DOI 10.15276/imms.v11.no1-2.36
УДК 004.85

Informatics and Mathematical Methods in Simulation
Vol. 11 (2021), No. 1-2, pp. 36-47

Діагностика хвороби серця на основі дерев рішень

В.Г. Пенко, І.М. Шпінарева , О.В. Ярощук

Одеський національний університет імені І. І. Мечнікова, вул.Дворянська,2,
м.Одеса, Україна, e-mail: vpenko@onu.edu.ua,
iryna.shpinareva@onu.edu.ua, ayaroshchuk43@gmail.com

Обсяг медичних даних в світі величезний. Швидко ростуть електронні історії хвороб. Тому для встановлення правильного діагнозу, при великій кількості різних аналізів (КТ, кардіограм і т.д.) на допомогу лікарю приходять інтелектуальні системи прогнозування серцево-судинних захворювань. Задачу прогнозування вирішують методами машинного навчання. Найбільш популярними методами машинного навчання в задачах класифікації та прогнозування є дерева прийняття рішень. Ідея, що лежить в основі дерев рішень, полягає в розбитті безлічі можливих значень вектора ознак (незалежних змінних) на непересічні безлічі і підгонці простої моделі для кожного такого безлічі. Дерева рішень дозволяють отримати високу точність у вирішенні багатьох задач, зберігаючи при цьому високий рівень інтерпретації. Дерево рішень будується автоматично в залежності від статистичних даних. У даній роботі досліджуються методи дерев прийняття рішень: CART, ID3, C4.5, Random Forest, Gradient Boosting. На основі аналізу даних методів кращий результат прогнозування серцево-судинних захворювань отримано алгоритмами Random Forest та Gradient Boosting. Метод випадкових лісів заснований на побудові ансамблю дерев рішень, кожне з яких будується за вибіркою, що отримується з вихідної навчальної вибірки за допомогою бутстрепа (тобто вибірки з поверненням). Іншим ансамблем є метод Gradient Boosting. Його основна відмінність від Random Forest полягає в тому, що в Random Forest дерева будуються незалежно один від одного, в той час як Gradient Boosting на кожному кроці покращує попередню модель. За допомогою дерева рішень (Random Forest та Gradient Boosting) можна з розумною точністю передбачити вразливість до серцевих захворювань у пацієнтів. У роботі пропонуються поліпшення методу Gradient Boosting шляхом модифікації бустінга. А саме, на кожному кроці алгоритму новий єлемент ансамблю будується спираючись не на всю навчальну вибірку, а лише на випадкову підвибірку фіксованого розміру. Ця ідея є об'єднанням технік градієнтного бустінга і беггінга В якості вихідних даних виористовується набір Heart Disease UCI. Для перевірки результатів роботи поліпшеного алгоритму Gradient Boosting використовувалася набір heart_failure_clinical_records. В результаті проведеної роботи отримано алгоритм, який дозволяє збільшити точність прогнозування серцево-судинних захворювань з 89% до 94%.

Ключові слова: інтелектуальна система прогнозування; алгоритм дерева рішень; прогнозування серцевих захворювань; ансамбль дерев рішень.

Вступ

Сьогодні багато країн зіштовхуються з настанням кризи в системі охорони здоров'я через зниження рівня глобального здоров'я населення, зростання поширеності хронічних захворювань (в першу чергу, серцево-судинних, онкологічних, covid), надзвичайно високу вартість лікування невідкладної патології. Тому обсяг медичних даних в світі величезний і швидко зростає у багатьох галузях медицини за останні десятиліття. Майбутнє медицини пропонує персоналізований мультимодальний підхід, орієнтований на інтегровану допомогу пацієнтові інтелектуальними системами підтримки прийняття рішень для лікарів. Системи підтримки прийняття рішень стають дуже важливою частиною прийняття медичних рішень, особливо в тих ситуаціях, коли рішення лікарем повинно прийматися швидко, ефективно і надійно. Оскільки для

виконання таких задач слід розглядати концептуально прості моделі прийняття рішень з можливістю автоматичного навчання, дерева рішень є дуже підходящим кандидатом. Вони вже успішно використовуються для прийняття ефективних рішень в багатьох предметних областях.

Переваги дерев рішень [1].

1. Інтуїтивність дерев рішень.
2. Дерева рішень дають можливість отримувати правила з бази даних на природній мові.
3. Алгоритм конструювання дерева рішень не вимагає від користувача вибору вхідних атрибутів.
4. Висока точність створюваних моделей.
5. Швидкий процес навчання.
6. Більшість алгоритмів конструювання дерев рішень мають можливість спеціальної обробки пропущених значень.

Огляд літератури

Точність прогнозування серцевої хвороби залежить від ефективності алгоритму машинного навчання. А навіть малі помилки в прогнозуванні хвороби можуть привести до смерті людини. На сьогоднішній день багато авторів досліджують методи прогнозування серцево-судинних захворювань.

