Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра обчислювальної математики

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня бакалавра
за спеціальністю 113 Прикладна математика
на тему:
Прогнозування результатів лікування раку методами машинного навчання

Виконав студент IV курсу
Киричек Микола Павлович _____
(підпис)

Науковий керівник:
асистент кафедри ОМ
Денисов Сергій Вікторович _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань

Студент _____
(підпис)

Роботу розглянуто і допущено до захисту на засіданні кафедри обчислювальної математики
«29» травня 2023р.,
Протокол № 8
Завідувач кафедри
Ляшко С.І _____
(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи 52 сторінки, 18 ілюстрацій, 10 таблиць, 22 джерел посилань.

МАШИННЕ НАВЧАННЯ, ПРЕПРОЦЕСИНГ ДАНИХ, НЕЙРОННА МЕРЕЖА, ГРАДІЄНТНИЙ БУСТИНГ, ОПТИМІЗАЦІЯ ГІПЕРПАРАМЕТРІВ, OPTUNA, MULTI-LAYER PERCEPTRON, LIGHT GRADIENT BOOSTED MACHINE

Об’єктом дослідження є діагностичні алгоритми для прогнозування радіойодрезистентності раку щитоподібної залози на основі методів градієнтного бустингу та нейронних мереж.

Метою роботи є дослідження методів машинного навчання для прогнозування радіойодрезистентності раку щитоподібної залози та розробка програмного забезпечення.

Методи розроблення: комп’ютерне моделювання, алгоритми машинного навчання. Інструменти розроблення: Jupyter Notebook, Google Colaboratory, мова програмування Python, середовище програмування PyCharm, бібліотеки lightgbm, sklearn, optuna.

Результати роботи: розглянуто декілька моделей прогнозування радіойодрезистентності раку щитоподібної залози, розроблено застосунок для програмного представлення задач прогнозування раку, виконано тестування програмного засобу на наборах даних реальних пацієнтів, проведене порівняння швидкодії та точності методів.

ЗМІСТ

ВСТУП

Актуальність роботи. Онкологічні захворювання є основною причиною смертності у світі. За даними Всесвітньої організації охорони здоров'я, в 2020 році на їхню частку припало близько 10 млн смертей. Найбільш ефективним способом боротьби з раком є його рання діагностика, оскільки рак, виявлений на ранній стадії, краще піддається лікуванню. Важливим сучасним направленням боротьби з раком є розробка способів прогнозування його появи та поведінки з метою використання адекватних ефективних методів лікування та моніторингу злоякісних пухлин.

Проблемою діагностики та лікування раку щитоподібної залози є радіойодрезистентні метастази, для яких радіойодтерапія неефективна. Актуальним питанням є можливість раннього прогнозування та вчасної діагностики радіойодрезистентні метастази на основі виявлення цитоморфологічних особливостей раку щитоподібної залози та його метастазів, які корелюють із розвитком радіойодрезистентності.

Об’єктом дослідження є діагностичні алгоритми для прогнозування радіойодрезистентності раку щитоподібної залози на основі методів градієнтного бустингу та нейронних мереж.

Метою роботи є дослідження методів машинного навчання для прогнозування радіойодрезистентності раку щитоподібної залози та розробка програмного забезпечення.

Методи розроблення: комп’ютерне моделювання, алгоритми машинного навчання. Інструменти розроблення: Jupyter Notebook, Google Colaboratory, мова програмування Python, середовище програмування PyCharm, бібліотеки lightgbm, sklearn, optuna.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Задачі прогнозування радіойодрезистентності раку щитоподібної залози

Особливе місце серед злоякісних захворювань в Україні займає рак щитоподібної залози, частота якого зросла в 4-10 разів після аварії на ЧАЄС, особливо у дітей і підлітків, щитоподібна залоза яких є найбільш вразливою до дії радіоактивних ізотопів йоду [1,2,3] Частота цього захворювання зростає також у всьому світі, і за останнє десятиліття показник темпу зростання захворюваності на рак ЩЗ (РЩЗ) вийшов на перше місце серед усіх злоякісних захворювань [4-6]. На відміну від інших злоякісних захворювань, рак щитоподібної залози вважається сприятливим, оскільки існує ефективний специфічний спосіб його лікування – радіойодтерапія, завдяки чому більшість пацієнтів з раком щитоподібної залози мають тривалість життя, яка не відрізняється такої від здорових людей [7-9].

Але, найбільшою проблемою діагностики і лікування раку щитоподібної залози є так звані радіойодрезистентні метастази, які втрачають здатність до накопичення радіойоду та їх лікування стає неможливим [10, 11]. Такі пацієнти мають значно знижену медіану виживання та складають 5-15% від загальної кількості хворих на рак щитоподібної залози. Відомо, що запізнення у виявленні метастазів різко погіршує показники виживання хворих на рак щитоподібної залози. Тому актуальними є питання прогнозування появи радіойодрезистентності раку щитоподібної залози.

Злоякісні пухлин щитоподібної залози мають певні характеристики, які можуть бути показниками їх агресивної поведінки. Вони відображають агресію пухлин, їх здатність розповсюдження за межі первинного органа, до метастазування, до неконтрольованого росту в організмі хворого. Важливими показниками агресії злоякісних пухлин є такі їх характеристики, як здатність

до багатофокусного росту (F), інвазії в капсулу (Inv tumour) та судини (Inv vessel), наявність метастазів в лимфовузли (Mts). Відомо, що для пацієнтів з екстратиреоїдним розповсюдженням (Extrathyroid) раку щитоподібні залози ризик рецидивів і смерті є більшим, ніж для пацієнти без такого розповсюдження [12]. Судинну інвазію (Inv vessel) вважають важливим предиктором розвитку віддалених метастазів та ефективності радіойодтерапії [13]. Пацієнти з мультицентричним раком (F) щитоподібної залози демонструють вищу частоту рецидивів і смертність, ніж пацієнти з уніфокальною мікрокарциномою [14]. Присутність пухлинного некрозу (Nekroz) та оксифілії (O) в гістологічних зразках не лише відображає пухлинну біологію, а й надає додаткову цінну інформацію щодо несприятливого прогнозу пухлини [15-17].

Важливе значення мають як гістологічні характеристики пухлин, які досліджують після проведення оперативного втручання на післяопераційному матеріалі, так і цитологічні, тобто клітинні характеристики, які вивчають ще до проведення операції.

TNM – гістологічна класифікація пухлини

T – характеристика первинної пухлини, яка відображає її розмір та здатність до екстратиреоїдного розповсюдження, тобто за межі пухлини

N – наявність та кількість регіонарних метастазів пухлини

M – наявність віддалених метастазів.

Якщо існують деякі літературні дані щодо кореляції агресії та радіойодрезистентності пухлин з гістологічними характеристиками, то відносно цитологічних характеристик подібні дослідження не проводились. Тому дуже важливо оцінити цінність комплексу цитологічних та гістологічних характеристик пухлин щитоподібної залози як факторів розвитку їх радіойодрезистентності. Це може стати основою розробки комплексних

методів прогнозування радіойодрезистентності раку щитоподібної залози, з метою використання адекватних методів терапії.

1.2. Діагностичні вибірки

У роботі використано матеріал ППР ЩЗ, отриманий від пацієнтів від 13 до 81 років, які проходили обстеження, хірургічне лікування та радіойодтерапію в ДУ «Інститут ендокринології ті обміну речовин ім. В.П.Комісаренка НАМН України».

Для дослідження було обрано дані 306 пацієнтів, які мають 29 характеристик (ознак), що можуть бути показниками агресії злоякісних пухлин.

Навчальна вибірка складається з двох груп пацієнтів: 149 пацієнтів з радіойодрезистентним раком щитоподібної залози і 157 пацієнтів з виліковним раком щитоподібної залози.

Контрольна вибірка складається з 62 пацієнтів. Для кожного пацієнта отримується по 29 ознак. Така контрольна вибірка, як правило, не має жодних спільних елементів із початковою.

РОЗДІЛ 2. ДІАГНОСТИЧНИЙ АЛГОРИТМ НА ОСНОВІ МЕТОДУ ГРАДІЄНТНОГО БУСТИНГУ

2.1. Градієнтний бустинг

Метод градієнтного бустингу запропоновано вперше у 1999р. у роботі J. H. Friedman Greedy Function Approximation: A Gradient Boosting Machine [18]. Градієнтний бустинг, він же Gradient Boosting, Gradient Boosting on Decision Trees (GBDT) є ансамбль дерев рішень, навчений з використанням градієнтного бустингу. В основі цього алгоритму лежить ітеративне навчання дерев рішень з метою мінімізації функції втрат. Завдяки особливостям дерев рішень градієнтний бустинг здатний працювати з категорійними ознаками. Такий бустинг здатний ефективно знаходити нелінійні залежності даних різної природи. Цю властивість мають всі алгоритми, що використовують дерева рішень, однак саме GBDT зазвичай виграє в переважній більшості завдань. Завдяки цьому він широко застосовується в задачах з індустрії (пошукове ранжування, рекомендаційні системи, таргетування реклами, передбачення погоди, пункту призначення таксі та багатьох інших).

Бустинг – це метод перетворення слабонавчених моделей на добре навчені. У бустингу кожне нове дерево навчається на модифікованій версії вихідного датасету. Градієнтний бустинг схожий на алгоритм адаптивного бустингу (AdaBoost). AdaBoost навчає дерево, в якому кожному спостереженню надається вага. Після оцінки дерева збільшуються ваги тих спостережень, які складно класифікувати, та зменшуються ваги тих, що легко класифікувати. Наступне дерево спирається на ці скориговані ваги. Таким чином, поточну модель можна позначити як Tree 1 + Tree 2. Потім підраховується помилка цієї моделі та створюється наступне дерево для видачі нових прогнозів. Цей процес триває стільки разів, скільки вказано. Наступні дерева допомагають класифікувати спостереження, які погано класифікуються попередніми

деревами (або знайти значення, якщо це регресія). Через війну прогнози остаточної ансамблевої моделі є зваженою сумою прогнозів, зроблених попередніми деревами [19].

Gradient Boosting навчає безліч моделей поступово, адитивно та послідовно. Різниця між градієнтним та адаптивним бустингами полягає в тому, як алгоритми ідентифікують слабкі моделі. AdaBoost виявляє їх на підставі високих значень ваг, а GB – на підставі градієнтів функцій втрат. Функція втрат – це міра, яка показує, наскільки добре коефіцієнти моделі відповідають базовим даним. Тому вибір тієї чи іншої функції втрат у цьому алгоритмі машинного навчання важливий.

2.1.1. Функції втрат

Для градієнтного бустингу передбачено різні функції втрат. Нижче наведено таблицю з основними функціями втрат, де N – кількість спостережень, y_i – клас/значення спостереження i, x_i – ознаки спостереження i, $F(x_i)$ – передбачений клас/значення спостереження i.

Таблиця 2.1.

Втрата	Задача	Формула	Опис
Логістична втрата (log loss)	Класифікація	$2\sum_{i=1}^{N}\log(1+\exp(-2y_iF(x_i)))$	Застосовується тільки в бінарній класифікації
Квадратична помилка	Регресія	$\sum_{i=1}^{N}(y_i-F(x_i))^2$	L2-втрати
Абсолютна помилка	Регресія	$\sum_{i=1}^{N}\lvert y_i-F(x_i)\rvert$	L1-втрати

2.1.2. Побудова композиції

Розглянемо найбільш загальний спосіб бустингу [18], окремими випадками чи модифікаціями якого, так чи інакше, є всі сучасні методи бустингу. Розглянемо задачу розпізнавання об'єктів із багатовимірного простору X з простором міток Y. Нехай нам дана навчальна вибірка $\{x_i\}_{i=1}^{N}$, де $x_i \in X$. І нехай на ній відомі справжні значення міток кожного об'єкта $\{y_i\}_{i=1}^{N}$, де $y_i \in Y$. Необхідно побудувати оператор, що розпізнає, який якомога точніше зможе передбачати мітки для кожного нового об'єкта $x \in X$.

Нехай нам задано деяке сімейство базових алгоритмів H, кожен елемент $h(x; a) \in H : X \to R$ якого визначається деяким вектором параметрів $a \in A$.

Шукатимемо фінальний алгоритм класифікації у вигляді композиції

$$F_M(x) = \sum_{m}^{M} b_m h(x; a_m), b_m \in R, a_m \in A.$$

Однак вибір оптиматильного набору параметрів $\{a_m, b_m\}_{m=1}^{M}$ дуже трудомістка задача. Тому ми намагатимемося побудувати таку композицію, щоразу додаючи до суми доданок, що є найбільш оптимальним алгоритмом із можливих. Вважатимемо, що нами вже побудований класифікатор F_{m-1} довжини m−1. Таким чином, завдання зводиться до пошуку пари найбільш оптимальних параметрів $\{a_m, b_m\}$ для класифікатора довжини m:

$$F_m(x) = F_{m-1}(x) + b_m h(x; a_m), b_m \in R, a_m \in A.$$

Оптимальність тут розуміється відповідно до принципу явної максимізації відступів. Це означає, що вводиться певна функція втрат $L(y_i, F_m(x_i))$, $i=\overline{1, N}$, що показує, наскільки "сильно" прогнозована відповідь $F_m(x_i)$ відрізняється від правильної відповіді y_i. І потім мінімізується функціонал помилки

$$Q = \sum_{i=1} L(y_i, F_m(x_i)) \to \min .$$

Зауважимо, що функціонал помилки Q – дійсна функція, яка залежить від точок $\{F_m(x_i)\}_{i=1}^{N}$ в N-мірному просторі, і нам необхідно вирішити задачу мінімізації цього функціоналу. Зробимо це, реалізуючи один крок способу градієнтного спуску [17]. Як точку, для якої ми шукатимемо оптимальне прирощення, розглянемо F_{m-1}. Знайдемо градієнт функціоналу помилки:

$$\left.\frac{\partial(\sum_{i=1}^{N} L(y_i, F_{m-1}))}{\partial F_{m-1}}(x_i)\right]_{i=1} = \left[\frac{\partial L(y_i, F_{m-1})}{\partial F_{m-1}}(x_i)\right]_{i=1}^{N}.$$

Таким чином, в силу методу градієнтного спуску найбільш вигідно додати новий доданок в класифікатор наступним чином:

$$F_m = F_{m-1} - b_m \nabla Q, b_m \in R,$$

де b_m підбирається лінійним пошуком за дійсними числами R:

$$b_m = \underset{b \in R}{\operatorname{argmin}} \sum_{i=1} L(F_{m-1}(x_i) - b \nabla Q_i).$$

Однак ∇Q є лише вектором оптимальних значень для кожного об'єкта x_i, а не базовим алгоритмом з родини H, визначений ∀x∈X. Тому нам необхідно знайти $h(x, a_m) \in H$ найбільш схожий на ∇Q. Зробимо це, знову мінімізуючи функціонал помилки, що базується на принципі явної максимізації відступів. що просто відповідає базовому алгоритму навчання. Далі знайдемо коефіцієнт b_m, використовуючи лінійний пошук:

$$b_m = \underset{b \in R}{\operatorname{argmin}} \sum_{i=1} L(F_{m-1}(x_i) - b h(x_i, a_m)).$$

2.1.3. Окремі випадки градієнтного бустингу: розв'язання задачі класифікації

Ідея бустингу застосовна для задачі класифікації. У разі бінарної класифікації це означає, що Y = {-1, +1}. Тоді часто мається на увазі, що кожен алгоритм h∈H повертає дійсний «ступінь» належності об'єкта до деякого

класу, а результуюча відповідь F˜ виходить застосуванням порогового правила до композиції.

