Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Магістерська кваліфікаційна робота на тему:

"ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ХВОРИХ НА ІНСУЛЬТ"

Виконав: студент 2 курсу, групи 2ІСТ-22м
спеціальності 126 – «Інформаційні системи та технології»
_____ Євгеній ГОНТКОВСЬКИЙ
Керівник: к.т.н., доц. каф. САІТ
_____ Олексій КОЗАЧКО
«08» 12 2023 р.
Рецензент: к.т.н., доц. каф. КН
_____ Володимир ОЗЕРАНСЬКИЙ
«11» 12 2023 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«08» 12 2023 р.

Вінниця ВНТУ – 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – ІІ-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 126 Інформаційні системи та технології
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
д.т.н., проф. Мокін В. Б.
«04» 09 2023 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Гонтковському Євгенію Юрійовичу

1. Тема роботи: "Інформаційна технологія передбачення хворих на інсульт", керівник роботи: Олексій КОЗАЧКО, к.т.н., доц. каф. САІТ, затверджені наказом закладу вищої освіти від «18» 09 2023 року №247
2. Строк подання студентом роботи «30» 11 2023 року
3. Вихідні дані до роботи:
   Датасет «Kaggle Stroke Prediction Dataset» з даними для передбачення ймовірності інсульту у пацієнтів.
   https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
4. Зміст текстової частини:
   1) Актуальність проблеми;
   2) Аналіз бібліотек для вирішення задачі;
   3) Підготовка даних;
   4) Побудова моделей машинного навчання
   5) Економічна частина.
5. Перелік ілюстративного матеріалу:
   1) Кореляційна матриця показників;
   2) Конфузійна матриця Decision Tree;
   3) Конфузійна матриця Random Forest;
   4) Конфузійна матриця Ada Boost;
   5) Конфузійна матриця SVM;
   6) Конфузійна матриця XG Boost;
   7) Конфузійна матриця Gradient Boosting;
   8) Діаграма порівняння результатів кожної моделі.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Наталія БУРЄННІКОВА, д.е.н., проф.каф. ЕПВМ	25.10.23	10.11.23

7. Дата видачі завдання «04» 09 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	04.09	20.09	
2	Вибір оптимальних інформаційних технологій та проведення розвідувального аналізу	20.09	05.10	
3	Розробка моделі класифікації	05.10	25.10	
4	Економічна частина	25.10	10.11	
5	Оформлення матеріалів до захисту МКР	10.11	30.11	

Студент _____ Євгеній ГОНТКОВСЬК

Керівник роботи _____ Олексій КОЗАЧКО

АНОТАЦІЯ

УДК 004.9:616.831-005

Гонтковський Є.Ю. Інформаційна технологія передбачення хворих на інсульт. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2023. с.103

На укр. мові. Бібліогр.: 32 назв; рис.: 61 ; табл.: 12.

У магістерській кваліфікаційній роботі розроблено інформаційну технологію передбачення хворих на інсульт. Під час розробки було обрано датасет, виконано підготовку даних та розвідувальний аналіз, де визначено, яка група людей є найбільш вразливою до інсульту, розроблено інформаційну технологію, та побудовано моделі передбачення.

Ілюстративна частина складається з 8 плакатів із розвідувальним аналізом та результатом тестування моделей.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної технології передбачення хворих на інсульт.

Ключові слова: Python, інсульт, розвідувальний аналіз, хвороба.

ABSTRACT

Hontkovskyi Y.Y. Information technology for predicting stroke patients. Master's thesis in specialty 126 - Information systems and technologies, educational and professional program - Information technologies of data and image analysis. Vinnytsia: VNTU, 2023. 103 p.

In Ukrainian. Bibliogr.: 32 titles; figs: 61 ; tables: 12.

During the master's thesis, an information technology for predicting strokes was developed. A dataset was chosen, data preparation and exploratory analysis were performed, identifying the most vulnerable group to strokes. An information technology was developed, and prediction models were built.

The illustrative part consists of 8 posters with reconnaissance analysis and the results of model testing.

In the economic section, the question of the feasibility of developing and implementing information technology for predicting stroke patients is considered.

Keywords: Python, stroke, intelligence analysis, disease.