В роботі [2] виконано прогнозування діагнозу серцевих захворювань за допомогою дерева рішень і наївного байєсівського алгоритму. Результати показують, що точність наївного Байеса і дерева рішень становить 85,03% і 84,01%.

В роботі [3] спочатку великі медичні дані розподілено в різні кластери за допомогою алгоритму KNN. Потім кожен кластер класифікується за допомогою класифікаційного алгоритму випадкового лісу. Порівняно з існуючими системами, експериментальні результати показують, що запропонований алгоритм підвищує точність даних.

Бабу та ін. в роботі [4] виконали діагностику серцевих захворювань з використанням генетичного алгоритму, алгоритму K-Means та дерева рішень. Результати показали, що дерево рішень має велику ефективність після застосування генетичного алгоритму.

В роботі [5] автори продемонстрували кілька моделей CART для прогнозування ішемічної хвороби серця, які демонструють максимально досяжну точність 100%. В роботі [6] автори показують, що комбінація алгоритму C4.5 і нечіткої експертної системи в прогнозування ішемічної хвороби досягає найвищої точності 81,82%.

Метою роботи є підвищення точності прогнозування захворювання серця шляхом дослідження і модифікації методу дерева рішень.

Алгоритми побудови рішення дерев: CART, ID3, C4.5, Random Forest, Gradient Boosting decision trees

Для прогнозування захворювання серця у пацієнта в роботі були досліджені алгоритми побудови рішення дерев: CART, ID3, C4.5, Random Forest, Gradient Boosting.

Дерево рішень – спосіб прийняття рішення, заснований на застосуванні різних функцій поділу вихідного набору даних, зокрема простих порогових правил [7]. Класифікаційно-регресійні дерева є популярною структурною моделлю прогнозування часових рядів.

Структурні моделі CART розроблені для моделювання процесів, на які впливають як безперервні зовнішні фактори, так і категоріальні.

Етапи побудови дерева рішень включають в себе наступні пункти:

– вибір критерію точності прогнозу;

– вибір типу розгалуження;

– визначення моменту припинення розгалужень;

–визначення відповідних розмірів дерева.

Дерева рішень, як і будь-який інший алгоритм машинного навчання, мають параметри, які визначають мету та напрямок виконання алгоритму і називаються критерієм інформативності.

Критерії інформативності можуть бути різними [7]: функція втрат, критерій Джині, ентропійний критерій і т. д. Також існують різні критерії зупинки.

Головна відмінність зазначених алгоритмів полягає в різних умовах інформативності. В алгоритмі ID3 використовується ентропійний критерій:

$$E(S) = \sum_{i=1}^{c} -p_i \, log_2 p_i \ , \tag{1}$$

де p_i – частка об'єктів класу i, які потрапили в вершину S.

У алгоритмі C4.5 (поліпшена версія ID3) вибір атрибута відбувається на підставі нормалізованого приросту інформації (Gain Ratio). В алгоритмі CART використовується критерій Джині:

$$Gini(T) = 1 - \sum_{i=1}^{n} p_i^2 \ , \tag{2}$$

якщо набір даних T містить дані n класів.

Дерева рішень ID3, C4.5, CART схильні до перенавчання і мають низьку узагальнюючу здатність, тому популярні алгоритми засновані на побудові великої кількості (ансамблю) дерев рішень: Random forest, Gradient Boosting.

Випадковий ліс рішень Random forest замість одного дерева використовує сукупність (ансамбль) дерев рішень, побудованих алгоритмом шляхом модифікації попереднього дерева. При цьому суть боротьби з проблемою неефективного вибору ознак полягає в використанні в процесі побудови дерева деяких випадкових вибірок, що прибирає детермінованість побудови дерева і робить цей процес стохастичним.

Опишемо алгоритм Random forest:

1) генерує випадкову підвибірку з повторенням розміром n з начальної вибірки;

2) будуємо дерево рішень, яке класифікує приклади такої підвибірки. Причому в ході створення чергового вузла дерева будемо вибирати ознаку, на основі якої проводиться розбивка, не по всіх M ознак, а лише з m випадково обраних. Вибір найкращого з цих m ознак може здійснюватися різними способами;

3) дерево будується до повного вичерпання підвибірки і не піддається процедурі відсікання.

Зрозуміло, що така схема побудови відповідає головному принципу ансамблювання: базові алгоритми повинні бути хорошими і різноманітними (тому кожне дерево будується на своїй навчальній вибірці і при виборі розщеплення є елемент випадковості).