У разі класифікації зазвичай використовують функцію втрат від одного аргументу L(y, F) = L(yF), тобто відступ замінюється добутком справжнього класу та пронозованого значення. У такому разі існує трохи інший погляд на підхід градієнтного бустингу. Під градієнтом функціоналу помилки можна мати на увазі вектор ваг навчальних об'єктів поелементно помножений на вірні значення класів:

$$\nabla Q = \left[\frac{\partial L(y_i F_{m-1})}{\partial F_{m-1}}(x_i)\right]_{i=1}^{N} = \left[y_i \frac{\partial L(y_i F_{m-1})}{\partial (y_i F_{m-1})}(x_i)\right]_{i=1}^{N} = [y_i w_i]_{i=1}^{N},$$

Тоді алгоритм навчання відповідно до принципу максимізації відступів набуває наступного вигляду:

$$h(x, a_m) = \text{обучить}\left(\{x_i\}_{i=1}^{N}, \{\nabla Q_i\}_{i=1}^{N}\right) =$$

$$= \underset{a_m \in A}{argmin} \sum_{i=1}^{N} L\left(y_i w_i h(x_i, a_m)\right).$$

Таким чином w_i можна розглядати з погляду терезів (ступеня важливості), які надаються об'єктам та враховуються під час навчання кожного базового алгоритму. Цей погляд склався історично раніше, ніж градієнтний підхід. До того ж він більш інтуїтивно зрозумілий.

2.1.4. Переваги градієнтного бустингу

На сьогоднішній день градієнтний бустинг є одним із найпотужніших алгоритмів розпізнавання. Це досягається завдяки вищезгаданій адаптивній техніці побудови композиції. До того ж, бустинг надає багато можливостей для варіацій. По-перше, можна розглядати різні функції втрат. Це дозволяє вирішувати як задачі класифікації, так і задачі регресії. До того ж

можливість вибору довільної функції втрат дозволяє акцентувати увагу на особливостях даних в задачі [20].

По-друге, можливий розгляд будь-якого сімейства базових алгоритмів. А це дає широкі можливості врахування особливостей задачі. Бустинг над вирішальними деревами вважається одним з найбільш ефективних варіантів бустингу. А враховуючи, що вирішальні дерева також використовують базові алгоритми (наприклад, порогові, лінійні тощо), в результаті виходить величезна кількість варіантів для налаштування.

По-третє, завдяки достатній простоті методу та чіткому математичному обґрунтуванню, у кожній конкретній варіації бустингу не складно провести деякі математичні та алгоритмічні оптимізації, які помітно пришвидшать роботу алгоритму.

Зрозуміло, що алгоритм бустингу не позбавлений певних недоліків. По-перше, бустинг - трудомісткий метод, і він працює досить повільно. Найчастіше потрібно побудувати сотні або навіть тисячі базових алгоритмів для композиції. По-друге, без додаткових модифікацій він має властивість повністю підлаштовуватися під дані (перевчитися), у тому числі під помилки та викиди у них. По-третє, ідея бустингу зазвичай погано застосовується до побудови композиції із досить складних та потужних алгоритмів. Побудова такої композиції займає дуже багато часу, а якість суттєво не збільшується. По-четверте, результати роботи бустингу складно інтерпретовані, якщо в композицію входять десятки алгоритмів.

2.2. Алгоритм Light Gradient Boosted Machine

Light Gradient Boosted Machine (LightGBM) – це швидкий, розподілений, високопродуктивний фреймворк для градієнта, що базується на алгоритмах дерева рішень. Його можна використовувати для сортування, класифікації, регресії та багатьох інших завдань машинного навчання. LightGBM розширює

алгоритм градієнтного бустингу, додаючи тип автоматичного вибору об'єктів, фокусуючись на прикладах бустингу з великими градієнтами. Це може призвести до різкого прискорення навчання та покращення прогнозних показників. Таким чином LightGBM – це реалізація градієнтного бустингу з відкритим вихідним кодом, що надає ефективну та дієву реалізацію алгоритму градієнтного бустингу.

Оскільки він заснований на алгоритмі дерева рішень, він використовує оптимальну листову стратегію для поділу листових вузлів, однак інші алгоритми покращення зазвичай використовують глибинну або рівневу замість листової для поділу дерев. Отже, в алгоритмі LightGBM при переході до того самого листового вузла листовий алгоритм зменшує більше втрат, ніж рівневий алгоритм. Це призводить до більш високої точності, яка не може бути досягнута іншими існуючими алгоритмами підйому. У той же час його швидкість така висока, що відображається в назві алгоритму Light.

LightGBM був описаний Guolin K. та співавторами у статті 2017 року під назвою «LightGBM: A Highly Efficient Gradient Boosting Decision Tree» [21]. Реалізація вводить дві ключові ідеї: GOSS та EFB.

Градієнтна одностороння вибірка (GOSS) є модифікацією градієнтного бустингу, який фокусує увагу на тих тренувальних даних, які призводять до більшого градієнта, у свою чергу, прискорюючи навчання та зменшуючи обчислювальну складність методу.

За допомогою GOSS виключається значна частка екземплярів даних з невеликими градієнтами та використовуються лише інші екземпляри для оцінки приросту інформації. Доводиться, що оскільки екземпляри даних з великими градієнтами відіграють важливу роль у обчисленні інформаційного виграшу, GOSS може отримати досить точну оцінку інформаційного виграшу зі значно меншим розміром даних..

Exclusive Feature Bundling (об'єднання взаємовиключних ознак), або EFB, - це підхід об'єднання розріджених (переважно нульових) взаємовиключних ознак, таких як категоріальні змінні вхідних даних, закодовані унітарним кодуванням. Таким чином, це тип автоматичного підбору ознак.

Разом ці дві зміни можуть прискорити час навчання алгоритму до 20 разів. Таким чином, LightGBM можна розглядати як дерева рішень із градієнтним бустингом (GBDT) з додаванням GOSS та EFB.

Принципова схема зростання дерев рішень у LightGBM наведена на Рис.2.1.

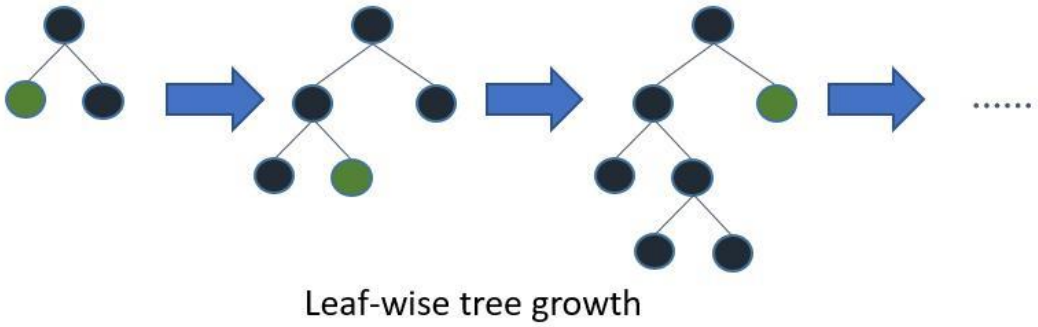


Рис. 2.1. Принципова схема зростання дерев рішень у LightGBM

Після першого поділу лівий вузол мав більші втрати і вибирається для наступного поділу. Тепер у нас є три листові вузли, і середній листовий вузл має найвищі втрати. Листовий поділ алгоритму LightGBM дозволяє працювати з великими наборами даних.

Щоб прискорити процес навчання, LightGBM використовує метод на основі гістограм для вибору найкращого розподілу. Для будь-якої безперервної змінної замість використання окремих значень вони поділяються на комірки чи корзини. Це прискорює процес навчання та знижує використання пам'яті.

Розколи по листах призводять до збільшення складності та можуть призвести до перенавчання. Але це можна подолати, встановивши параметр max_depth, максимальну глибину дерева, яке він розділяє.

Серед переваг LightGBM у порівнянні з іншими алгоритмами можна виділити:

- *Вища швидкість навчання та висока ефективність:* LightGBM використовує алгоритм на основі гістограми. Наприклад, він пакує неперервні набори власних значень в окремі кошики, що швидше під час навчання.
- *Менший обсяг пам'яті:* використання дискретних кошиків для збереження та заміни неперервних значень призводить до меншого використання пам'яті.
- *Більш висока точність (порівняно з будь-яким іншим алгоритмом посилення):* використовує методи листового розщеплення для отримання більш складних дерев, ніж методи розщеплення за рівнями, що є основним фактором для досягнення вищої точності. Тим не менш, це іноді викликає перенавчання, але ми можемо запобігти перенавченню, встановивши параметр max_depth.
- *Великі можливості обробки даних:* у порівнянні з XGBoost, завдяки скороченню часу навчання, він також може обробляти великі дані.
- *Підтримує паралельне навчання*

Бібліотека LightGBM має власний API, хоча можна використовувати метод через класи-оболонки scikit-learn: LGBMRegressor та LGBMClassifier. Це дозволить застосовувати весь набір інструментів із бібліотеки машинного навчання scikit-learn для підготовки даних та оцінки моделей.

Обидві моделі працюють однаково і використовують одні й самі аргументи, що впливають на те, як дерева рішень створюються і додаються в ансамбль. При побудові моделі використовується випадковість. Це означає, що кожного разу, коли алгоритм запускається на тих самих даних, він створює дещо іншу модель.

При використанні алгоритмів машинного навчання зі стохастичним алгоритмом навчання рекомендується оцінювати їх шляхом усереднення їхньої продуктивності за декількома запусками або повтореннями крос-валідації. При підгонці остаточної моделі може бути бажаним або збільшувати кількість дерев до тих пір, поки дисперсія моделі не зменшиться при повторних оцінках, або навчати кілька кінцевих моделей і усереднити їх прогнози (або подати, наприклад, лінійну регресію для ансамблю).

2.2.1. Порівняння продуктивності

У Таблиці 2.2 наведено результати порівняння показників алгоритму LightGBM.

Таблиця 2.2.

Алгоритм	accuracy score	auc score	Час виконання (S)
LightGBM	0.861501	0.764492	0.283759
XGBoost	0.861398	0.764284	2.047220

З наведених вище результатів порівняння продуктивності, точність LightGBM та значення AUC порівняно з XGBoost трохи краще. Проте важливою відмінністю є час виконання процесу навчання моделі. Швидкість навчання LightGBM майже в 7 разів вище, ніж у XGBoost, і різниця стає все більш очевидною зі збільшенням кількості даних навчання.

Це доводить величезні переваги навчання LightGBM для великих наборів даних, особливо в порівнянні з обмеженнями часу.

2.2.2. Гіперпараметри LightGBM

Розглянемо деякі гіперпараметри, важливі для ансамблю LightGBM, а також їхній вплив на продуктивність моделі. У LightGBM є безліч гіперпараметрів, серед яких кількість дерев та їх глибина, швидкість навчання та тип бустингу.

Дослідження типу бустингу

Особливість LightGBM у тому, що він підтримує ряд алгоритмів бустингу, що називаються типами бустингу. Тип бустингу вказується за допомогою аргументу boosting_type та для визначення типу приймає рядок. Можливі значення:

‘gbdt‘: дерево рішень з градієнтним бустингом (GDBT);

‘dart‘: поняття відсіву (dropout) вводиться в MART, отримуємо DART;

‘goss‘: одностороння вибірка на основі градієнта (GOSS).

За замовчуванням використовується GDBT, класичний алгоритм градієнтного бустингу.

DART описаний у статті 2015 року під назвою «DART: Dropouts meet Multiple Additive Regression Trees» і, як випливає з назви, додає поняття dropout з глибокого навчання алгоритм множинних адитивних регресійних дерев (MART), попередник дерев рішень з градієнтним бустингом.

GOSS представлений з роботою по LightGBM та бібліотекою lightgbm. Цей підхід спрямований на використання тільки тих екземплярів, які призводять до великого градієнта помилки, для оновлення моделі та видалення інших екземплярів.

Дослідження кількості дерев

Важливим гіперпараметром для алгоритму ансамблю LightGBM є кількість рішень, що використовуються в ансамблі. Нагадаємо, що дерева прийняття рішень додаються в модель послідовно у спробі виправити та покращити прогнози, зроблені попередніми деревами. Часто працює правило:

більше дерев – краще. Кількість дерев можна задати за допомогою аргументу n_estimators, за умовчанням рівного 100.

Дослідження глибини дерева

Зміна глибини кожного дерева, що додається в ансамбль, ще один важливий гіперпараметр для градієнтного бустингу. Глибина дерева визначає, наскільки кожне дерево спеціалізується на навчальному наборі даних: наскільки може бути загальним чи навченим. Переважні дерева, які не повинні бути надто дрібними та загальними (наприклад, AdaBoost) і не надто глибокими та спеціалізованими (наприклад бутстреп-агрегація).

Градієнтний бустинг зазвичай добре працює з деревами, що мають помірну глибину, що знаходить баланс між навченістю та узагальненістю. Глибина дерева контролюється аргументом max_depth і за замовчуванням використовується невизначене значення, оскільки механізм за замовчуванням для керування складністю дерев полягає у використанні кінцевої кількості вузлів.

Існує два основних способи керування складністю дерева: через максимальну глибину дерева та максимальну кількість термінальних вузлів (листя) дерева.

Дослідження швидкості навчання

Швидкість навчання контролює рівень вкладу кожної моделі в прогнозування ансамблю. Найменші швидкості можуть вимагати більшої кількості дерев рішень в ансамблі. Швидкість навчання можна контролювати за допомогою аргументу learning_rate, за замовчування вона дорівнює 0,1.

Дослідження частки даних кожної ітерації

Параметр випадково вибирає частину даних без повторної вибірки. Використовується для прискорення навчання та боротьби з перенавчанням. Можна контролювати за допомогою аргументу bagging_fraction.

Дослідження частки ознак кожної ітерації

Наприклад, 0,8 означає, що 80% ознак вибираються випадково в кожній ітерації для побудови дерева. Можна контролювати за допомогою аргументу feature_fraction.

РОЗДІЛ 3. ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДУ ГРАДІЄНТНОГО БУСТИНГУ ТА МЕТОДІВ ОПТИМІЗАЦІЇ ГІПЕРПАРАМЕТРІВ ДЛЯ ПРОГНОЗУВАННЯ РАДІОЙОДРЕЗИСТЕНТНОСТІ РАКУ ЩИТОПОДІБНОЇ ЗАЛОЗИ

3.1. Оптимізація гіперпараметрів

При розв'язанні задачі машинного навчання для підвищення якості прогнозування важливо не тільки провести роботу з вихідними даними, а й провести настроювання гіперпараметрів моделі. Гіперпараметри моделі – це параметри моделі, які визначаються до початку процесу навчання моделі.

Алгоритми градієнтного бустингу на відміну, наприклад, від Random Forest, мають велику кількість гіперпараметрів і часто не можуть бути підібрані оптимально за допомогою вже існуючих методів. Є кілька стандартних алгоритмів перебору гіперпараметрів: Grid search, Random Search, Optuna і тд.