ЗМІСТ

ВСТУП

Актуальність теми. Інформаційна технологія передбачення хворих на інсульт займає важливе місце серед актуальних проблем сучасної медицини та інформаційних технологій. З урахуванням того, що інсульт є однією з найпоширеніших і найнебезпечніших неврологічних патологій, його своєчасне виявлення та передбачення стає важливим завданням для забезпечення ефективної медичної допомоги та попередження негативних наслідків.

Актуальність теми обумовлена необхідністю вдосконалення методів діагностики та прогнозування інсульту, адже час відкриття та невідкладного лікування має ключове значення для підвищення шансів на повне відновлення хворого. Інформаційні технології, такі як штучний інтелект, машинне навчання та аналіз великих обсягів даних, надають унікальні можливості для створення точних та ефективних систем передбачення ризику інсульту.

Мета і завдання роботи. Метою дослідження є підвищення точності передбачення хворих на інсульт, за рахунок використання методів машинного навчання. Для досягнення мети в роботі поставленні та розв’язанні такі завдання:

– проаналізувати існуючі методи передбачення інсульту;
– підготувати дані та провести розвідувальний аналіз;
– побудувати моделі та виконати прогнозування;
– оцінити результати роботи моделей.

Об'єктом дослідження є процес розроблення інформаційної технології передбачення хворих на інсульт методами машинного навчання.

Предметом дослідження є методи інформаційної технології передбачення хворих на інсульт.

Методи дослідження. У дослідженнях було використано методи розвідувального аналізу даних та побудова моделей машинного навчання.

Новизна одержаних результатів полягає в подальшому розвитку інформаційної технології передбачення хворих на інсульт, яка на відміну від існуючих, використовує ансамбль моделей машинного навчання та дозволяє підвищити точність передбачення хворих на інсульт.

Практичне значення Отримані дані мають велике значення в медичній галузі для людей, що страждають на інсульт. Вони також можуть бути використані як фундамент для створення ефективних методів раннього виявлення цього захворювання.

Апробація результатів магістерської кваліфікаційної роботи. Результати кваліфікаційної роботи апробовані на LIII Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2023-2024 рр.).

Публікації результатів магістерської кваліфікаційної роботи. За результатами даної роботи опубліковано тези на LIII науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2023-2024 рр.), на тему «Інформаційна технологія передбачення хворих на інсульт» [1].

1 ХАРАКТЕРИСТИКА ПРОБЛЕМИ

1.1 Актуальність проблеми

Інсульт, також відомий як церебральний інсульт або апоплексія, є серйозним медичним захворюванням, що виникає внаслідок порушення кровопостачання мозку. Це стан, при якому частина мозку не отримує достатньо крові, що може призвести до ураження нейронів та втрати функцій.

Інсульти класифікуються на два основні типи: ішемічні та геморагічні. У випадку ішемічного інсульту, мозкові судини блокуються тромбами або емболами, забороняючи нормальний кровотік. Геморагічний інсульт виникає внаслідок розриву судини, що призводить до витоку крові в мозкову тканину.

Основні симптоми інсульту включають втрату чутливості або руху в обличчі, руках чи ногах, проблеми з мовленням, нагло виникаючі головні болі, а також втрату рівноваги чи координації. Важливо визнати ознаки інсульту якнайшвидше, оскільки невідкладне лікування може значно зменшити ризик ураження мозкової тканини та покращити прогноз для пацієнта [2].

У середньовіччі вчені, такі як Авіценна та Маймонід, почали робити перші спроби класифікації інсультів та визначення їхніх симптомів. Однак важливі аспекти цієї хвороби залишалися маловивченими до початку Нового часу .

У XIX столітті зростання медичних досліджень та розвиток технологій дозволили докладніше вивчити механізми інсульту. Один з перших класифікацій інсультів був представлений Рудольфом Вірховом у 1856 році. Також, в цей період, Франц Нойхер провів важливі дослідження щодо взаємодії серця та судин [2].

У XX столітті, з розвитком образної діагностики, такої як комп'ютерна томографія (КТ) та магнітно-резонансна томографія (МРТ), з'явилися можливості детально досліджувати мозкову тканину та визначати об'єм ураження .