Алгоритм Gradient Boosting починає роботу з побудови початкової моделі і коригує її, крок за кроком створюючи послідовність дерев регресії. Кожне дерево в послідовності створюється на підставі залишків моделі, яка зводиться на попередньому кроці. Залишки моделі по суті використовуються в якості цільової змінної. По суті вирішується задача (3).

$$F = \sum_{i=1}^{m} L(y_i, a(x_i) + b_i) \rightarrow min \tag{3}$$

де $a(x_i)$ – спочатку побудований алгоритм, b_i – далі вибудовується алгоритм, який здійснює коригування відповідей $a(x_i)$ до вірних. Таким чином, поліпшується функціонал $\varepsilon_i = y_i - a(x_i)$.

Gradient Boosting має три основні компоненти.

1) Функція збитків – роль функції збитків полягає в оцінці того, наскільки модель здатна робити прогнози з урахуванням даних. Класичною функцією збитків є квадратична функція витрат:

$$L = (y - f(x))^2 \tag{4}$$

2) Слабкий учень – метод посилення градієнта, який приймає дійсне значення у і шукає наближення у вигляді зваженої суми функцій $\hat{F}(x)$ з класу:

$$\hat{F}(x) = \sum_{i-1}^{M} \gamma_i h_i(x) + const \tag{5}$$

3) Адитивна модель – це ітеративний та послідовний підхід додавання дерев (слабких учнів) по одному кроку за раз. Іншими словами, кожна ітерація повинна зменшити значення нашої функції втрат.

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{6}$$

Джерела медичних даних

Джерела медичних даних включають в себе:

– клінічні дані для підтримки прийняття рішень різної спеціалізації (діагностичні, прогностичні, догляд за хворими і т.д.), у вигляді стандартизованих даних з електронних історій хвороби;

– зареєстровані дані з датчиків моніторингу і записуючих пристроїв.

В якості вихідних даних обрано діагностичні дані із набору Heart Disease [8]. Heart Disease містить у собі 303 екземпляри, що складаються з 14 атрибутів, таких як: вік, стать, біль грудях, тиск, холестерин, цукор у крові, електрокардіографічні зміни у стані покою, максимальний пульс людини, стенокардія, кількість великих судин, таласемія, наявність серцево-судинних захворювань. Більш точний опис атрибутів можна побачити у таблиці 1.

Таблиця 1

Опис атрибутів

Атрибут	Опис
age	Вік людини в роках
sex	Стать людини (1 = чоловік, 0 = жінка)
cp	Випробовувана біль в грудях (1: типова стенокардія, 2: атипова стенокардія, 3: чи не стенокардія, 4: безсимптомна)
restecg	Кров'яний тиск в стані спокою (мм рт.ст. при надходженні до лікарні)
chol	Вимірювання холестерину в мг / дл
fbs	Рівень цукру в крові людини натщесерце (> 120 мг / дл, 1 = true; 0 = false)
thalach	Електрокардіографічний вимір в спокої (0 = нормальне, 1 = з аномалією хвилі ST-T, 2 = виявлення ймовірної або певної гіпертрофії лівого шлуночка за

	критеріями Естес)
exang	Максимальний пульс людини
oldpeak	Стенокардія, індукована навантажувальним тестом (1 = yes; 0 = no)
trestbps	Депресія ST, викликана фізичними вправами щодо стану спокою
slope	Нахил сегмента ST в пік фізичних вправ (1: косовисхідний, 2: плоский, 3: кососпадний)
ca	Кількість великих судин (0-3)
thal	Захворювання крові під назвою таласемія (3 = нормальний; 6 = фікс. дефект; 7 = оборотний дефект)
target	Наявність серцево-судинного захворювання (0 = no, 1 = yes)

Для вибору алгоритму прогнозування захворювань в роботі були використани базові моделі алгоритмів ID3, C4.5, CART, Random forest та Gradient Boosting з бібліотеки sklearn. Тестування моделей відбувалося на наборі Heart Disease.

З метою підтвердження результативності роботи механізму прогнозування алгоритмів машинного навчання в роботі використовується метрика Accuracy. Accuracy – частка правильно класифікованих об'єктів показує ймовірність того, що клас буде передбачений правильно:

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})},$$

де TP -True positive, TN - True negative, FP - False positive, FN - False negative.

На рисунку 1 продемонстровано дерево прогнозування захворювання серця побудоване алгоритмом Random forest.