Алгоритм GridSearch заснований на ідеї кросвалідації. Вихідна навчальна множина ділиться на три частини: навчальну вибірку, вибірку валідації та тестову вибірку. Для кожного гіперпараметра, що підлягає визначенню, задається набір значень, що розглядаються. Потім в циклі перебираються всі можливі комбінації гіперпараметрів, для кожної комбінації на вибірці навчається модель, що задається поточною комбінацією, після цього на вибірці валідації обчислюється функція втрат. Модель з набором гіперпараметрів, що доставляє мінімум функції втрат на вибірці валідації, є оптимальною. Після цього обчислюється функція втрат для оптимальної моделі на тестовій вибірці. Якщо значення функції втрат на вибірці валідації сильно менше значення функції втрат на тестовій вибірці, це означає, що модель «перенавчена», тобто, вона підлаштувалася під випадково виявлені закономірності у вибірці валідації, яких немає в генеральній сукупності. Якщо значення функції втрат на вибірці

валідації і на тестовій вибірці приблизно рівні, то таку модель можна використовувати.

У кожного з алгоритмів є свої недоліки, наприклад, Random Search вибирає із заданих гіперпараметрів випадкові кількості заданих ітерацій і може не потрапити в оптимальні параметри. Алгоритм Grid search здійснює повний перебір всіх параметрів і на маленьких наборах даних працює непогано, але якщо гіперпараметрів багато для перебору або датасет досить великий, то час перебору параметрів зростає експоненційно, що може бути виправлено за рахунок великих продуктивних потужностей..

Для вирішення проблеми оптимізації гіперпараметрів запропоновано авторський алгоритм підбору гіперпараметрів. Він реалізований у вигляді модуля для автоматичного підбору гіперпараметрів, який працює за кількома сценаріями.

Перший сценарій. Повний перебір гіперпараметрів, який використовується для отримання високої точності та оптимальності моделі. Але на малопотужних комп'ютерах повний вибір гіперпараметрів може зайняти досить великий час.

Другий сценарій. Опція часткового підбору гіперпараметрів. Подібний перебір ґрунтується на послідовному переборі гіперпараметрів. Першим перебирається найважливіший гіперпараметр, другим наступний і так далі. Послідовність параметрів взята з особистого досвіду, а також різноманітних досліджень на тему які з гіперпараметрів більше впливають на навчання. Було створено свій власний список гіперпараметрів, використовуючи такі критерії:

для кращої підгонки

num_leaves: параметр використовується для встановлення кількості листя, що становлять кожне дерево. Теоретичний зв'язок між num_leaves та max_depth: num_leaves=2^ (max_depth). Однак, якщо використовується LightGBM, ця оцінка неправильна: він використовує leaf_wise замість depth_wise, щоб

розділити кінцеві вузли. Тому для num_leaves має бути встановлено менше 2^(max_depth). Інакше він може викликати перенавчання.

min_data_in_leaf: це також дуже важливий параметр для перенавчання. Встановлення його на особливо мале значення може призвести до перенавчання, тому ми повинні встановити його відповідним чином. Для великих наборів даних ми повинні встановити його значення від сотень до тисяч.

max_depth: визначає максимальну глибину кожного дерева або максимальну кількість шарів, які дерево може «виростити».

для більш високої швидкості

bagging_fraction: використовується для швидшого отримання результату;

feature_fraction: встановлює підмножину об'єктів, що використовується для кожної ітерації;

max_bin: чим менше значення max_bin, тим більше часу можна заощадити: коли воно поділяє значення елемента на різні сегменти, це потребує менше обчислювальних потужностей.

для більшої точноісті

num_leaves: встановлення його занадто великим робить дерево більш глибоким і точним, але це призводить до перенавчання, тому не варто встановлювати надто високе значення.

max_bin: ефект установки цього значення вище аналогічного ефекту зростання num_leaves і призведе до уповільнення нашого тренувального процесу.

3.2. Основні етапи роботи та результати авторського алгоритму

Алггоритм складається із 4 етапів. На першому етапі задаються вручну списки гіперпараметрів, спираючись на дослідження, а також статистичні

розподіли даних. Запускається їхній частковий перебір, який виявляє зони оптимальності кожного гіперпараметра.

На другому етапі алгоритму розширена зона оптимальності гіперпараметрів, отримана на першому етапі, використовується для другого етапу підбору. Запускається алгоритм повного перебору гіперпараметрів.

Третім етапом є запуск навчання основної моделі на вже підібраних гіперпараметрах на більшій кількості ітерацій. Під час роботи алгоритму фіксуються значення різних метрик. На графіку оптимальності метрик визначається зона оптимальності, що показує необхідну кількість ітерацій яка згодом передається в модель для реалізації четвертого етапу.

Четвертий етап полягає у тренуванні та подальшому збереженні моделей. Перша модель навчається на тренувальних даних, а будує прогноз на основі валідаційних даних. Наступна модель навчається на тренувальних та валідаційних даних, а будує прогноз на тестових даних.

Запропонований авторський алгоритм суттєво відрізняється від існуючих аналогічних алгоритмів тим, що зону оптимальності має визначити людина. У всіх алгоритмах гіперпараметри вибираються випадковим чином і тому вони можуть не потрапити в зону оптимальності. У запропонованому алгоритмі зону оптимальності бачить людина і визначає параметри за своїми припущеннями. Це дає можливість шукати глобальні мінімуми. Такий алгоритм за досить короткий термін дозволяє отримати оптимальний результат і має кілька різних метрик для порівняння результатів. Метрика використовується при виборі моделей з підібраними гіперпараметрами для збільшення точності самої моделі по даній метриці. Наприклад, коефіцієнт детермінації R2 можна використовувати для порівняння прогнозу з цільовою змінною на валідації. Метрику MAPE можна використовувати для зменшення процентної помилки між прогнозом та цільовою змінною.

Цей алгоритм можна зручно використовувати для ансамблю моделей з різними метриками, наприклад, R2 та MAPE.

Результати роботи

Для дослідження було взято дані 306 пацієнтів. У якості тренувальної вибірки було обрано 183 пацієнти з двох діагностичних груп. До валідаційної вибірки випадковим чином було включено 61 пацієнт. До тестової вибірки увійшло випадковим чином 62 пацієнти.

Таблиця 3.1

	AK	Year	Sex	cytol_P	cytol_M	cytol_S	cytol_B	cytol_F	TNM_T	TNM_N	TNM_M	Histological_P	Histological_F	Histological_S	Histological_O
0	238141	43	1	1	0	0	0	0	3	1	0.000	1	0	0	0
1	245152	22	0	1	0	0	0	0	1	1	0.000	1	1	1	0
2	266479	17	1	1	1	0	0	0	1	1	0.000	1	0	0	0
3	87048	23	1	1	1	1	0	0	1	1	0.000	1	1	0	0
4	148943	18	1	1	0	0	0	0	3	1	0.000	1	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
307	146564	46	0	1	0	0	0	0	3	1	0.000	1	1	0	0
308	153595	63	1	1	1	0	0	0	1	1	0.000	1	0	0	0
309	148587	43	1	1	1	0	0	0	1	1	0.000	1	0	0	0
310	1508113	32	1	1	1	0	0	0	3	1	0.000	1	1	0	0
311	153443	45	0	1	0	0	0	0	2	1	0.000	1	1	0	0

306 rows × 32 columns

gical_O	Inv vessel	Inv tumor	Inv gland	Intrathyroid	Extrathyroid	Oxiphilia	Nekroz	Mts_P	Mts_S	Mts_O	Mts_F	Mts_K	num_mts	Focus	thyroidit	size	classID
0	1	1	1	1	1	1	1	1	1	1	0	0	2	2	1	28.000	1
0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	8.000	1
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	10.000	1
0	0	0	1	1	0	1	0	1	0	0	0	0	1	2	1	18.000	1
1	0	1	1	1	1	1	0	0	1	0	0	0	1	1	0	15.000	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	1	1	0	1	1	0	0	1	0	0	1	1	2	1	0	45.000	0
0	1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	8.000	0
0	0	1	0	1	0	0	0	1	0	0	0	1	12	2	0	13.000	0
0	1	1	1	0	1	1	0	1	1	0	1	0	12	1	1	70.000	0
0	0	1	1	0	0	0	0	1	1	0	1	0	1	2	1	26.000	0

Кореляційні матриці наведено в Додатку 1.

В алгоритмі було використано метрику перевірки якості AUC і отримано наступні результати:

Training set score: 0.8962

Val set score: 0.7869

Test set score: 0.7742

Значення метрики в залежності від кількості ітерацій представлено на рисунку 3.1. Трошки кращий результат був отриманий на 100 ітераціях.

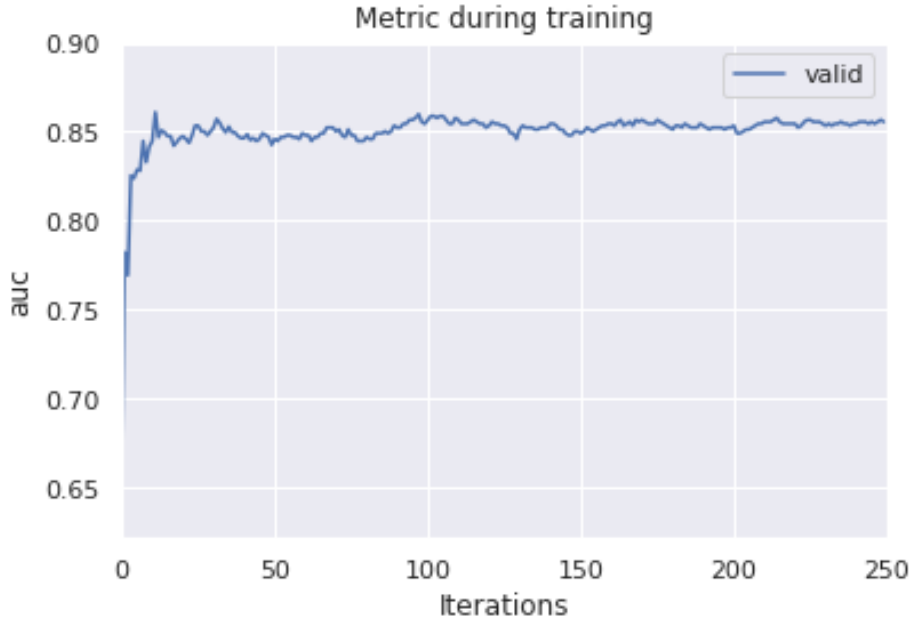


Рис. 3.1. Значення метрики в залежності від кількості ітерацій

Значення логістичної бінарної функції втрат в залежності від кількості ітерацій представлено на рисунку 3.2.

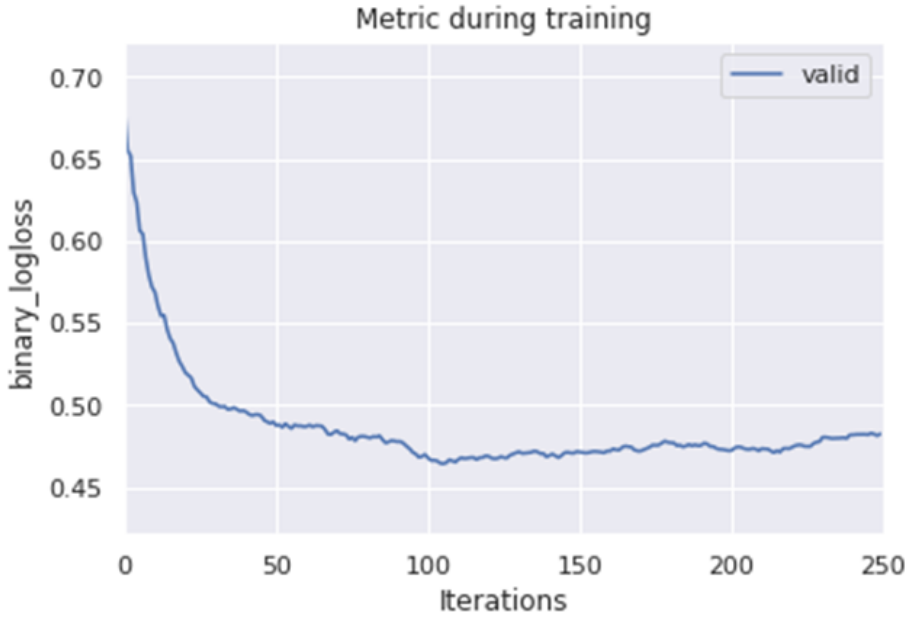
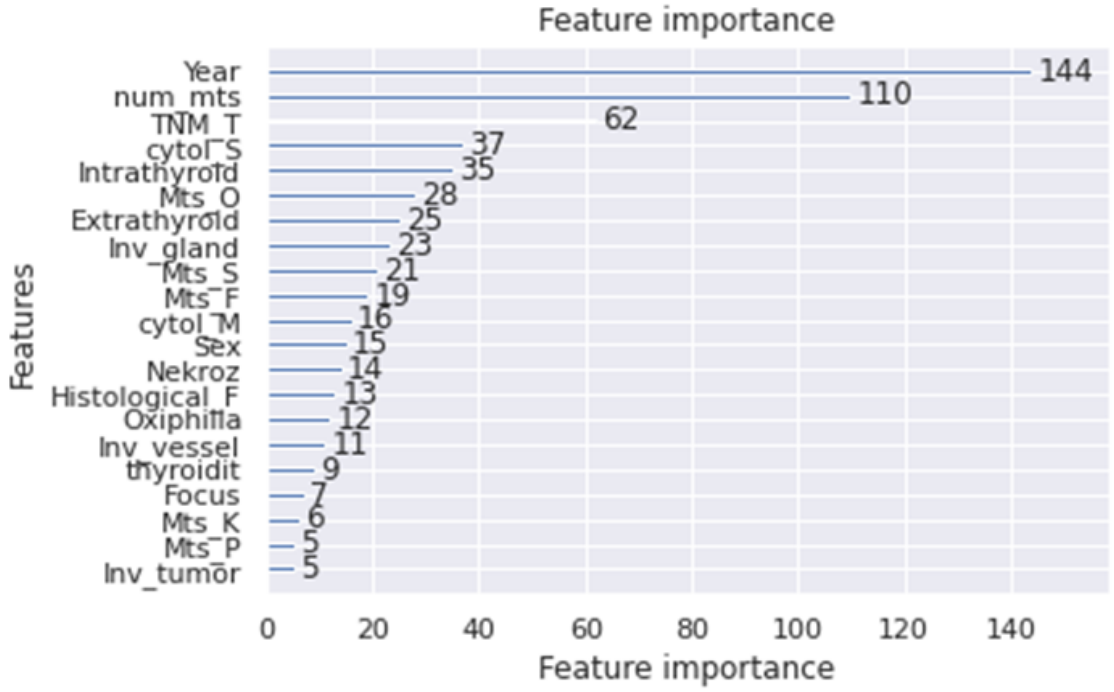


Рис. 3.2. Значення логістичної бінарної функції втрат в залежності від кількості ітерацій

На рисунку 3.3 представлено графік розподілу переходів дерева, тобто важливість ознак для моделі. Найважливішими ознаками визначено вік, кількість метостазів та гістологічна класифікації пухлини.



Відповідно до матриці помилок (Рис. 3.3) визначено що 23 пацієнти з тестової вибірки класифікуються правильно як 1, тобто виліковні йодом. 25 пацієнтів з тестової вибірки класифікуються правильно як 0, тобто невиліковні йодом. 7 пацієнтів з тестової вибірки класифікуються не правильно як 1. 7 пацієнтів з тестової вибірки класифікуються не правильно як 0.