Інсульт є серйозною медичною проблемою, і для ефективного розуміння та управління цією хворобою важливо визначити основні фактори, які сприяють її виникненню. Розглянемо ключові аспекти, що визначають ризик інсульту, зокрема вік, артеріальний тиск, серцево-судинні захворювання, діабет, куріння, генетичні фактори та інші, які можуть варіюватися в контексті різних груп населення [2].

Вік є одним з найважливіших факторів ризику для розвитку інсульту. Інсульт частіше вражає людей похилого віку. Це пов'язано з природним старінням організму, яке може супроводжуватися атеросклерозом (затвердінням артерій), підвищенням артеріального тиску та іншими чинниками, які сприяють виникненню інсульту.

Високий артеріальний тиск (гіпертонія) є ключовим фактором ризику інсульту. Постійний підвищений тиск може пошкоджувати стінки артерій, сприяти формуванню згущень та втраті їх еластичності. Це може призводити до утворення тромбів та емболів, які можуть блокувати кровообіг і викликати інсульт .

Діабет є фактором ризику для інсульту, оскільки впливає на кровоносну систему. Високий рівень цукру в крові може пошкоджувати артерії та венозні судини, що збільшує ризик утворення тромбів та атеросклерозу [2].

Куріння є модифікованим фактором ризику інсульту, оскільки тютюновий дим містить речовини, які можуть пошкоджувати судини та сприяти утворенню плащин, які можуть викликати інсульт.

Брак фізичної активності та сидячий спосіб життя можуть сприяти накопиченню зайвої ваги, гіпертонії та інших факторів ризику.

Іншим важливим фактором ризику є наявність серцево-судинних захворювань. Аритмії, такі як фібриляція передсердь, можуть призводити до утворення тромбів, які подорожують до мозку та викликають інсульт. Також, ураження клапанів серця або інші аномалії можуть сприяти формуванню емболів [2].

Розуміння різноманітності факторів ризику серед різних груп населення дозволяє вдосконалювати підходи до лікування та сприяє розробці глобальних стратегій громадського здоров'я

Інсульт залишається високопоширеним захворюванням в Україні, впливаючи на здоров'я та якість життя населення. За даними Національного інституту серця та судин України, щорічно в Україні фіксується тисячі випадків інсульту. Протягом останніх років відзначається збільшення кількості випадків серед різних вікових груп, зокрема серед молодого населення.

Згідно з дослідженнями, основні фактори ризику інсульту в Україні включають підвищений рівень артеріального тиску, недостатню фізичну активність, високий рівень куріння та несвоєчасне лікування хронічних захворювань, таких як діабет [2].

1.2 Аналіз можливостей мови програмування Python

Python є однією з найпопулярніших мов програмування, що визначається своєю простотою, читабельністю коду та універсальністю застосувань. У цьому розділі розглянемо плюси та мінуси використання Python у різних сферах [3].

Однією з найвизначальніших особливостей Python є його простий та читабельний синтаксис. Висловлення можуть бути виражені лаконічно, що робить код більш зрозумілим та менш помилковим. Наприклад, використання пробілів для визначення блоків коду замість фігурних дужок сприяє вирізненню структури програми .

Python підтримує об'єктно-орієнтоване програмування, що дає можливість створювати класи та об'єкти. Це сприяє структуруванню коду, полегшує його розширення та забезпечує використання парадигми ООП [3].

Python має величезну кількість стандартних бібліотек, які розширюють його функціональність. Додатково, існують різні фреймворки, такі як Django

для веб-розробки, Flask для створення легших веб-застосунків, та бібліотеки для наукових обчислень, такі як NumPy та Pandas [3].

Python використовується в широкому спектрі галузей, включаючи веб-розробку, наукові дослідження, обробку даних, машинне навчання, автоматизацію та багато інших. Django та Flask використовуються для розробки веб-застосунків, NumPy і Pandas для обробки та аналізу даних, а TensorFlow та PyTorch для машинного навчання .

Плюси: Простий синтаксис, широкі можливості бібліотек, динамічна типізація, підтримка ООП, крос-платформеність, активна спільнота.

Мінуси: Менша швидкодія, GIL може обмежити багатопотоковість, обмежена ефективність на мобільних платформах .