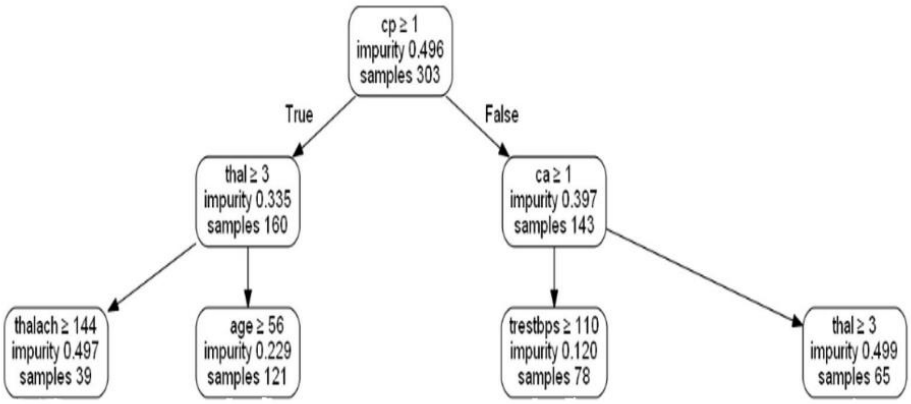

Рис. 1. Дерево рішень побудоване RF

Алгоритми з різними критеріями інформативності мають точність від 75,5% до 89,6% (рис.2).

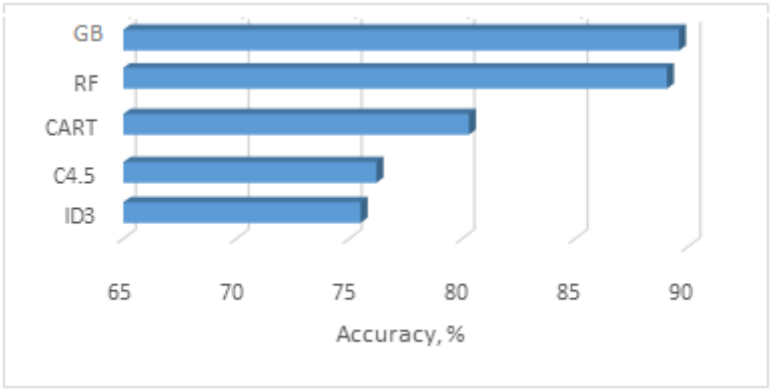

Рис. 2. Результати роботи алгоритмів дерева рішень для прогнозування захворювання

На основі даного аналізу в якості базового алгоритму обраний алгоритм Gradient Boosting, який має найбільшу точність в порівняні с другими алгоритмами – 89,6%.

Алгоритм Gradient Boosting використовує такі гіперпараметри як:

–min_samples_split: мінімальне число точок, необхідне для розподілення. Корисно, щоб уникнути перенавчання;

–min_samples_leaf: мінімальна кількість елементів у листі або вузлів дерева. Менші значення слід вибирати для незбалансованих вибірок;

–min_weight_fraction_leaf: схожий на попередній, лише замість кількості задає долю від загального числа елементів;

–max_depth: максимальна глибина дерева. Використовується для боротьби з перенавчанням;

–max_lead_nodes: Максимальне число листів у дерева. Якщо задати цей гіперпараметр, попередній ігнорується;

–max_features: кількість признаків, випробуваних алгоритмом при пошуку кращого розподілення.

Модифікації алгоритму Gradient Boosting

Враховуючи опис вище, реалізовано наступні модифікації базової версії алгоритму Gradient Boosting. Основна причина ефективності бустінга в тому, що алгоритм на кожній ітерації будує базовий алгоритм, який дійсно ефективний лише на частині підвибірки. Цей принцип можна посилити, зробивши модифікацію бустінга. А саме, на кожному кроці алгоритму новий доданок рахується спираючись не на всю навчальну вибірку, а лише на випадкову підвибірку фіксованого розміру. Ця ідея є об'єднанням технік градієнтного бустінга і беггінга. Також можна брати не випадкову підвибірку об'єктів, а ще й випадкову підвибірку ознак об'єктів. Це називається технікою випадкових підпросторів. Результати роботи таких модифікацій часто помітно перевершують за якістю різні нестохастичні варіанти.

На кожній ітерації градієнтного бустінга алгоритм прагне максимально виправити всі помилки на навчанні. Однак це безглуздо при наявності шуму в вихідних даних і веде до перенавчання. Проблему можна вирішити, враховуючи ваги об'єктів на кожній ітерації, адже по них можна судити про складність навчання. Дійсно, велика вага у об'єкта показує, що попередні алгоритми погано працювали на ньому і, можливо, цей об'єкт шумовий.

Тому для поліпшення роботи алгоритму Gradient Boosting будемо виконувати наступні кроки:

1. отримаємо важливість атрибутів у датасеті;
2. на кожному кроці алгоритму новий доданок рахується спираючись не на всю навчальну вибірку, а лише на випадкову підвибірку фіксованого розміру;

3. на кожній ітерації будемо враховувати ваги об'єктів, адже по них можна судити про складність навчання на них;
4. виконуємо зміну гіперпараметрів: глибину дерева та мінімальну кількість елементів в узлі дерева

Дані для навчання та тестування ділимо у співвідношенні 1/2 на 1/2. Поділ на тестову та тренувальну вибірку реалізується випадковим методом. Набір даних неодноразово підбирався з випадковим поділом даних на навчальний і тестовий набір.