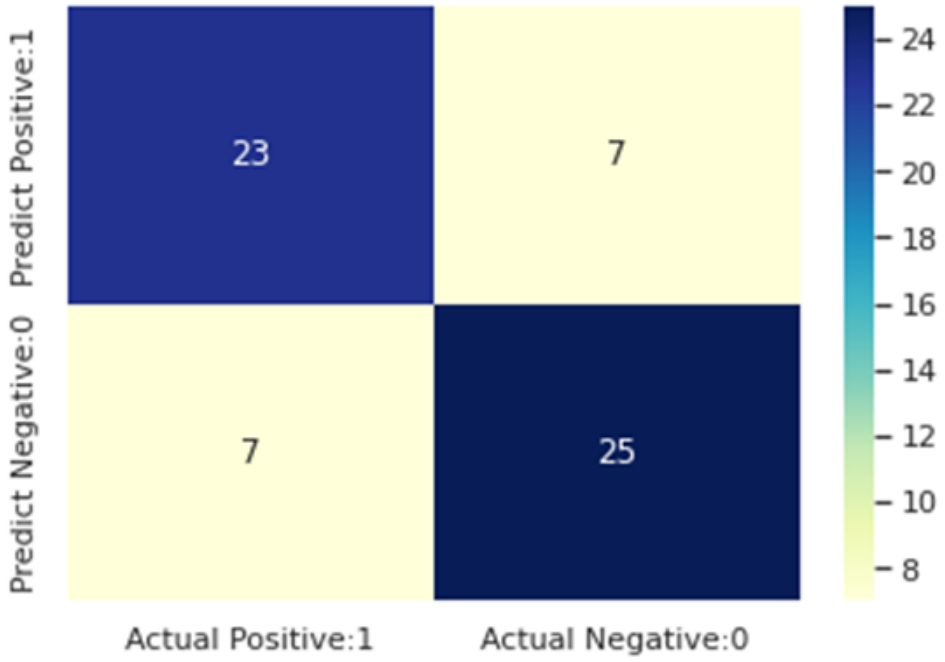


Рис.3.3. Матриця помилок

З таблиці метрик бачимо що f1-score дорівнює 0.77 та 0.78 тобто 77-78% точності. Отримані результати є достатньо високими та показують високу точність класифікації.

Таблиця 3.2.

	precision	recall	f1-score	support
0	0.77	0.77	0.77	30
1	0.78	0.78	0.78	32
accuracy			0.77	62
macro avg	0.77	0.77	0.77	62
weighted avg	0.77	0.77	0.77	62

Також був використаний метод головних компонент для визначення кількості ознак які будуть надавати 100% інформації без надлишку. За його результатами видно, що 29 ознак дають усі 100% інформації про дані, а 25 ознак будуть давати лише 95% інформації. Тобто потрібно використовувати усі ознаки для класифікації. Результати представлено на наступних графіках.

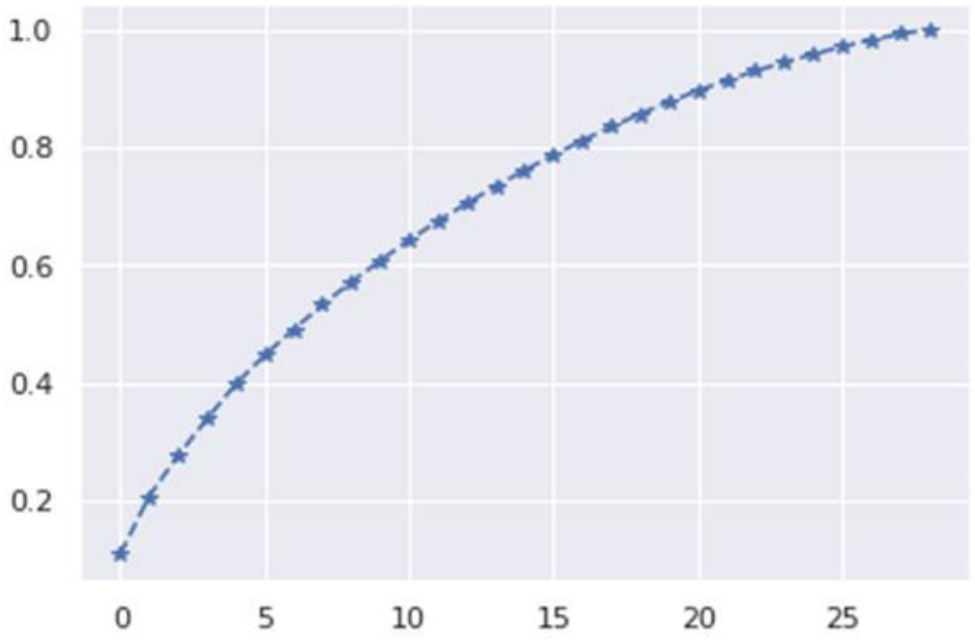


Рис. 3.4. Метод головних компонент

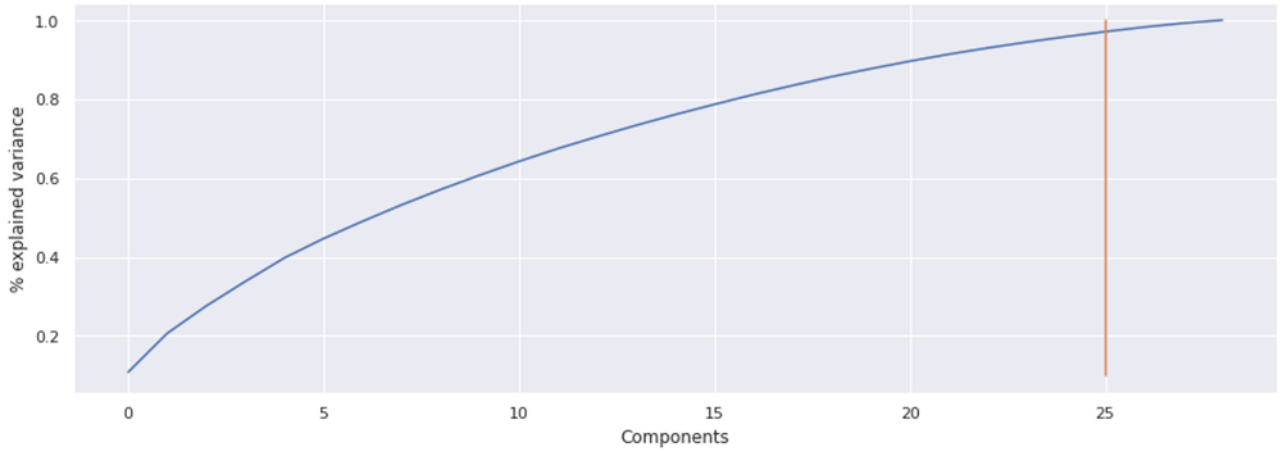


Рис.3.5. Метод головних компонент

Сітка гіперпараметрів для оптимізації на самому початку алгоритму мала такий вигляд:

'num_boost_round': [100, 200, 250, 300, 400],

'max_depth': [2, 3, 4, 6, 8, 10],

'num_leaves': [10, 20, 40, 60, 100],

'feature_fraction': [0.6, 0.7, 0.8, 0.9],

'bagging_fraction': [0.6, 0.7, 0.8, 0.9],

'learning_rate': [0.05, 0.1, 0.2, 0.3]

В результаті оптимізації гіперпараметрів були отримано такі значення:

'num_boost_round': 250,

'max_depth': 2,

'num_leaves': 10,

'feature_fraction': 0.6,

'bagging_fraction': 0.6,

'learning_rate': 0.1,

Можемо бачити, що продуктивність покращується для цього набору даних зі зменшенням максимальної глибини дерев до 2. Також на продуктивність алгоритм вплинуло зменшення кількості листів до 10. Зауважимо, що нижча швидкість навчання призводить до кращої продуктивності на цьому наборі даних і можна вважати, що не буде пропущений глобальний мінімум.

3.3. Основні етапи роботи та результати алгоритму Optuna

Optuna - це алгоритм, який використовується для автоматизованої гіперпараметричної оптимізації моделей машинного навчання. Він дозволяє знайти оптимальні значення гіперпараметрів моделі, максимізуючи або мінімізуючи певну метрику ефективності.

Основою Optuna є Tree-structured Parzen Estimator (TPE). TPE використовує вибірку наборів гіперпараметрів і відповідні їм значення метрики для побудови двох моделей: моделі P(x), що оцінює ймовірність покращення метрики за умови гіперпараметрів x, і моделі P(y|x), що оцінює розподіл метрик при заданих гіперпараметрах x.

Оптимізація гіперпараметрів здійснюється на основі пошуку по дереву, яке побудоване на основі цих моделей. Процес пошуку по дереву складається з трьох етапів:

1. Вибір нового набору гіперпараметрів: Optuna генерує новий набір гіперпараметрів, використовуючи модель P(x), яка оцінює ймовірності покращення метрики. Вибір здійснюється шляхом оптимізації апостеріорної імовірності відносно цільової функції, що вимагає мінімізації функції, яка оцінюється за допомогою моделі P(x).
2. Оцінка нового набору гіперпараметрів: Обчислюється значення метрики для нового набору гіперпараметрів. Значення метрики використовується для оновлення моделі P(y|x) для покращення оцінки розподілу метрик при заданих гіперпараметрах.
3. Оновлення моделей: Моделі P(x) і P(y|x) оновлюються на основі отриманих даних про гіперпараметри і значення метрики.

Оптимізація гіперпараметрів проводиться протягом певної кількості ітерацій або до досягнення певної умови зупинки, наприклад, максимальної кількості спроб.

Основним інтерфейсом Optuna є функція study.optimize(), яка приймає цільову функцію, що обчислює значення метрики для заданого набору гіперпараметрів, і діапазони значень гіперпараметрів.

Optuna є алгоритмом для оптимізації гіперпараметрів за новими критеріями, які базуються на трьох фундаментальних ідеях: define-by-run API, який дозволяє користувачам створювати та маніпулювати пошуковими просторами динамічно; efficient implementation, яка зосереджена на оптимальній функціональності стратегії sampling, а також стратегії pruning; і остання із ідей easy-to-setup, що зосереджується на універсальності, тобто дозволяє оптимізувати функції в легких середовищах, а також широкомасштабні експерименти в середовищах на основі розподілених і паралельних обчислень.

На рисунку 3.6 представлено візуальний опис архітектури Optuna.

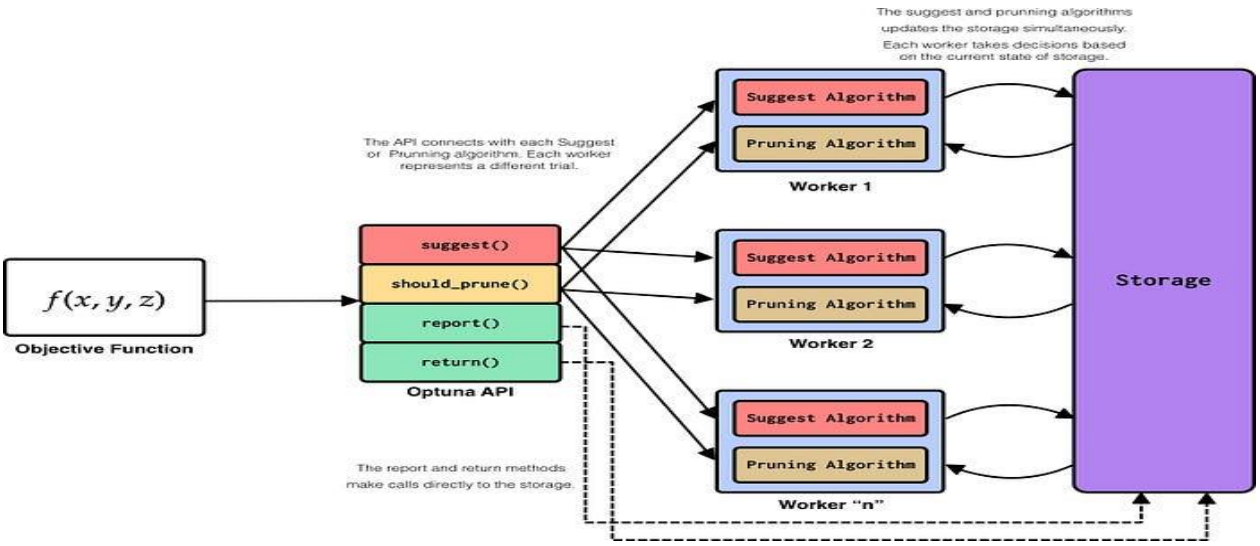


Рис. 3.6. Архітектура Optuna

Результати роботи

В алгоритмі було використано метрику перевірки якості AUC і отримано наступні результати:

Training set score: 0.9344

Val set score: 0.8197

Test set score: 0.7742

Значення метрики в залежності від кількості ітерацій представлено на рисунку 3.7. Кращий результат був отриманий приблизно на 70-ій ітерації, але алгоритм обрав 100 ітерацій як оптимальне число.

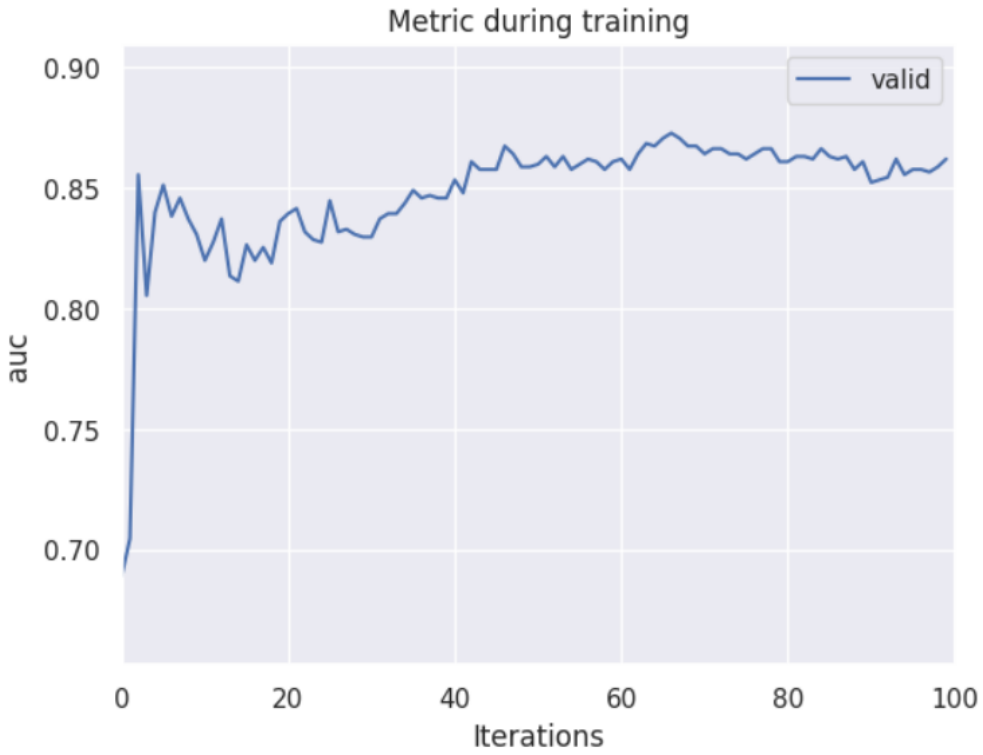


Рис. 3.7. Значення метрики в залежності від кількості ітерацій

Значення логістичної бінарної функції втрат в залежності від кількості ітерацій представлено на рисунку 3.8.