Хоча Python дозволяє ефективно вирішувати багато завдань, його швидкодія не завжди рівня іншим компільованим мовам. Присутність Global Interpreter Lock (GIL) може обмежувати використання багатопотоковості та призводити до проблем в високонавантажених програмах .

Python залишається популярним завдяки поєднанню простоти та потужності. Запровадження нововведень, таких як удосконалення GIL та вдосконалення продуктивності, вказують на те, що майбутнє мови буде залишатися яскравим та перспективним [3].

1.3 Аналіз бібліотек для рішення задачі

За останнє десятиліття використання машинного навчання в медицині стрімко розвивалося. У випадку інсульту комерційно доступні алгоритми машинного навчання вже впроваджені в клінічні застосування для швидкої діагностики. Створення та вдосконалення методів глибокого навчання суттєво поліпшили клінічне використання інструментів машинного навчання, і нові алгоритми продовжують з'являтися, підвищуючи точність діагностики і прогнозування інсульту. Визначення ознак і сегментація на основі зображень значно полегшили швидку діагностику та сортування інсульту, прогноз

інсульту залежить від численних пацієнт-специфічних та клінічних факторів, отже, точне прогнозування результату залишається складною задачею.

У сучасному стані технологія машинного навчання служить корисним і ефективним інструментом для швидкого прийняття клінічних рішень, тоді як контроль з боку клінічних експертів все ще потрібний для вирішення конкретних аспектів, які не враховані в автоматизованому алгоритмі.

Машинне навчання — це галузь штучного інтелекту, що вивчає алгоритми та моделі, які дають системам здатність навчатися та покращувати свою продуктивність на основі даних. Воно використовує методи, які дозволяють комп'ютерам автоматично вивчати та вдосконалювати свою роботу, не будучи явно програмованими. [4]

- Застосування машинного навчання;
- Класифікація;
- Регресія;
- Кластеризація;
- Аналіз великих обсягів даних;
- Рекомендаційні системи;
- Обробка природної мови (NLP);
- Медичне дослідження;
- Автономні автомобілі;
- Фінансовий аналіз.

Прогнозування ринкових тенденцій та прийняття інвестиційних рішень.

Машинне навчання може використовуватися для аналізу медичних даних, ідентифікації ризикових факторів, та розробки моделей передбачення. Алгоритми можуть враховувати різні параметри, такі як вік, стать, артеріальний тиск, рівень холестерину та інші медичні показники для прогнозування індивідуального ризику виникнення інсульту та прийняття вчасних заходів для запобігання. [4]

Python має розгалужену екосистему бібліотек і фреймворків, які охоплюють різні сфери розробки. Декілька ключових бібліотек варто відзначити:

– NumPy надає інструменти для роботи з багатовимірними масивами та виконання математичних операцій;

– SciPy розширює можливості NumPy, додаючи функціонал для наукових обчислень, оптимізації та обробки сигналів;

– Pandas використовується для аналізу та обробки даних у формі таблиць. Це потужний інструмент для маніпулювання та очищення даних;

– Django та Flask є фреймворками для веб-розробки, які допомагають швидко створювати високоякісні веб-додатки. Django відомий своєю повнотою та конвенціями, в той час як Flask надає більшу гнучкість і простоту;

– TensorFlow та PyTorch є ключовими бібліотеками для машинного навчання та глибокого навчання. Вони дозволяють створювати, навчати та валідувати складні моделі штучного інтелекту.

Python став домінуючою мовою для розробки моделей машинного навчання та проведення аналізу даних. Масштабні фреймворки, такі як TensorFlow та PyTorch, надають інструменти для розробки та тренування складних моделей. Бібліотеки, такі як scikit-learn та pandas, допомагають у виконанні аналізу та обробці даних [4].

Python займає центральне місце в світі машинного навчання та аналізу даних завдяки ряду потужних бібліотек і фреймворків.

Машинне навчання надає широкий спектр методів для розв'язання різноманітних завдань, включаючи Random Forests, Decision Trees, Boosted Trees, Logistic Regression, Naive Bayes та Deep Learning. Кожен з цих методів має свої унікальні особливості та алгоритми роботи, що робить їх ефективними у різних сценаріях.