Для виконання таких умов була обрана мова програмування Python. Були використанні такі бібліотеки, як: matplotlib – для побудови графіків, pandas для перетворення даних, sklearn.model_selection – для розбиття даних.

Оцінка ефективності модифікованого алгоритму

Для виконання аналізу результатів потрібно створити порівняння декількох версій створеного алгоритму, щоб побачити чи принесли модифікації позитивний вплив на якість роботи алгоритму, або навпаки.

Для порівняння будуть використані наступні версії:

– базова версія алгоритму Gradient Boosting;

– версія алгоритму з попередньою обробкою даних;

– версія алгоритму з використанням техніки випадкових підпросторів;

– версія з видаленням шуму.

Кількість елементів тестової вибірки залежить від навчальної та складатиме 1/2 від початкової вибірки.

Спочатку перевіримо важливості атрибутів, що обчислені в Gradient Boosting (рис.3).

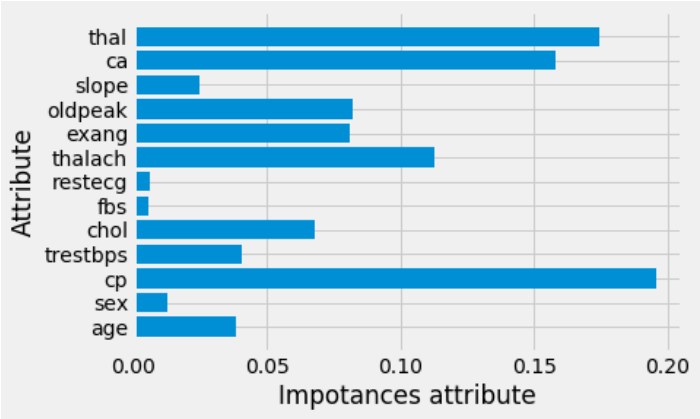

Рис. 3. Важливість атрибутів градієнтного бустінгу

З цього графіку можна зробити висновок, що такі параметри, як кров'яний тиск в стані спокою та рівень цукру в крові мають найнижчу вагомість серед усіх ознак.

Базова версія цього алгоритму використовує 100 дерев с максимальною глибиною 3 і швидкості навчання 0.1, що дає результат у 84%

Для того, щоб побачити зміни в точності алгоритму, вирішено зробити попередню обробку даних – для цього з набору даних було видалено два атрибути з найменшою важливістю, та залишилися атрибути з важливістю більш рівною від 0.045 до 0.23 (рис.4). Це вплинуло на точності класифікації, яка склала 89%.

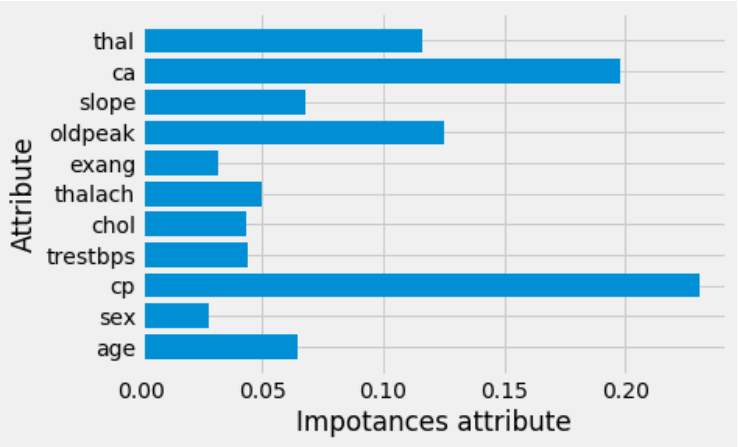

Рис. 4. Важливість атрибутів після попередньої модифікації

Перевіримо версію алгоритму з використанням випадкових підпросторів, але не використовуємо попередню обробку даних. Будемо використовувати 25 дерев з максимальною глибиною 3 і швидкістю навчання 0.1. Цей алгоритм дав результат 92% точності. Графік залежності точності модифікованого алгоритму на тестовій виборці на 25 тестах, зображений на рисунку 5.

Розглянемо той самий алгоритм, але з попередньою обробкою даних, де також використовуємо 25 дерев з тими самими параметрами. Результат роботи алгоритму з використанням випадкових підпросторів з попередньою обробкою даних має точність на тестовому наборі 93.814%. Графік залежності точності модифікованого алгоритму на тестовій виборці, на 25 тестах, зображений на рисунку 6.

