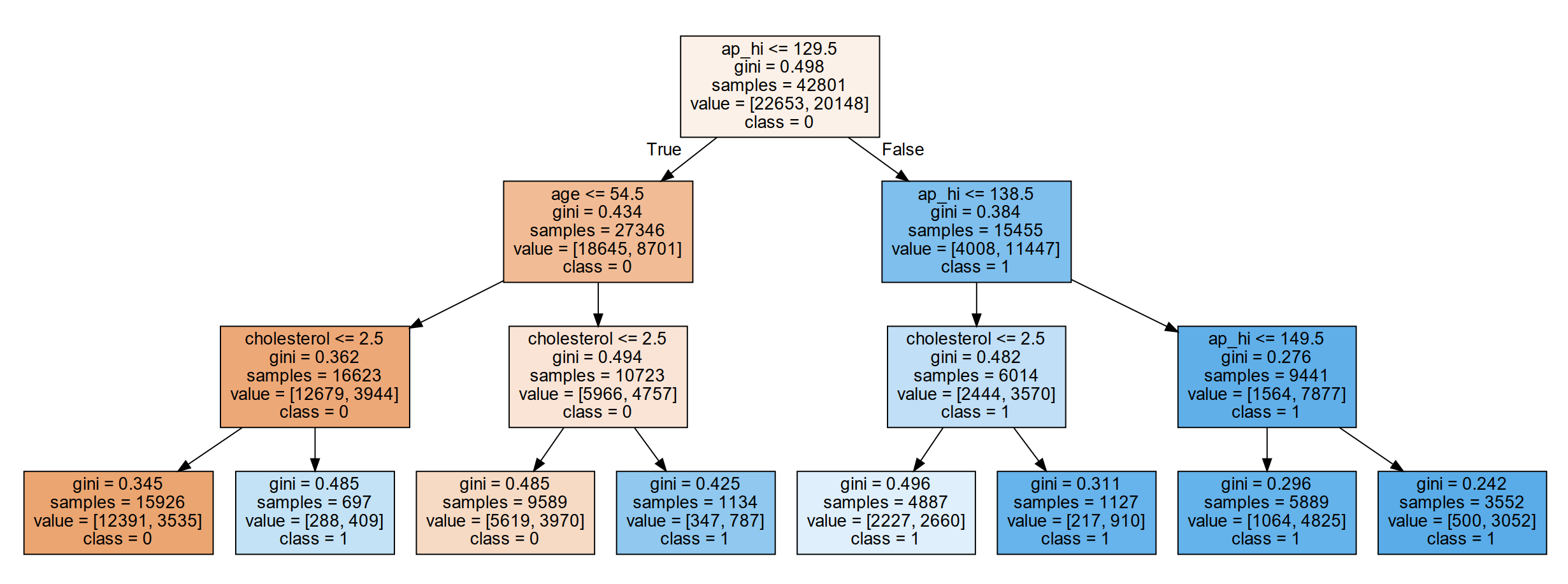












**ВИСНОВКИ**

**В результаті виконаної лабораторної роботи одержати знання для побудови дерев рішень за допомогою яких можна автоматизовано приймати рішення.**

**Усі матеріали викладенні у репозіторії GitHub, за посиланням** <https://github.com/Max11mus/Artifition-Intelect-Lab10.git>**.**