Хоча для виконання аналізу даних на Python є безліч бібліотек, ось кілька з них, які допоможуть нам почати роботу:

Logistic Regression використовує логістичну функцію для визначення ймовірності того, що екземпляр належить до конкретного класу. Цей метод легко впроваджується та інтерпретується, зокрема використовується для бінарної класифікації та регресії [6].

KNearest Neighbors визначає клас нового екземпляра, порівнюючи його з k найближчими сусідами у просторі ознак. Цей метод ефективний для невеликих обсягів даних та використовується в задачах класифікації та регресії (рис. 1.1) [7].

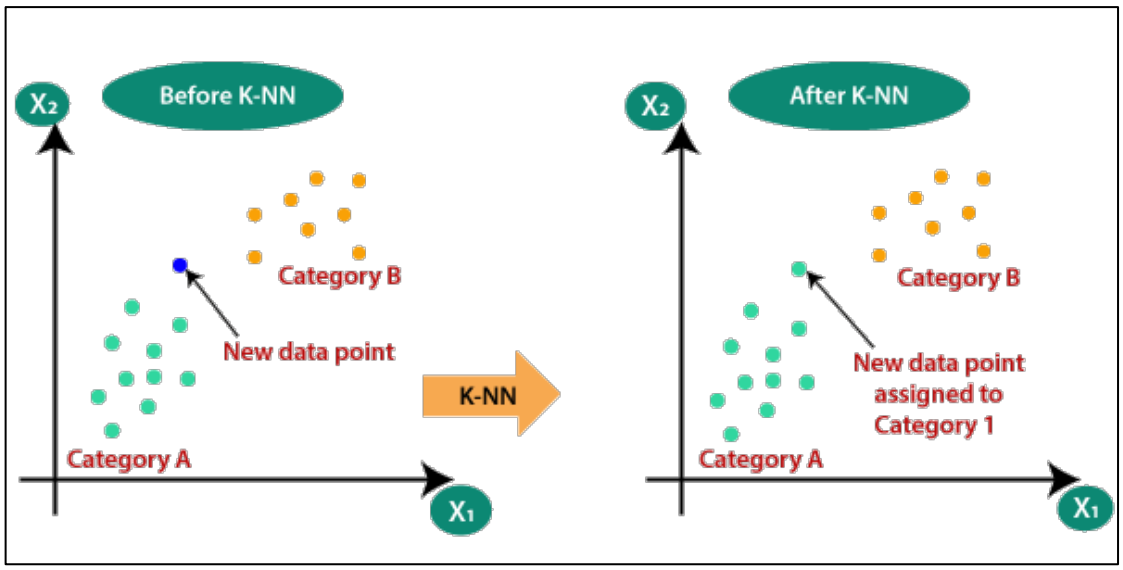


Рисунок 1.1 – Алгоритм роботи KNearest Neighbors

Decision Tree Classifier використовує дерева рішень для прийняття рішень на основі значень ознак. Кожен шлях в дереві представляє конкретний варіант рішення. Його простота та інтерпретованість роблять його відмінним для пояснення логіки прийняття рішень (рис 1.2) [8].