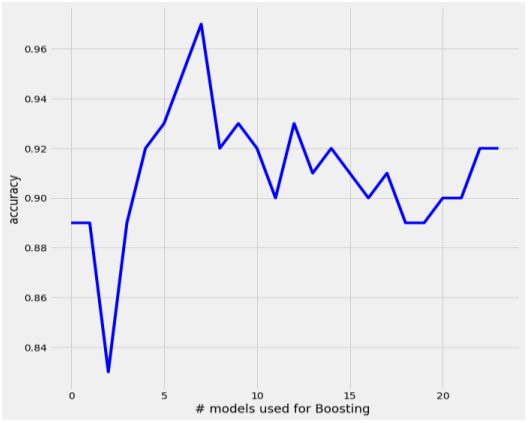

Рис. 5. Результат роботи алгоритму з використанням випадкових підпросторів без попередньої обробки даних

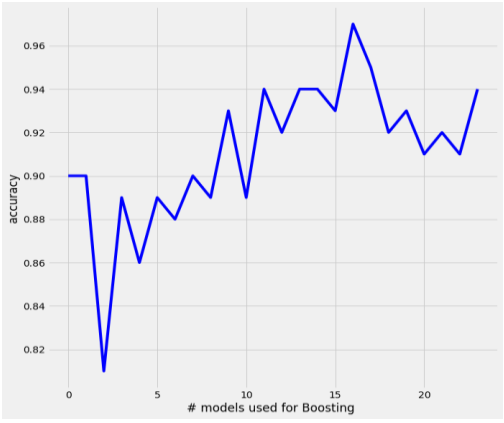

Рис. 6. Результат роботи алгоритму з використанням випадкових підпросторів з попередньою обробкою даних

Розглянемо модифікацію алгоритму з використанням випадкових підпросторів, але не використовуємо попередню обробку даних. Будемо використовувати 25 дерев з максимальною глибиною 2 і швидкості навчання 0.1. Цей алгоритм дав результат у 90% точності (рис. 7).

Розглянемо модифікацію алгоритму з використанням випадкових підпросторів, з попередньою обробкою даних. Використовуємо 25 дерев з максимальною глибиною 2 і швидкістю навчання 0.1. Цей алгоритм дав результат у 92% точності (рис. 8).

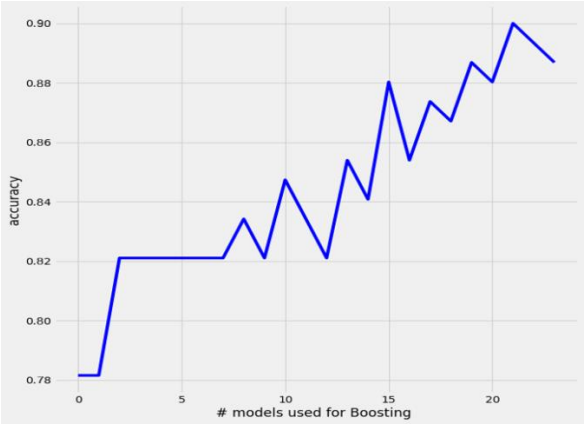

Рис.7. Результат роботи алгоритму з використанням випадкових підпросторів без попередньої обробки даних з глибиною 2

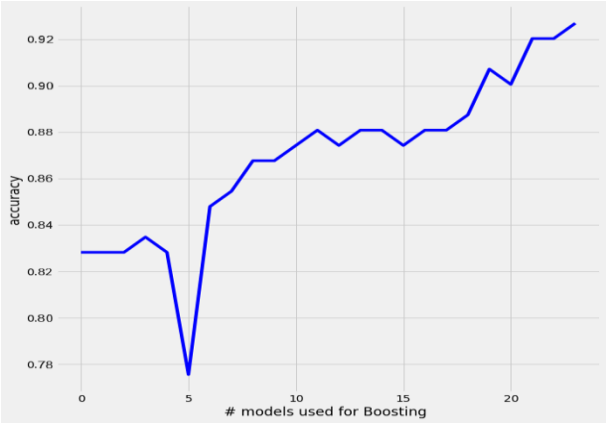

Рис. 8. Результат роботи алгоритму з використанням випадкових підпросторів з попередньою обробкою даних з глибиною 2

Розглянемо версію алгоритму з видаленням шуму без попередньої обробки даних. Запустимо алгоритм з видаленням шуму на наборах даних, з максимальною глибиною 3 та швидкістю навчання 0.1. Результатом роботи цього алгоритму є 85% на тестовій виборці, та 100% на навчальній (рис.9). З чого можна зробити висновок, що алгоритм видаляє не тільки шуми, але і важливі ознаки, тобто використання цього алгоритму не є доцільним.

Розглянемо цей самий алгоритм з попередньою обробкою даних з тими самими параметрами. Як зображено на рисунку 10, попередня обробка даних не дала поліпшення результатів алгоритму, а навпаки, трохи зменшила точність.