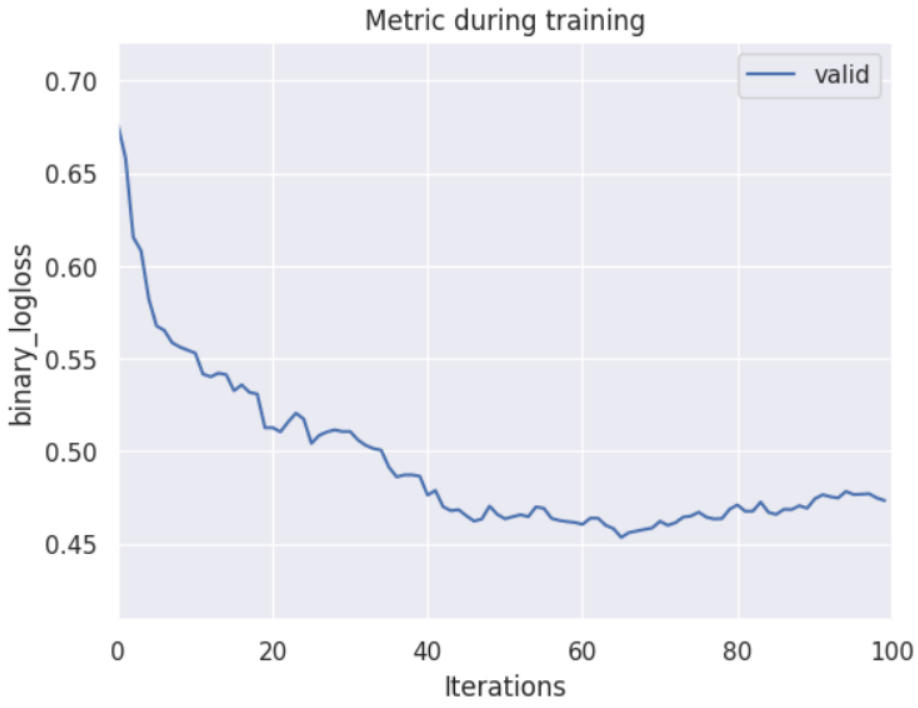


Рис.3.8. Значення логістичної бінарної функції втрат
в залежності від кількості ітерацій

На рисунку 3.9 представлено графік розподілу переходів дерева, тобто важливість ознак для моделі. Найважливішими ознаками визначено вік, кількість метостазів та гістологічна класифікація пухлини.

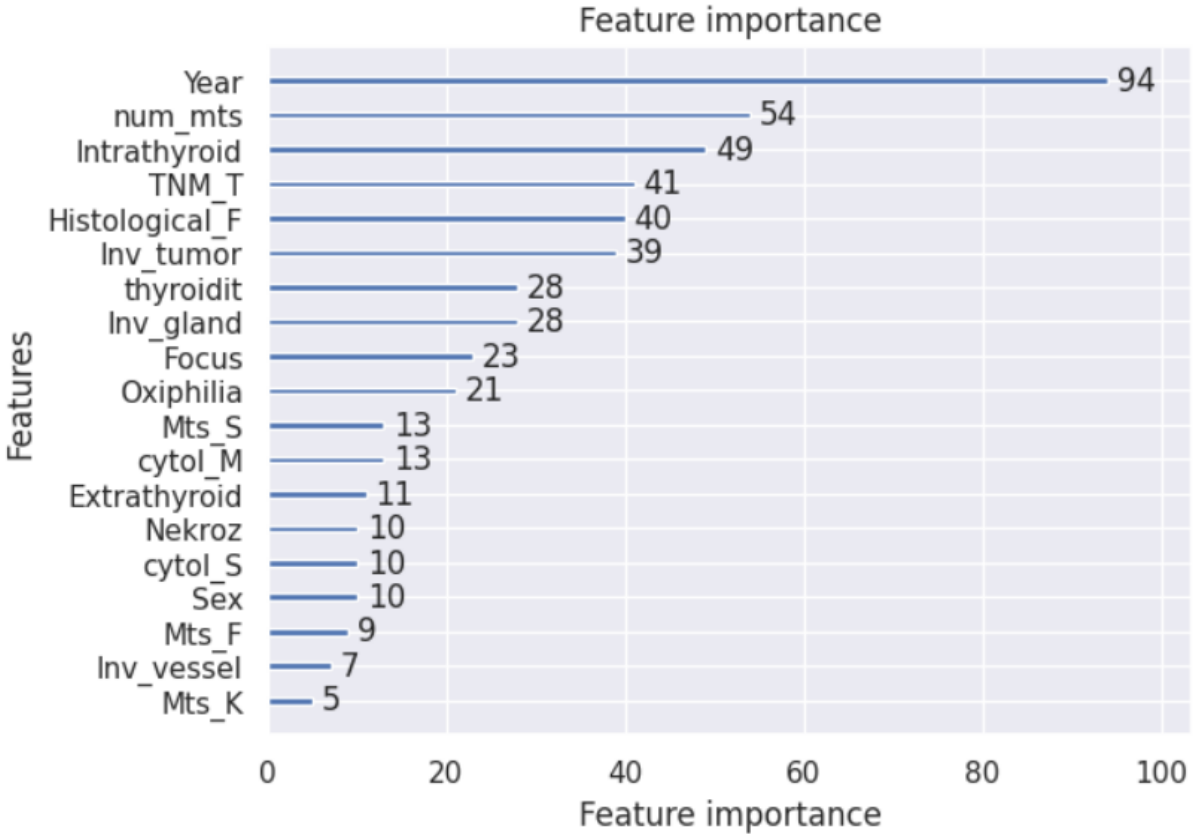


Рис. 3.9. Графік розподілу переходів дерева

Відповідно до матриці помилок (Рис. 3.10) визначено що 21 пацієнт з тестової вибірки класифікуються правильно як 1, тобто виліковні йодом. 27 пацієнтів з тестової вибірки класифікуються правильно як 0, тобто невиліковні йодом. 9 пацієнтів з тестової вибірки класифікуються не правильно як 1. 5 пацієнтів з тестової вибірки класифікуються не правильно як 0.

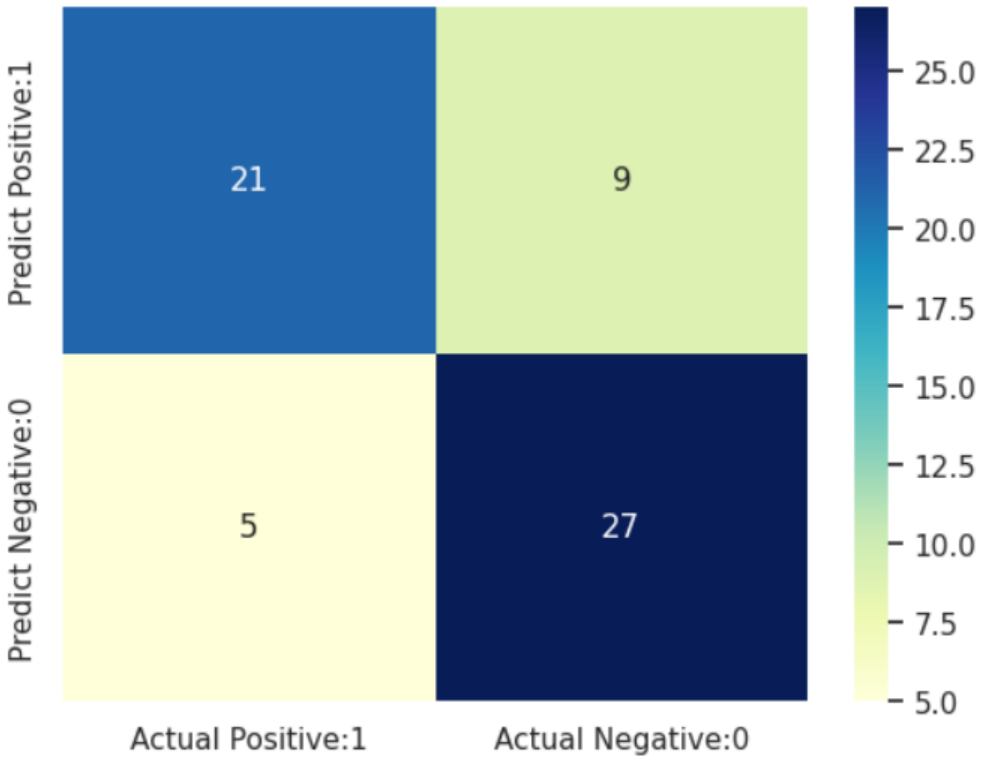


Рис. 3.10. Матриця помилок

З таблиці метрик бачимо що f1-score дорівнює 0.87 та 0.89 тобто 87-89% точності. Отримані результати є достатньо високими та показують високу точність класифікації.

Таблиця 3.3

	precision	recall	f1-score	support
0	0.89	0.87	0.88	157
1	0.87	0.89	0.88	149
accuracy			0.88	306
macro avg	0.88	0.88	0.88	306
weighted avg	0.88	0.88	0.88	306

При використанні метода головних компонент для визначення кількості ознак які будуть надавати 100% інформації без надлишку отримали наступні результати: 26 ознак дають усі 100% інформації про дані, а 25 ознак будуть давати лише 95% інформації. Тобто потрібно використовувати усі ознаки для класифікації (можливо навіть збільшити кількість фічей). Результати представлено на наступних графіках (Рис. 3.11, Рис.3.12).

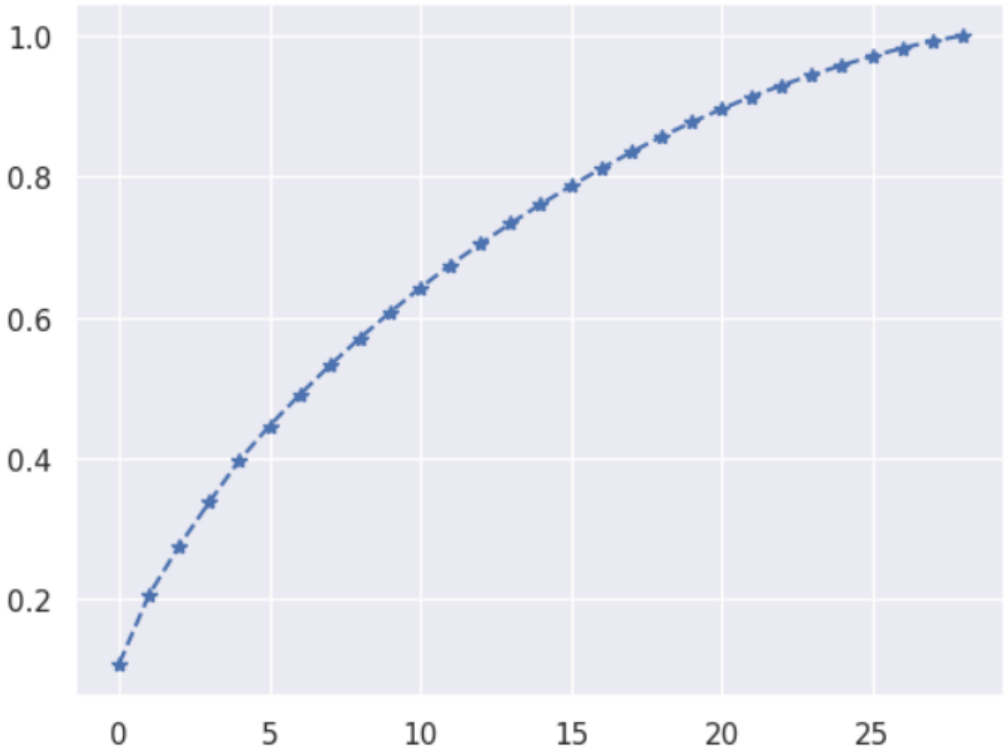


Рис.3.11. Метод головних компонент

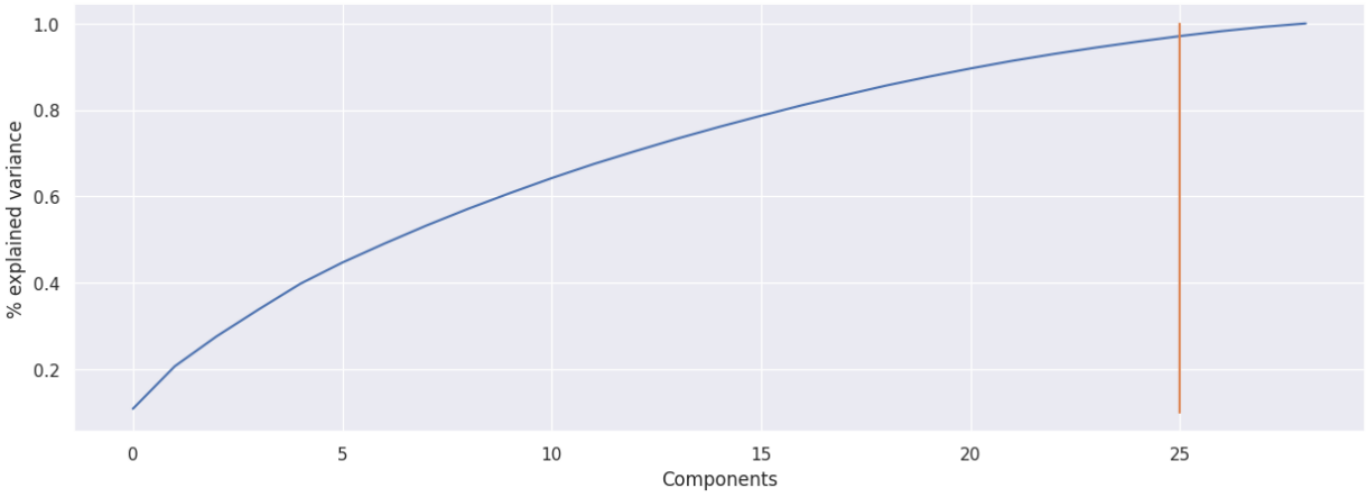


Рис.3.12. Метод головних компонент

Сітка гіперпараметрів для оптимізації на самому початку алгоритму мала такий вигляд:

'num_boost_round': trial.suggest_categorical([100, 200, 300, 500, 700]),

'max_depth': trial.suggest_int('max_depth', 2, 50, step=1),

'num_leaves': trial.suggest_int('num_leaves', 5, 505, step=10),

'feature_fraction': trial.suggest_float("feature_fraction", 0.2, 1, step=0.1),

'bagging_fraction': trial.suggest_float("bagging_fraction", 0.2, 1, step=0.1),

'learning_rate': trial.suggest_float("learning_rate", 0.01, 0.3)

В результаті оптимізації гіперпараметрів були отримано такі значення:

'num_boost_round': 100,

'max_depth': 42,

'num_leaves': 185,

'feature_fraction': 0.3,

'bagging_fraction': 0.2,

'learning_rate': 0.1791387794647433,

Можемо бачити, що продуктивність покращується для цього набору даних із збільшенням максимальної глибини дерев до 42. Також на продуктивність алгоритм вплинуло зменшення кількості листів до 185. Зауважимо, швидкість навчання знаходиться приблизно в середині розподілу що призводить до кращої продуктивності без особливого зменшення швидкості на цьому наборі даних і можна вважати, що не буде пропущений глобальний мінімум.