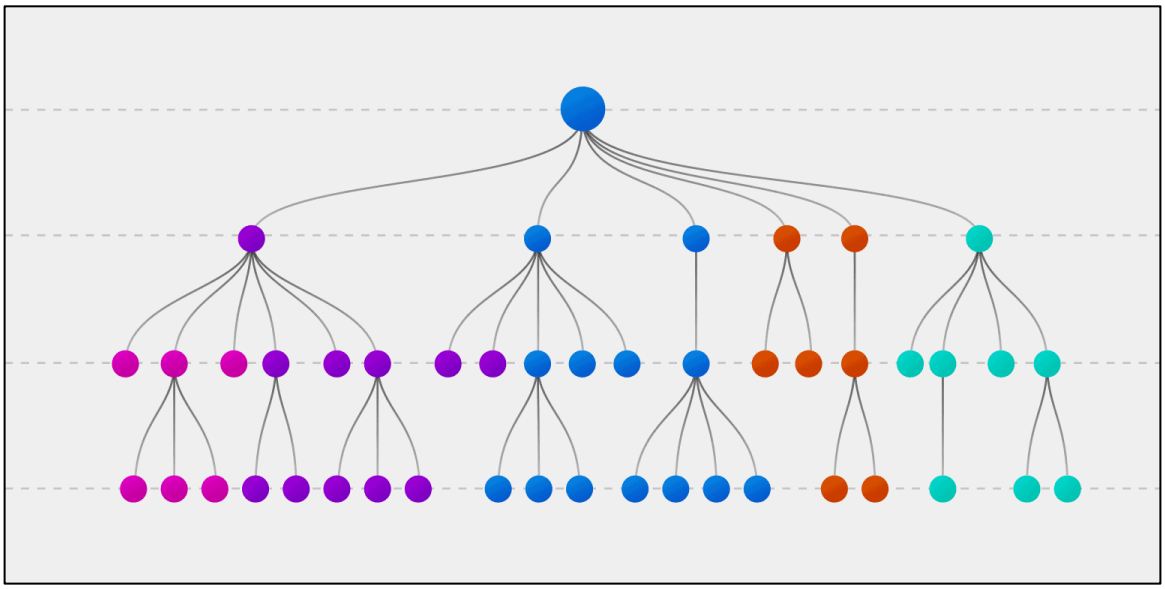


Рисунок 1.2 – Візуалізація роботи Decision trees

Random Forest Classifier — це ансамбль дерев рішень, який працює разом для забезпечення вищої точності та уникнення перевантаження. Кожне дерево навчається на випадковому підмножині даних, і результати об'єднуються для класифікації чи регресії (рис. 1.3) [9].

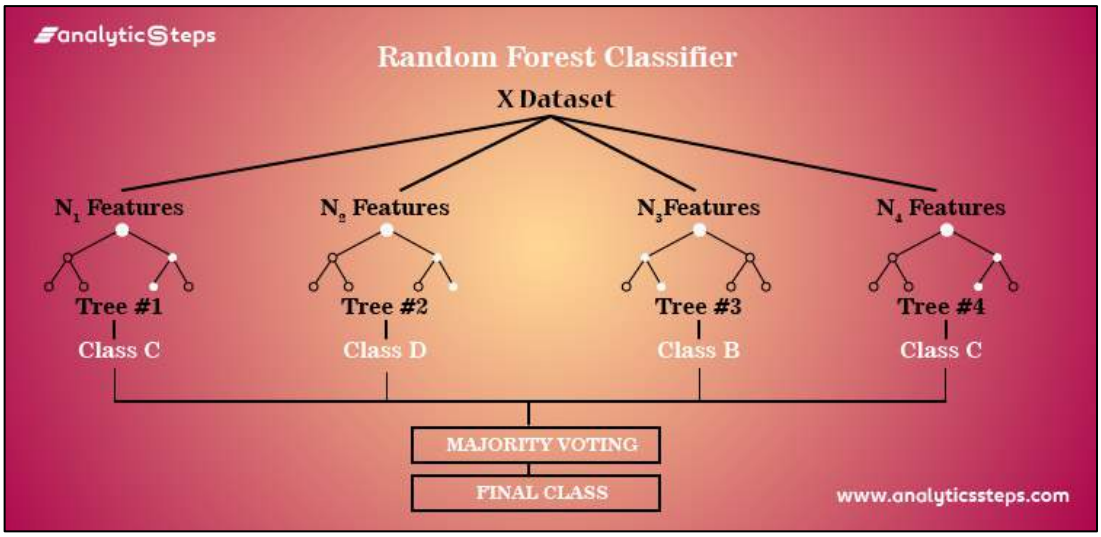


Рисунок 1.3 – Алгоритм роботи Random Forest Classifier

Ada Boost використовує ансамбль слабких класифікаторів та вдосконалює їхні результати, надаючи більший вагомий коефіцієнт правильно

класифікованим екземплярам. Це допомагає знизити помилки та покращити загальну ефективність (рис. 1.4) [10].