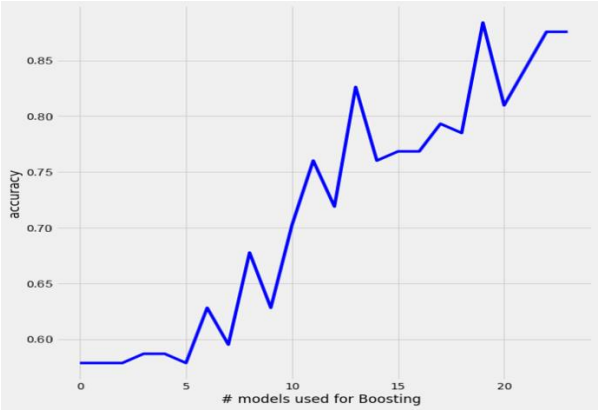

Рис. 9. Результат роботи алгоритму з видаленням шуму без попередньої обробки даних

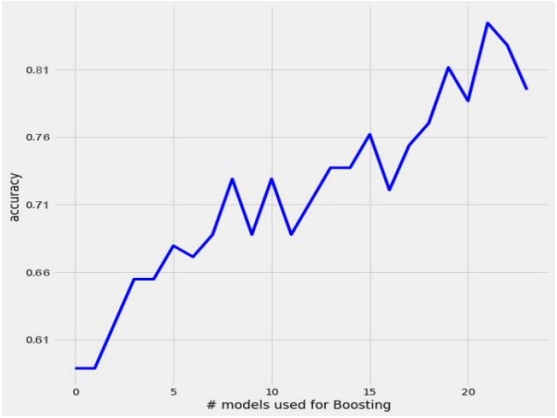

Рис.10. Результат роботи алгоритму з видаленням шуму без попередньої обробки даних

Таблиця 2

Порівняння результатів роботи алгоритму GB і запропонованого алгоритму

Алгоритм	Accuracy, %
Базова версія алгоритму GB	84
Базова версія алгоритму з попередньою обробкою даних	89
Версія алгоритму з використанням випадкових підпросторів	92
Версія алгоритму з використанням випадкових підпросторів з попередньою обробкою даних	94
Версія алгоритму з видаленням шуму	85
Версія алгоритму з видаленням шуму з попередньою обробкою даних	81

Для того, щоб переконатися в результатах тестування запропонованих варіантів модифікованого алгоритму Gradient Boosting, протестуємо на dataset: kaggle / input / heart-failure-clinical-data/heart_failure_clinical_records_dataset.csv.

Результати стандартного алгоритму Gradient Boosting тестування з урахуванням всіх параметрів 82.667%, з видаленням незначних параметрів 85.333%. Точність роботи модифікованого алгоритму Gradient Boosting з використанням випадкових підпросторів з max_dept = 3 і з видаленням незначних параметрів на даному датасеті має точність 93.97% (рис.11,12)

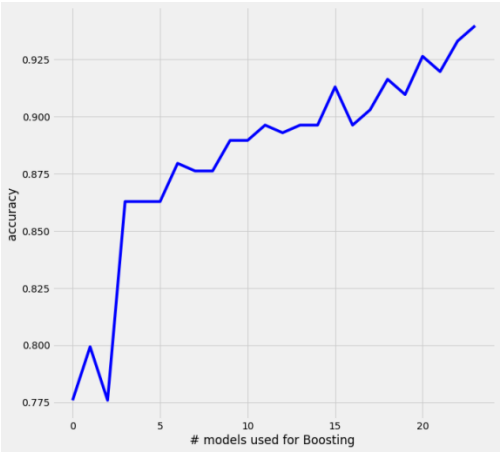

Рис. 11. Результат роботи модифікованого алгоритму Gradient Boosting використанням випадкових підпросторів з max_dept = 3

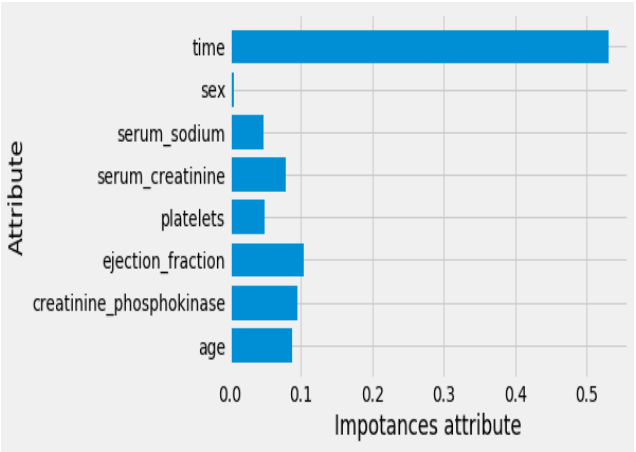

Рис. 12. Важливість атрибутів heart_failure_clinical_records_dataset.csv

Висновки

Аналізуючи алгоритми дерев прийняття рішень в задачі прогнозування серцево-судинних захворювань можна зробити висновок, що дерева рішень ID3, C4.5, CART мають недоліки, а саме, схильні до перенавчання і мають низьку узагальнюючу здатність, тому ефективніше використовувати ансамблі дерев рішень Random forest та Gradient Boosting. Дані методи були навчені і протестовані на базі Heart Disease та дали точність від 75,5 до 89,6. В роботі використано ансамблі дерев Gradient Boosting для прогнозування серцево-судинних захворювань. Виконані модифікації алгоритму Gradient Boosting: урахування важливості параметрів, використано метод випадкових підпросторів, видалення шуму та зміна гіперпараметрів. В результаті проведеної роботи отримано алгоритм, який дозволяє збільшити точність встановлення діагнозів з 89% до 94%.