3.4. Результати порівняння

Порівняння за основною метрикою двох методів оптимізації гіперпараметрів наведене в таблиці 3.4.

Таблиця 3.4

	Авторський алгоритм	Optuna
Training set score	0.8962	0.9344
Val set score	0.7869	0.8197
Test set score	0.7742	0.7742

Бачимо що результати у алгоритма Optuna для тренувального датасету, а також для валідаційного датасету кращі. Тобто для даних які бачить модель, алгоритм Optuna відпрацьовує краще. Але якщо подивитись на метрики тесту, то вона однакова. Тобто дані які моделі не бачать, оптимізовані різними алгоритмами видають однакові метрики.

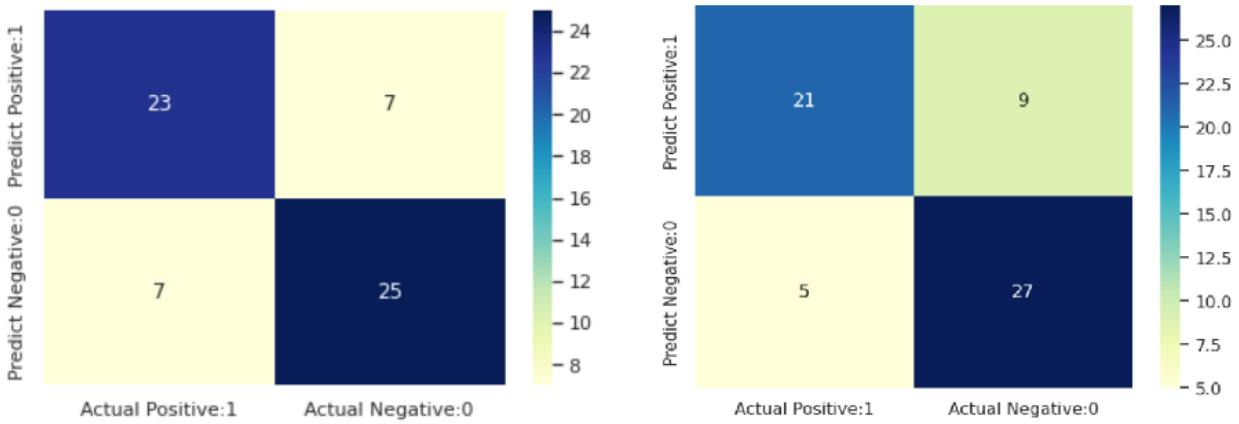


Рис.3.13. Матриці помилок

Дивлячись на матриці помилок можемо встановити, що модель на основі авторського алгоритму (матриця зліва) краще налаштована на розпізнавання маркеру 1. Модель на основі Optuna краще налаштована на розпізнавання маркерів 0.

РОЗДІЛ 4. ПРАКТИЧНЕ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ РАДІОЙОДРЕЗИСТЕНТНОСТІ РАКУ

4.1. Нейронні мережі

Багатошаровий персептрон (Multi-layer Perceptron або MLP) є одним з найпоширеніших алгоритмів нейронних мереж і використовується для вирішення задач класифікації та регресії. У бібліотеці scikit-learn є реалізація MLPClassifier для задач класифікації та MLPRegressor для задач регресії.

MLP складається з кількох шарів нейронів, де кожен нейрон підключений до всіх нейронів на попередньому шарі і наступних шарах. Першим шаром є вхідний шар, останнім - вихідний шар, і між ними можуть бути один або кілька прихованих шарів.

Формула для обчислення вагованої суми (входу) для нейрона на шарі:

$$z = \sum_{i=1}^{n} (x_i * w_i) + b$$

де:

- z - вагова сума
- x_i - вхідний сигнал з попереднього шару
- w_i - вага, що відповідає x_i
- b - зсув (bias)

Після обчислення вагової суми, використовується активаційна функція для отримання виходу нейрона. У MLP можна використовувати різні активаційні функції, такі як логістична (sigmoid), гіперболічний тангенс (tanh) або ReLU (rectified linear unit).

Одна з найпоширеніших активаційних функцій - логістична функція (sigmoid): $f(z) = \frac{1}{(1+e^{-z})}$ і її похідна для зворотнього поширення помилки: $f'(z) = f(z) * (1 - f(z))$

ReLU (Rectified Linear Unit) є однією з найпоширеніших активаційних функцій, що використовується в нейронних мережах, включаючи багатошаровий персептрон (MLP). ReLU є нелінійною функцією активації, яка використовується для введення нелинійності в мережу.

ReLU функція визначається наступним чином:

$$f(z) = \begin{cases} z, \text{якщо } z > 0 \\ 0, \text{інакше} \end{cases}$$

де z є ваговою сумою вхідних сигналів до нейрона.

Одна з головних переваг ReLU полягає у його простоті обчислення та швидкості. Він не має складних обчислень, як у випадку сигмоїди або гіперболічного тангенсу, що робить його більш практичним для використання в глибоких нейронних мережах. Крім того, ReLU допомагає уникнути проблеми "зниклих градієнтів", яка може виникати при використанні інших активаційних функцій. Коли вхідний сигнал менше або дорівнює нулю, похідна ReLU також дорівнює нулю, що може призвести до зупинки зворотного поширення помилки. У випадку зворотного поширення помилки, коли вхідний сигнал більше нуля, похідна ReLU дорівнює одиниці. Це означає, що градієнт проходить без затримки, дозволяючи ефективну передачу градієнту помилки назад по мережі для оновлення ваг. Ще одна важлива особливість ReLU полягає у тому, що він не обмежує значення виходу нейрона від 0 до 1, як у випадку сигмоїди. Це може бути корисним у випадках, коли потрібно отримати виходи нейронів з більш широким діапазоном значень.

Проте, важливо зазначити, що ReLU може мати проблему "мертвих нейронів", коли нейрони можуть бути неактивними (завжди видають нульовий вихід) під час навчання. Якщо вагова сума нейрона завжди менше або дорівнює нулю, то градієнт не буде оновлювати ваги нейрона, і він ніколи не активується. Цю проблему можна вирішити за допомогою інших варіантів ReLU, таких як

Leaky ReLU або Parametric ReLU. У scikit-learn бібліотеці для MLPClassifier і MLPRegressor, активаційна функція ReLU використовується за замовчуванням для прихованих шарів, і вона може бути налаштована за допомогою параметра activation.

Для навчання MLP використовується алгоритм зворотного поширення помилки (backpropagation), який зменшує помилку прогнозування, оновлюючи ваги нейронів зворотним напрямком через мережу.

Алгоритм зворотного поширення помилки базується на градієнтному спуску і використовує функцію втрат для оцінки помилки прогнозування. У випадку класифікації може використовуватися функція втрат, наприклад, категоріальна кросс-ентропія або середньоквадратична помилка для задач регресії.

Для оновлення ваг нейронів використовується градієнтний спуск, який змінює ваги у напрямку, протилежному до градієнту функції втрат. Формула оновлення ваг має вигляд: $\Delta w_{ij} = \eta * \frac{\partial E}{\partial w_{ij}}$ де:

- Δw_{ij}- зміна ваги між нейронами і та j
- η - швидкість навчання (learning rate)
- $\frac{\partial E}{\partial w_{ij}}$- похідна функції втрат E по вазі w_{ij}

Для оновлення ваг використовується правило градієнтного спуску, наприклад: $w_{ij} = w_{ij} - \Delta w_{ij}$

Процес навчання в MLP триває кілька епох, де на кожній епосі вхідні дані подаються на вхід мережі, робиться прямий прохід (forward pass), обчислюються вихідні значення, порівнюються з очікуваними значеннями та виконується зворотнє поширення помилки для оновлення ваг.

У scikit-learn MLP можна налаштувати шляхом вибору кількості шарів, кількості нейронів у кожному шарі, функції активації, швидкості навчання та інших параметрів, щоб досягти найкращих результатів в конкретній задачі.

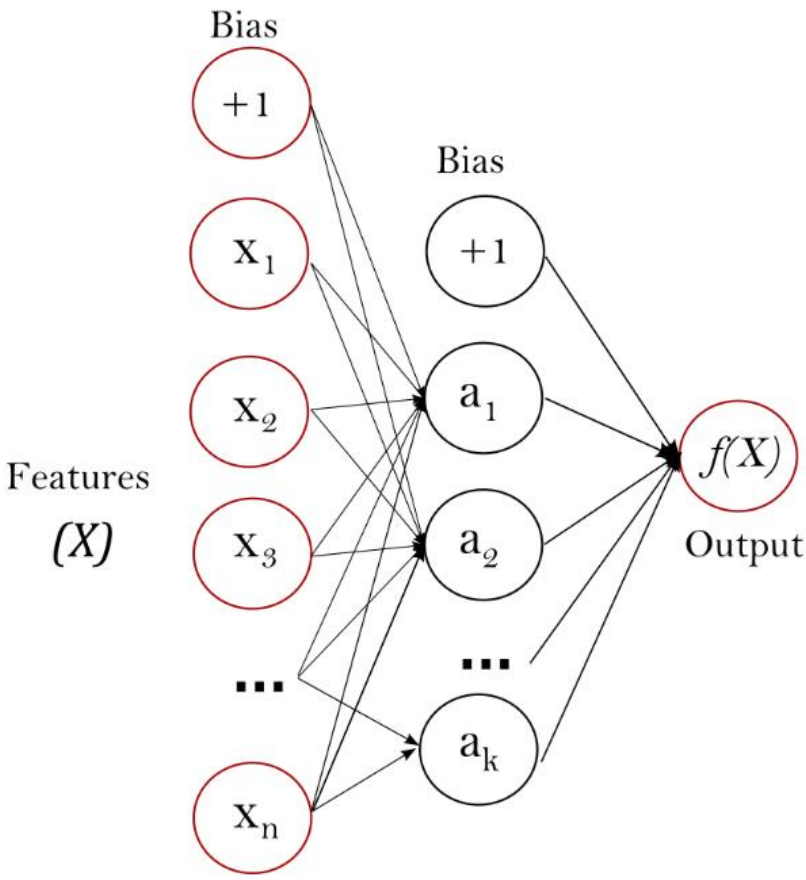


Рис.4.1. Один прихований шар MLP [22]

Результати роботи

В нашому дослідженні застосуємо нейронну мережу Multi-layer Perceptron для вирішення задачі прогнозування радіойодрезистентності раку.

Застосуємо класифікатор з такими параметрами:

MLPClassifier(alpha=1e-05, hidden_layer_sizes=(150, 100, 50), max_iter=300, random_state=1, activation = 'relu').

hidden_layer_sizes – відповідає за кількість прихованих шарів нейронної мережі та кількість нейронів у кожному шарі

activation – функція активації

max_iter – кількість ітерацій, проходів по нейронній мережі

В алгоритмі було використано метрику перевірки якості AUC і отримано наступні результати:

Training set score: 0.9727

Val set score: 0.7705

Test set score: 0.7581

Відповідно до матриці помилок визначено що 24 пацієнт з тестової вибірки класифікуються правильно як 1, тобто виліковні йодом. 23 пацієнтів з тестової вибірки класифікуються правильно як 0, тобто невиліковні йодом. 6 пацієнтів з тестової вибірки класифікуються не правильно як 1. 9 пацієнтів з тестової вибірки класифікуються не правильно як 0.

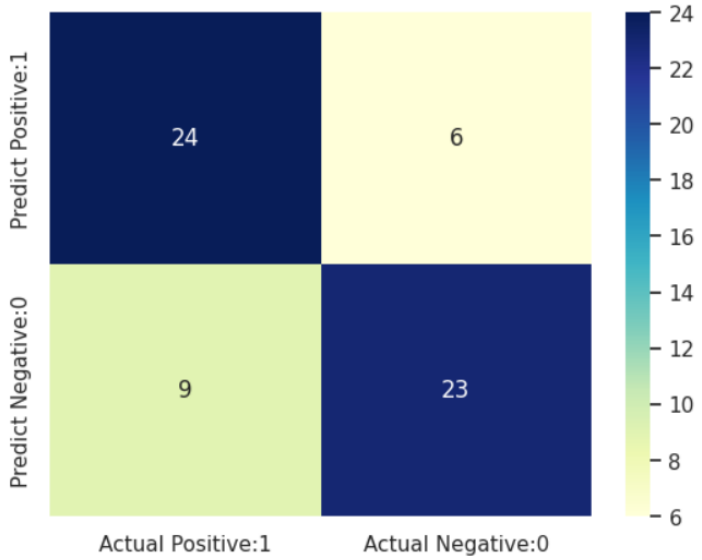


Рис.4.2. Матриця помилок

З таблиці метрик (Таблиця 4.1) бачимо, що f1-score дорівнює 0.89 тобто 89% точності. Отримані результати є достатньо високими та показують високу точність класифікації.

Таблиця 4.1

	precision	recall	f1-score	support
0	0.89	0.89	0.89	157
1	0.89	0.89	0.89	149
accuracy			0.89	306

macro avg	0.89	0.89	0.89	306
weighted avg	0.89	0.89	0.89	306

Можемо бачити, що продуктивність даного алгоритму достатньо висока. Він видає 75 % точності на тестовому датасеті

4.2. Порівняння алгоритмів градієнтного бустингу з нейронною мережею

Порівняння за основною метрикою двох алгоритмів для прогнозування агресивності раку щитоподібної залози наведено в Таблиці 4.2.

Таблиця 4.2

	LGBM + Optuna	MLP
Training set score	0.9344	0.9727
Val set score	0.8197	0.7705
Test set score	0.7742	0.7581

Результати роботи алгоритма LGBM та алгоритма оптимізації optuna для тренувального датасету гірше, але для валідаційного датасету та тестового кращі. Тобто для тренувальних даних нейронна мережа MLP гарно підлаштовується. Метрика тесту практично однакова, на 2% краще у алгоритму LGBM. Тобто LGBM та алгоритм оптимізації Optuna видає більш точний результат з меншим використанням ресурсів та часу. Використання MLP включає тренування на GPU, коли LGBM можна тренувати на CPU без великої втрати продуктивності.

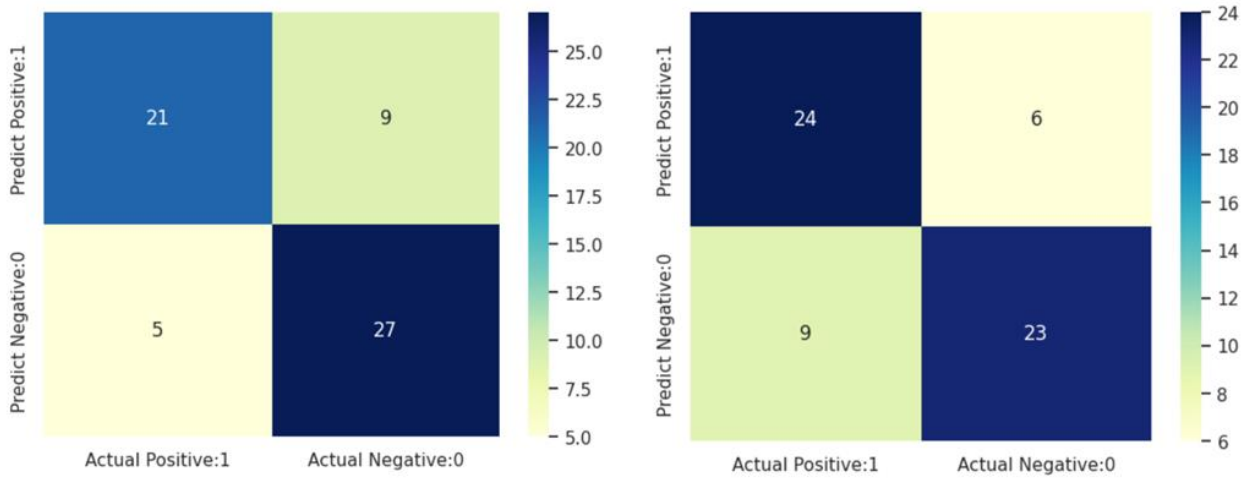


Рис.4.3. Матриці помилок

За матрицями помилок можемо встановити, що модель на основі алгоритму LGBM та алгоритму оптимізації Optuna (рисунок ліворуч) краще налаштована на розпізнавання маркеру 0, але й помилка більше в маркері 0. Модель на основі нейронної мережі MLP (рисунок праворуч) практично однаково налаштована на розпізнавання маркерів 0 та 1, помилка більше в маркера які справді дорівнюють 1.