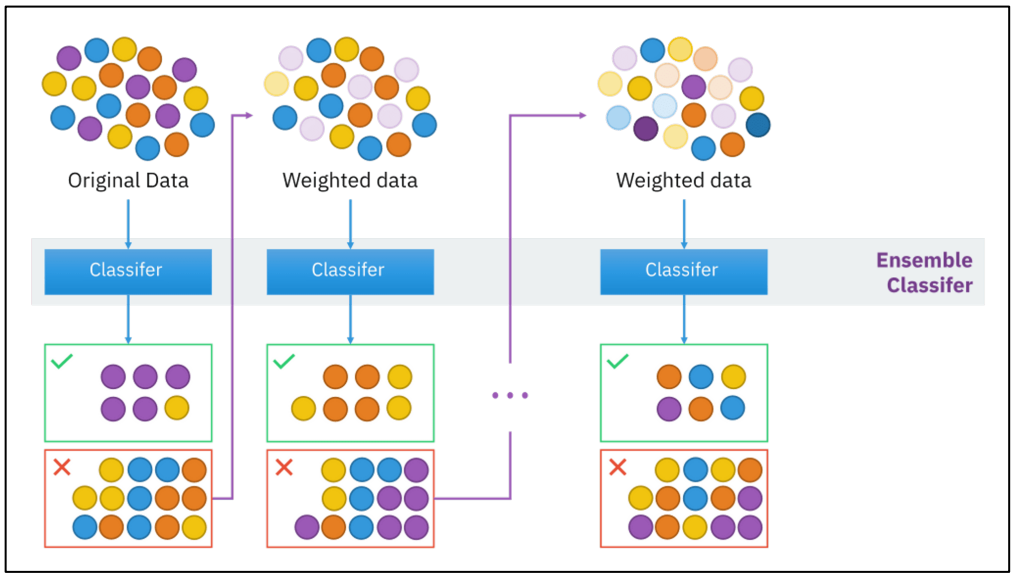


Рисунок 1.4 – Візуалізація роботи AdaBoost

Support Vector Machine (SVM) шукає оптимальний гіперплощину, яка розділяє дані різних класів у просторі ознак. Його ефективність збільшується за допомогою ядерних функцій, що дозволяють робити нелінійні розділення [11].

XGBoost — це оптимізована бібліотека для градієнтного бустінгу, яка ефективно вирішує проблеми класифікації, регресії та ранжування. Вона базується на ансамблі дерев рішень та відома своєю швидкодією та високою точністю (рис. 1.5) [13].

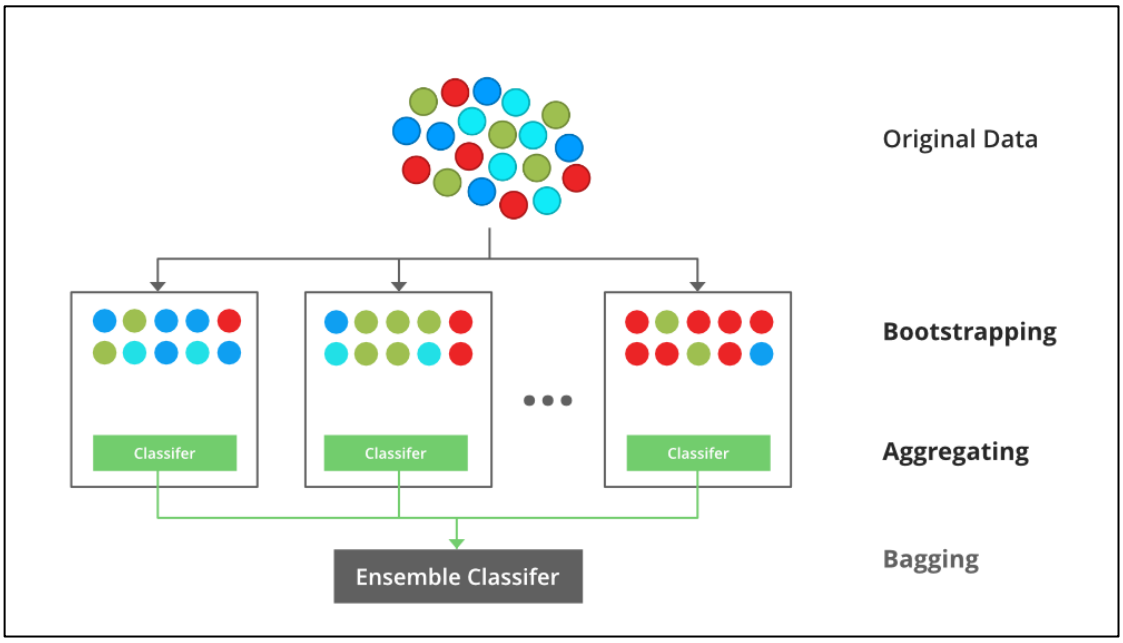


Рисунок 1.5 – Візуалізація роботи XGBoost

Ці методи представляють різні підходи до машинного навчання та дозволяють вирішувати різноманітні завдання в залежності від специфіки даних та потреб проекту.