Список літератури

1. Т.В. Зайцева, Н.В. Васина, О.П. Пусная, Н.Н. Смородина Программная реализация метода деревьев решений для решения задач классификации и прогнозирования Научные ведомости Серия История. Политология. Экономика. Информатика. 2013. №7 (150). Выпуск 26/1
2. Venkatalakshmi, B., & Shivsankar, M. V. (2014). Heart disease diagnosis using predictive data mining. International Journal of Innovative Research in Science, Engineering and Technology, 3(3), 1873-7. 2014 року
3. Saravana Kumar R., Manikandan P. Medical Big Data Classification Using a Combination of Random Forest Classifier and KMeans Clustering, I.J. Intelligent Systems and Applications, 2018, 11, 11-19 Published Online November 2018 in MECS (http://www.mecs-press.org/)
4. Babu, S., Vivek, E. M., Famina, K. P., Fida, K., Aswathi, P., Shanid, M., & Hena, M. (2017, April). Heart disease diagnosis using data mining technique. In Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of (Vol. 1, pp. 750-753). IEEE.
5. Mohammad M.Ghiasi, Sohrab Zendehboudia, Ali Asghar Mohsenipourb Decision tree-based diagnosis of coronary artery disease: CART model. Computer Methods and Programs in Biomedicine , Volume 192, August 2020
6. Baihaqi, W. M., Setiawan, N. A., & Ardiyanto, I. (2016, August). Rule extraction for fuzzy expert system to diagnose coronary artery disease. In Information Technology, Information Systems and Electrical Engineering (ICITISEE), International Conference on (pp. 136-141). IEEE.
7. С. А. Митрофанов. О модификации алгоритма обучения дерева решений, Журнал Решетневские чтения, т.2., 2018
8. Heart Disease Data Set [Електронний ресурс]. – Режим доступу: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

Diagnosis of heart disease based on decision trees

V.G. Pienko, I.M. Shpinareva, A.V. Yaroshchuk

Odessa I.I.Mechnikov National University, 2 Dvoryanska str.,
Odessa, Ukraine,e-mail: vpenko@onu.edu.ua,
iryna.shpinareva@onu.edu.ua, ayaroshchuk43@gmail.com

The volume of medical data in the world is enormous. Electronic medical records are growing rapidly. Therefore, in order to establish the correct diagnosis, with a large number of different analyzes (CT, cardiograms, etc.), intelligent systems for predicting cardiovascular diseases come to the aid of the doctor. The prediction problem is solved by machine learning methods. The most popular machine learning method for classification and prediction are decision trees. The idea behind decision trees is to split the set of possible values of the feature vector (independent variables) into disjoint sets and fit a simple model for each such set. Decision trees allow you to get high accuracy in solving many problems, while maintaining a high level of interpretation. The decision tree is built automatically depending on the statistical data. This paper examines different types of decision trees: CART, ID3, C4.5, Random Forest, Gradient Boosting. Based on the analysis of these types, the best result for predicting cardiovascular diseases was obtained by the Random Forest and Gradient Boosting methods. The random forest method is based on the construction of an ensemble of decision trees, each of which is constructed from a sample obtained from the original training sample using a bootstrap. Another ensemble is the Gradient Boosting method. Its main difference from Random Forest is that in Random Forest trees are built independently of each other, while Gradient Boosting improves the previous model at every step. Using a decision tree (Random Forest and Gradient Boosting), you can predict the vulnerability to heart disease in patients with reasonable accuracy. The paper proposes improvements to the Gradient Boosting method by modifying boosting. Namely, at each step of the algorithm, a new ensemble item is constructed based not on the entire training sample, but only on a random subsample of a fixed size. This idea is a combination of gradient boosting and bagging techniques. The Heart Disease UCI set is used as initial data. The heart_failure_clinical_records set was used to test the performance of the improved Gradient Boosting algorithm. As a result of the work carried out, an algorithm was obtained that allows increasing the accuracy of predicting cardiovascular diseases from 89% to 94%.

Keywords: intelligent forecasting system; decision tree algorithm; predicting cardiovascular disease; an ensemble of decision trees.