ВИСНОВКИ

Результатом дипломної роботи є проведене дослідження та побудова методу у сфері комп'ютерної діагностики агресивності раку щитоподібної залози на основі даних 306 пацієнтів, які мають 29 характеристик (ознак), що можуть бути показниками агресії злоякісних пухлин. Ознаки включають комплекс цитологічних та гістологічних характеристик пухлин щитоподібної залози як факторів розвитку їх радіойодрезистентності. Це стало основою розробки комплексного методу прогнозування радіойодрезистентності раку щитоподібної залози.

В результаті проведеного дослідження виявлено, що найбільш значимими характеристиками для класифікації пацієнтів за допомогою запропонованого алгоритму є наступні ознаки: вік, кількість метостазів та гістологічні класифікації пухлини.

Проведено порівняння двох методів машинного навчання, метод градієнтного бустингу LGBM та нейронної мережі MLP. В результаті виявлено, що метод градієнтного бустингу видає більш точні результати на тестовій виборці за менший період часу та при меншому використанні ресурсів.

Отриманий метод діагностики раку на основі градієнтного бустингу має наступні показники результатів контролю: 77% – відсоток правильних діагнозів, 23% – відсоток неправильних діагнозів. В результаті оптимізації гіперпараметрів авторським методом та методом Optuna було визначено їх оптимальні значення, а саме максимальної глибини дерев, кількості листків, швидкості навчання, кількості дерев, частка ознак, що використовуються під час навчання та частка даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Онкологія в Україні: рівень захворюваності та смертності в 2019-2020 роках. Електронний ресурс: https://www.slovoidilo.ua/2021/05/14/ infografika /suspilstvo/onkolohiya-ukrayini-riven-zaxvoryuvanosti-ta-smertnosti-2019-2020-rokax
2. Bogdanova T, Zurnadzhy L, Nikiforov Y Histopathological features of papillary thyroid carcinomas detected during four screening examinations of a Ukrainian-American cohort. Br J Cancer. 2015; 113:1556–1564.
3. Drozd V, Branovan I, Shiglik N Thyroid Cancer Induction: Nitrates as Independent Risk Factors or Risk Modulators after Radiation Exposure, with a Focus on the Chernobyl Acciden. Eur Thyroid J. 2018; 7:67–74.
4. Darre T, Amana B, Pegbessou E et al. Descriptive epidemiology of thyroid cancer in Togo. Asian Pac J Cancer Prev. 2015; 16(15):6715-6717.
5. Lubitz CC, Kong CY, McMahon PM et al. Annual financial impact of well-differentiated thyroid cancer care in the United States. Cancer. 2014; 120(9):1345-1352.
6. Mehra S, Tuttle RM, Milas M et al. Database and registry research in thyroid cancer: striving for a new and improved national thyroid cancerdatabase. Thyroid. 2015; 2:157-168.
7. Dzepina D, Zurak K, Petric V, Cupic H. Pathological characteristics and clinical perspectives of papillary thyroid cancer: study of 714 patients. Eur Arch Otorhinolarygol. 2014; 271(1):141-148.
8. Lei S, Ding Z, Ge J, Zhao D. Association between prognostic factors and clinical outcome of well-differentiated thyroid carcinoma: a retrospective 10-year follow-up study. Oncol Lett. 2015; 10(3):1749-1754

9. Markovina S, Grigsby PW, Schwarz J. K et al. Treatment approach, surveillance, and outcome of well-differentiated thyroid cancer in childhood and adolescence. Thyroid. 2014; 24(7):1121-1126.
10. Van Nostrand D. Radioiodine Refractory Differentiated Thyroid Cancer: Time to Update the Classifications. Thyroid. 2018 Sep; 28(9): 1083-93. doi:10.1089/thy.2018.0048.
11. Зелінська ГВ, Коваленко АЄ, Остафійчук МВ, Кваченюк АМ, Устименко АЯ, Кулініченко ГМ, et al. Цитоморфологічні особливості папілярного раку ЩЗ з розвитком радіойодрезистентності. Український радіологічний та онкологічний журнал. 2021; 29(3):76-68. doi: 10.46879/ukroj.3.2021.76-88.
12. Ortiz S, Rodríguez JM, Soria T, Pérez-Flores D, Piñero A, Moreno J, Parrilla P. Extrathyroid spread in papillary carcinoma of the thyroid: clinicopathological and prognostic study. Otolaryngol Head Neck Surg. 2001 Mar;124(3):261-5.
13. Wreesmann VB, Nixon IJ, Rivera M, Katabi N, Palmer F, Ganly I, Shaha AR, Tuttle RM, Shah JP, Patel SG, Ghossein RA. Prognostic value of vascular invasion in well-differentiated papillary thyroid carcinoma. Thyroid. 2015;May; 25(5):503-8. doi: 10.1089/thy.2015.0052..
14. Lin JD, Chao TC, Hsueh C, Kuo SF. High recurrent rate of multicentric papillary thyroid carcinoma. Ann Surg Oncol. 2009 Sep; 16(9):2609-16. doi: 10.1245/s10434-009-0565-7.
15. Lars A. Akslen M.D., Virginia A. LiVolsi M.D. Prognostic significance of histologic grading compared with subclassification of papillary thyroid carcinoma. Cancer. 2000;Apr;88(8):1902–1908. https://doi.org/10.1002/(SICI)1097-0142(20000415)88:8<1902::AID-CNCR20>3.0.CO;2-Y

16. Rivera M, Ghossein R, Schoder H, et al. Histopathologic characterization of radioactive iodine-refractory fluorodeoxyglucose-positron emission tomography-positive thyroid carcinoma. Cancer. 2008; 113:48-56.
17. Deandreis D, Ghuzlan A, Leboulleux S, et al. Do histological, immunohistochemical, and metabolic (radioiodine and fluorodeoxyglucose uptakes) patterns of metastatic thyroid cancer correlate with patient outcome? Endocr Relat Cancer. 2011; 1:159-169.
18. Friedman J. H.. Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics v. 29, p.1189-1232, 1999
19. Appel Ron, Thomas J Fuchs, Piotr Dollar, and Pietro Perona. Quickly boosting decision trees-pruning underachieving features early. In ICML (3), pages 594–602, 2013. Sochman J., Matas J. AdaBoost. Center for Machine Perception Czech Technical University, Prague.
20. Zhang Zixuan Boosting Algorithms Explained Theory, Implementation, and Visualization. Published in Towards Data Science. https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30.
21. Guolin Ke LightGBM: a highly efficient gradient boosting decision tree. – NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing SystemsDecember 2017. P. 3149–3157.
22. Neural network models. Multi-layer Perceptron. https://scikit-learn.org/stable/modules/neural_networks_supervised.html

ДОДАТКИ

Кореляційна матриця

	AK	Year	Sex	cytol_P	cytol_M	cytol_S	cytol_B	cytol_F	TNM_T	TNM_N	TNM_M	Histological_P	Histological_F	Histological_S	Histological_O	Inv vessel	Inv tumor	Inv gland	Intrathyroid	Extrathyroid	Oxiphilia	Nekroz	Mts_P	Mts_S	Mts_O	Mts_F	Mts_K	num_mts	Focus	thyroidit	size	classID
AK	1.000000	0.009525	0.056385	-0.029714	0.224710	0.052489	-0.031917	0.052238	-0.073613	-0.053552	-0.038096	0.144984	-0.088713	0.016695	0.068870	-0.144430	0.050737	0.046997	-0.008070	0.021941	0.171791	0.056343	0.027905	0.113532	0.046118	0.088643	0.004748	0.001892	-0.104651	0.164430	0.143933	0.127412
Year	0.009525	1.000000	0.002414	-0.124470	0.017098	0.030365	0.106835	0.030041	-0.008801	-0.052372	0.009032	-0.074195	-0.229137	-0.206625	-0.042550	-0.239391	-0.368746	-0.150718	-0.196035	-0.187084	-0.133301	-0.139183	0.013662	-0.159284	-0.045314	-0.121775	-0.062518	-0.054818	0.095939	-0.056671	0.077644	-0.017159
Sex	0.056385	0.002414	1.000000	0.006034	0.122168	-0.030258	0.008824	-0.071706	0.124856	0.024046	-0.024941	0.062007	-0.141286	0.049584	0.082016	-0.045233	0.010973	0.082022	-0.010329	0.108042	0.092622	0.058993	-0.043404	0.044832	0.118873	-0.073883	-0.025161	0.045533	0.024951	0.103494	-0.018664	0.032454
cytol_P	-0.029714	-0.124470	0.006034	1.000000	-0.256474	-0.397596	-0.372846	-0.047026	0.133674	0.034623	-0.235233	0.086454	0.097477	0.019484	-0.126153	0.034684	0.110778	0.043474	0.062438	0.138648	-0.111849	-0.076200	0.088696	0.064983	-0.093180	-0.023961	-0.090064	-0.007403	0.126121	-0.065092	-0.032520	0.135137
cytol_M	0.224710	0.017098	0.122168	-0.256474	1.000000	0.198839	0.042911	0.054561	-0.066252	0.015450	0.017807	0.032383	0.009549	0.057705	-0.020407	-0.055333	-0.038203	0.000885	0.001183	-0.035782	0.044575	-0.003844	0.035642	0.001323	0.100462	0.115593	0.068782	0.169764	0.010890	0.052771	-0.048823	-0.014706
cytol_S	0.052489	0.030365	-0.030258	-0.397596	0.198839	1.000000	-0.045568	-0.085749	-0.182919	-0.109270	-0.035381	0.068272	0.039864	-0.009349	0.071886	-0.127886	0.020353	0.048677	0.058089	-0.141813	0.019774	0.015103	-0.063116	0.051971	0.068124	0.054616	0.020788	-0.021003	-0.106966	0.049662	-0.115619	-0.240406
cytol_B	-0.031917	0.106835	0.008824	-0.372846	0.042911	-0.045568	1.000000	0.081274	-0.034722	0.026556	0.248736	-0.252046	-0.003932	-0.034083	0.218512	-0.011528	-0.087734	-0.067430	-0.012367	-0.037257	0.122130	0.124429	-0.310323	0.028039	0.137903	-0.000838	-0.049258	-0.079970	-0.074087	0.022992	0.144442	0.132300
cytol_F	0.052238	0.030041	-0.071706	-0.047026	0.054561	-0.085749	0.081274	1.000000	0.041580	-0.099944	0.120536	-0.029803	0.027346	-0.006414	-0.053107	0.012761	0.061083	0.033392	0.011158	0.038578	0.013565	0.010361	-0.067651	0.035652	0.021992	-0.015775	-0.049912	-0.059573	0.058702	-0.073246	0.102791	0.020615
TNM_T	-0.073613	-0.008801	0.124856	0.133674	-0.066252	-0.182919	-0.034722	0.041580	1.000000	0.055508	0.004113	-0.030203	0.065096	-0.009310	-0.088292	0.239591	0.044454	0.162454	-0.002455	0.585518	-0.065063	-0.002027	0.066264	-0.020340	-0.074880	-0.029171	0.007200	0.096582	0.071873	-0.130898	0.314268	0.062169
TNM_N	-0.053552	-0.052372	0.024046	0.034623	0.015450	-0.109270	0.026556	-0.099944	0.055508	1.000000	0.252869	0.144679	-0.035530	0.008173	-0.009668	0.023036	-0.031135	-0.016210	-0.021666	0.074095	-0.032924	0.033006	0.011494	0.078746	0.065390	0.132340	0.078746	0.144677	-0.040519	-0.007235	0.056057	-0.051191
TNM_M	-0.038096	0.009032	-0.024941	-0.235233	0.017807	-0.035381	0.248736	0.120536	0.004113	0.252869	1.000000	0.019146	-0.084024	-0.026464	0.136835	0.044398	-0.082276	-0.025699	-0.022996	-0.000879	0.110544	0.074413	-0.081093	0.061319	0.198327	-0.050197	0.061418	-0.063060	-0.057142	0.074753	0.101256	0.102127
Histological_P	0.144984	-0.074195	0.062007	0.086454	0.032383	0.068272	-0.252046	-0.029803	-0.030203	0.144679	0.019146	1.000000	-0.229250	-0.090987	0.042282	-0.074582	0.108255	0.072853	0.103255	-0.011048	0.014447	0.065991	0.117322	0.021160	0.061283	-0.033683	0.021160	0.038784	-0.010888	0.056098	0.017055	-0.092853
Histological_F	-0.088713	-0.229137	-0.141286	0.097477	0.009549	0.039864	-0.003932	0.027346	0.065096	-0.035530	-0.084024	-0.229250	1.000000	0.018767	-0.152695	0.234322	0.188944	0.126144	0.095240	0.214064	-0.129352	-0.114462	-0.089762	0.036073	-0.175307	0.416939	0.075850	0.121085	-0.023678	-0.103457	-0.038448	-0.223454
Histological_S	0.016695	-0.206625	0.049584	0.019484	0.057705	-0.009349	-0.034083	-0.006414	-0.009310	0.008173	-0.026464	-0.090987	0.018767	1.000000	0.068657	0.029566	0.111127	0.020805	0.080987	0.091440	0.121277	-0.003954	0.040422	0.136829	0.053652	-0.100406	-0.061418	0.016856	0.092807	0.104007	-0.076550	0.139183
Histological_O	0.068870	-0.042550	0.082016	-0.126153	-0.020407	0.071886	0.218512	-0.053107	-0.088292	-0.009668	0.136835	0.042282	-0.152695	0.068657	1.000000	-0.038470	0.105791	0.048604	0.023024	-0.023579	0.421623	0.180726	-0.302939	0.244619	0.310228	-0.110376	-0.083685	-0.089862	0.018928	0.119732	0.056672	0.068309
Inv vessel	-0.144430	-0.239391	-0.045233	0.034684	-0.055333	-0.127886	-0.011528	0.012761	0.239591	0.023036	0.044398	-0.074582	0.234322	0.029566	-0.038470	1.000000	0.138375	0.189114	0.160180	0.323195	0.098261	0.038208	0.019959	0.072312	0.035482	0.072720	0.072312	0.214174	0.064268	-0.100509	0.144830	-0.147660
Inv tumor	0.050737	-0.368746	0.010973	0.110778	-0.038203	0.020353	-0.087734	0.061083	0.044454	-0.031135	-0.082276	0.108255	0.188944	0.111127	0.105791	0.138375	1.000000	0.202245	0.171427	0.258221	0.262223	0.124799	-0.108947	0.223183	0.178289	0.251215	0.158449	0.126355	-0.081530	0.034214	0.003313	-0.182307
Inv gland	0.046997	-0.150718	0.082022	0.043474	0.000885	0.048677	-0.067430	0.033392	0.162454	-0.016210	-0.025699	0.072853	0.126144	0.020805	0.048604	0.189114	0.202245	1.000000	0.341119	0.173970	0.145046	0.069736	0.030503	0.079218	0.115924	0.055969	0.059558	0.128891	0.075572	0.088822	0.010790	-0.152900
Intrathyroid	-0.008070	-0.196035	-0.010329	0.062438	0.001183	0.058089	-0.012367	0.011158	-0.002455	-0.021666	-0.022996	0.103255	0.095240	0.080987	0.023024	0.160180	0.171427	0.341119	1.000000	0.165300	0.175116	0.057412	-0.067680	0.050041	0.147340	0.167457	0.128871	0.037593	0.045234	0.078698	-0.103084	-0.075067
Extrathyroid	0.021941	-0.187084	0.108042	0.138648	-0.035782	-0.141813	-0.037257	0.038578	0.585518	0.074095	-0.000879	-0.011048	0.214064	0.091440	-0.023579	0.323195	0.258221	0.173970	0.165300	1.000000	0.086584	0.005170	-0.005633	0.204737	0.037781	0.139892	0.080309	0.173588	0.086726	-0.069116	0.141265	-0.126650
Oxiphilia	0.171791	-0.133301	0.092622	-0.111849	0.044575	0.019774	0.122130	0.013565	-0.065063	-0.032924	0.110544	0.014447	-0.129352	0.121277	0.421623	0.098261	0.262223	0.145046	0.175116	0.086584	1.000000	0.135840	-0.104745	0.208819	0.417116	-0.102766	0.065065	-0.093774	-0.002659	0.212843	0.085008	0.021575
Nekroz	0.056343	-0.139183	0.058993	-0.076200	-0.003844	0.015103	0.124429	0.010361	-0.002027	0.033006	0.074413	0.065991	-0.114462	-0.003954	0.180726	0.038208	0.124799	0.069736	0.057412	0.005170	0.135840	1.000000	0.037653	0.029513	0.224902	-0.065989	0.157611	-0.021841	-0.097577	-0.003300	0.207497	0.115769
Mts_P	0.027905	0.013662	-0.043404	0.088696	0.035642	-0.063116	-0.310323	-0.067651	0.066264	0.011494	-0.081093	0.117322	-0.089762	0.040422	-0.302939	0.019959	-0.108947	0.030503	-0.067680	-0.005633	-0.104745	0.037653	1.000000	-0.311287	-0.206892	-0.197755	0.023289	0.069959	0.108525	-0.115270	0.036485	0.091988
Mts_S	0.113532	-0.159284	0.044832	0.064983	0.001323	0.051971	0.028039	0.035652	-0.020340	0.078746	0.061319	0.021160	0.036073	0.136829	0.244619	0.072312	0.223183	0.079218	0.050041	0.204737	0.208819	0.029513	-0.311287	1.000000	0.388535	0.148673	-0.028522	0.087911	0.097840	0.076603	0.021701	-0.097838
Mts_O	0.046118	-0.045314	0.118873	-0.093180	0.100462	0.068124	0.137903	0.021992	-0.074880	0.065390	0.198327	0.061283	-0.175307	0.053652	0.310228	0.035482	0.178289	0.115924	0.147340	0.037781	0.417116	0.224902	-0.206892	0.388535	1.000000	0.093811	0.116627	0.020230	0.027433	0.222257	0.032646	0.030976
Mts_F	0.088643	-0.121775	-0.073883	-0.023961	0.115593	0.054616	-0.000838	-0.015775	-0.029171	0.132340	-0.050197	-0.033683	0.416939	-0.100406	-0.110376	0.072720	0.251215	0.055969	0.167457	0.139892	-0.102766	-0.065989	-0.197755	0.148673	0.093811	1.000000	0.294956	0.228911	-0.045137	0.058472	-0.073276	-0.247077
Mts_K	0.004748	-0.062518	-0.025161	-0.090064	0.068782	0.020788	-0.049258	-0.049912	0.007200	0.078746	0.061418	0.021160	0.075850	-0.061418	-0.083685	0.072312	0.158449	0.059558	0.128871	0.080309	0.065065	0.157611	0.023289	-0.028522	0.116627	0.294956	1.000000	0.256040	-0.038246	0.013421	0.029843	-0.176262
num_mts	0.001892	-0.054818	0.045533	-0.007403	0.169764	-0.021003	-0.079970	-0.059573	0.096582	0.144677	-0.063060	0.038784	0.121085	0.016856	-0.089862	0.214174	0.126355	0.128891	0.037593	0.173588	-0.093774	-0.021841	0.069959	0.087911	0.020230	0.228911	0.256040	1.000000	0.120063	-0.030808	0.096743	-0.359552
Focus	-0.104651	0.095939	0.024951	0.126121	0.010890	-0.106966	-0.074087	0.058702	0.071873	-0.040519	-0.057142	-0.010888	-0.023678	0.092807	0.018928	0.064268	-0.081530	0.075572	0.045234	0.086726	-0.002659	-0.097577	0.108525	0.097840	0.027433	-0.045137	-0.038246	0.120063	1.000000	0.031023	-0.043016	-0.030364
thyroidit	0.164430	-0.056671	0.103494	-0.065092	0.052771	0.049662	0.022992	-0.073246	-0.130898	-0.007235	0.074753	0.056098	-0.103457	0.104007	0.119732	-0.100509	0.034214	0.088822	0.078698	-0.069116	0.212843	-0.003300	-0.115270	0.076603	0.222257	0.058472	0.013421	-0.030808	0.031023	1.000000	-0.004076	0.066997
size	0.143933	0.077644	-0.018664	-0.032520	-0.048823	-0.115619	0.144442	0.102791	0.314268	0.056057	0.101256	0.017055	-0.038448	-0.076550	0.056672	0.144830	0.003313	0.010790	-0.103084	0.141265	0.085008	0.207497	0.036485	0.021701	0.032646	-0.073276	0.029843	0.096743	-0.043016	-0.004076	1.000000	0.193828
classID	0.127412	-0.017159	0.032454	0.135137	-0.014708	-0.240406	0.132300	0.020615	0.062169	-0.051191	0.102127	-0.092853	-0.223454	0.139183	0.068309	-0.147660	-0.182307	-0.152900	-0.075067	-0.126650	0.021575	0.115769	0.091988	-0.097838	0.030976	-0.247077	-0.176262	-0.359552	-0.030364	0.066997	0.193828	1.000000

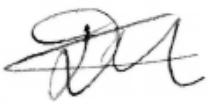
ВІДГУК
на кваліфікаційну роботу бакалавра
«Прогнозування результатів лікування раку методами машинного навчання»
студента 4-го курсу спеціальності 113 Прикладна математика
факультету комп'ютерних наук та кібернетики
Київського національного університету імені Тараса Шевченка
Киричека Миколи Павловича

Важливим сучасним напрямком боротьби з раком є розробка способів прогнозування його появи та поведінки з метою використання адекватних ефективних методів лікування та моніторингу злоякісних пухлин.

В дипломній роботі студентом Киричеком М.П. проведено дослідження та розроблено метод комп'ютерної діагностики агресивності раку щитоподібної залози на основі аналізу 29 характеристик 306 пацієнтів. Отриманий комплексний метод прогнозування радіойодрезистентності раку щитоподібної залози базується на цитологічних та гістологічних характеристиках пухлин, які корелюють із радіойодрезистентністю.

Було розглянуто декілька моделей прогнозування радіойодрезистентності раку щитоподібної залози, розроблено застосунок для програмного представлення задач прогнозування раку, виконано тестування програмного засобу на наборах даних реальних пацієнтів, проведене порівняння швидкодії та точності методів. Порівняння двох методів машинного навчання показало, що метод градієнтного бустингу LGBM є точнішим на тестовій вибірці та ефективнішим у використанні ресурсів.

Під час роботи над дипломом студент продемонстрував здатність до самостійного мислення та вміння застосовувати теоретичний матеріал в практичних задачах. Таким чином, на підставі аналізу кваліфікаційної роботи бакалавра Киричека Миколи Павловича «Прогнозування результатів лікування раку методами машинного навчання» можна стверджувати, що робота виконана на належному науково-методологічному рівні, відповідає вимогам вищої школи, які висуваються до такого роду робіт і заслуговує на оцінку «відмінно», а її автор заслуговує на присвоєння кваліфікації бакалавра.

Асистент кафедри обчислювальної математики
Факультету комп'ютерних наук та кібернетики
Київського національного університету
імені Тараса Шевченка  Сергій ДЕНИСОВ

РЕЦЕНЗІЯ
на кваліфікаційну роботу бакалавра
«Прогнозування результатів лікування раку методами машинного навчання»
студента 4-го курсу спеціальності 113 Прикладна математика
факультету комп'ютерних наук та кібернетики
Київського національного університету імені Тараса Шевченка
Киричека Миколи Павловича

Киричек М.П. вдало обґрунтував актуальність теми дослідження, наголошуючи на загрозі онкологічних захворювань та необхідності ранньої діагностики раку. Зазначено, що радіойодрезистентні метастази становлять серйозну проблему у лікуванні раку щитоподібної залози. Відзначається, що розробка методів прогнозування та діагностики радіойодрезистентних метастазів може сприяти використанню ефективних методів лікування та моніторингу цих пухлин.

Мета дослідження, яку сформульовано в роботі, є чіткою і спрямованою на дослідження вкористання методів машинного навчання для прогнозування радіойодрезистентності раку щитоподібної залози та розробку програмного забезпечення. Студент грамотно підійшов до подання необхідних теоретичних аспектів – висвітлив основні поняття, чітко навів постановку задачі, побудову моделі та дослідив поведінку алгоритмів. У другій частині роботи автор проводить обчислювальні експерименти на основі аналізу 29 характеристик у 306 пацієнтів та демонструє результати досліджень за допомогою таблиць та графіків.

Висновки кваліфікаційної роботи підсумовують проведене дослідження та розробку методу комп'ютерної діагностики агресивності раку щитоподібної залози. Зазначено значимість віку пацієнта, кількості метастазів та гістологічної класифікації пухлини у прогнозуванні радіойодрезистентності. Також підкреслено перевагу методу градієнтного бустингу LGBM у порівнянні з нейронною мережею MLP. Висновки також вказують на досягнуті результати контролю з високою точністю діагнозів та оптимальні значення гіперпараметрів, визначених за допомогою оптимізації.

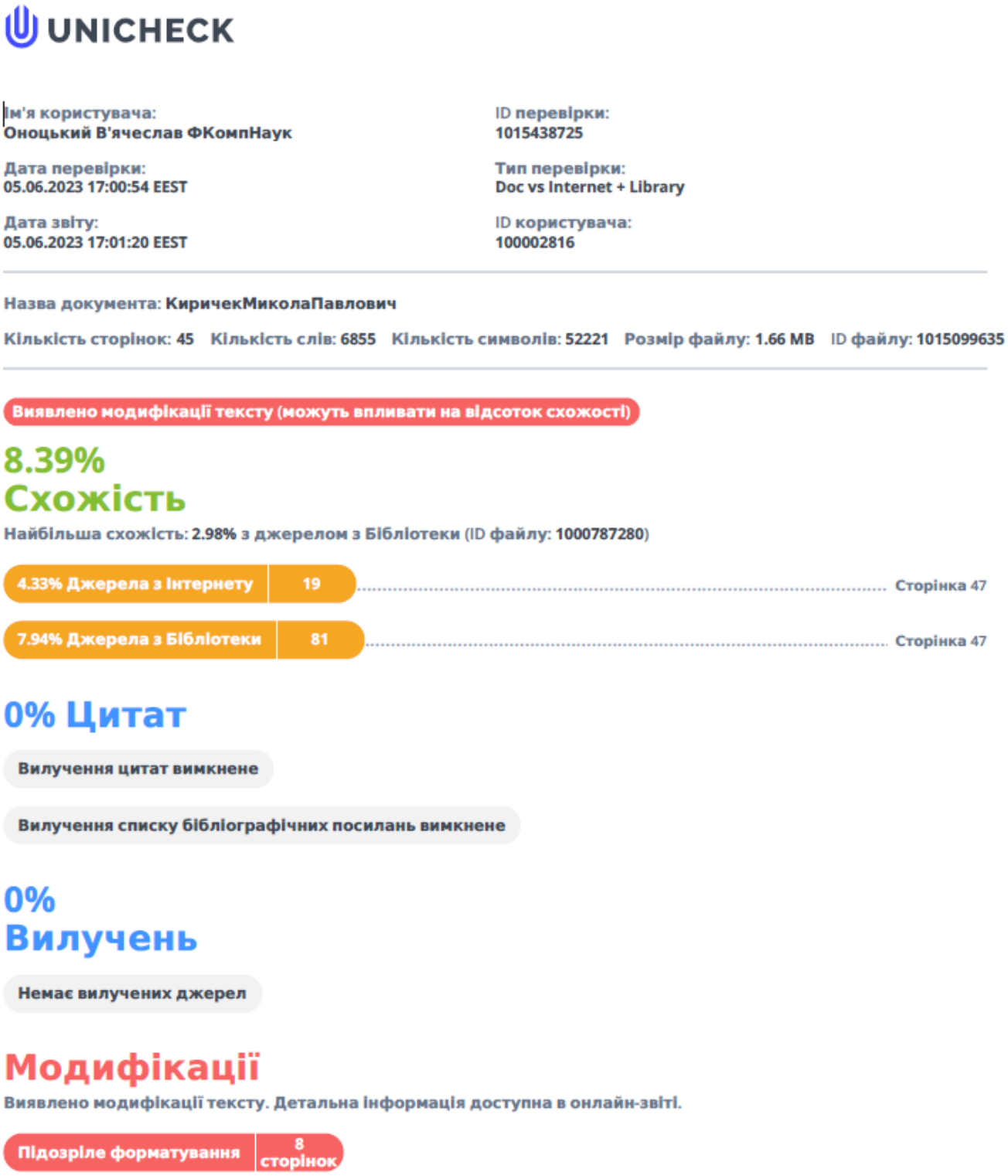
Таким чином, на підставі аналізу кваліфікаційної роботи бакалавра можна стверджувати, що робота виконана на високому науково-методологічному рівні, відповідає вимогам вищої школи, які висуваються до такого роду робіт і заслуговує на оцінку «відмінно», а її автор, Киричек Микола Павлович, заслуговує на присвоєння кваліфікації «бакалавр».

Рецензент:
доктор фізико-математичних наук,
професор, член-кореспондент НАН України,
завідувач кафедри обчислювальної математики
факультету комп'ютерних наук та кібернетики
Київського національного університету
імені Тараса Шевченка С.І.Ляшко

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

СИСТЕМА ЗАПОБІГАННЯ ТА ВИЯВЛЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ

Довідка про оригінальність кваліфікаційної роботи за освітнім рівнем бакалавр



Експертна оцінка роботи науковим керівником : Робота виконана самостійно та не містить відомостей без посилань на джерела. Найбільш схоже джерело – наукова стаття з стандартними означеннями та базовими фактами.

Науковий керівник: _____ Денисов С.В.
(підпис)

Оператор: _____ Оноцький В.В.
(підпис)