Gradient Boosting Classifier - це алгоритм машинного навчання, який використовується для задач класифікації та регресії. Він побудований на принципі градієнтного підсилення, де кожна нова модель намагається виправити помилки попередньої моделі (рис. 1.6) [14].

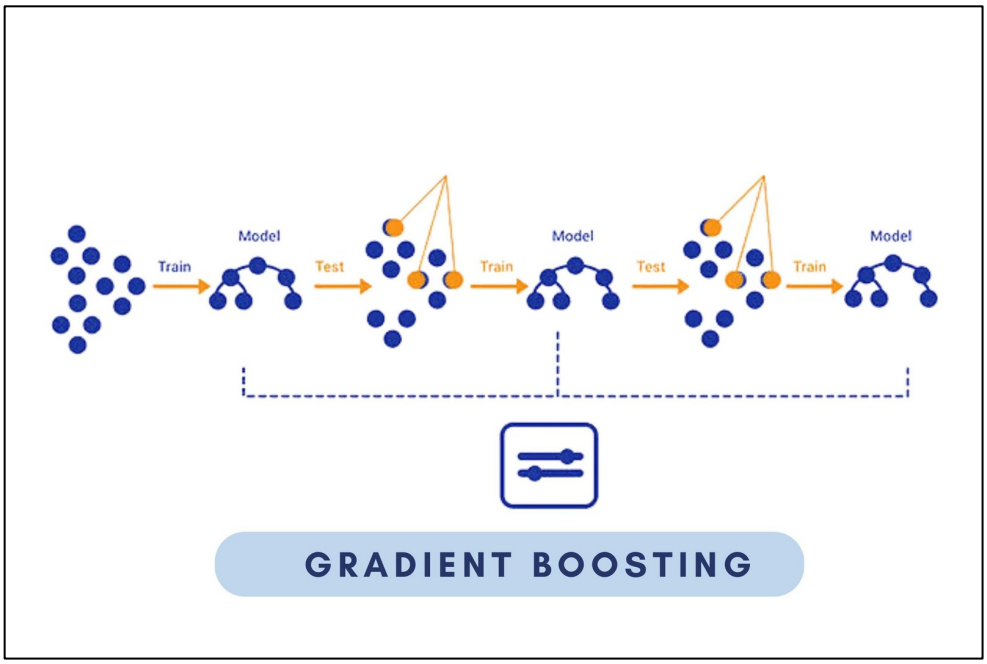


Рисунок 1.6 – Візуалізація роботи Gradient Boosting

Цей алгоритм будує адитивну модель послідовно, дозволяючи оптимізувати довільні диференційовані функції втрат. Він включає в себе декілька слабких моделей, які об’єднуються в сильну модель. Кожна нова модель навчається мінімізувати функцію втрат, таку як середньоквадратична помилка або перехресна ентропія, попередньої моделі за допомогою градієнтного спуску.

1.4 Аналіз аналогів з прогнозування стану хворих на інсульт

Прогнозування стану хворих на інсульт є критичним аспектом сучасної медицини та вимагає вдосконалених методів, які забезпечують точність та надійність результатів. У цьому розділі ми ретельно проаналізуємо ряд аналогів, що вже застосовуються у галузі прогнозування інсультів, з метою визначення найбільш ефективних та перспективних підходів.

На платформі Kaggle виявлено значну кількість готових рішень, спрямованих на прогнозування стану хворих на інсульт. Це свідчить про

активний інтерес спільноти до даної проблематики та важливість розв'язання цього завдання в сучасній медицині та науці про дані.

Розгянемо рішення під назвою «stroke-prediction by using RF with accuracy 93%» (рис. 1.7)

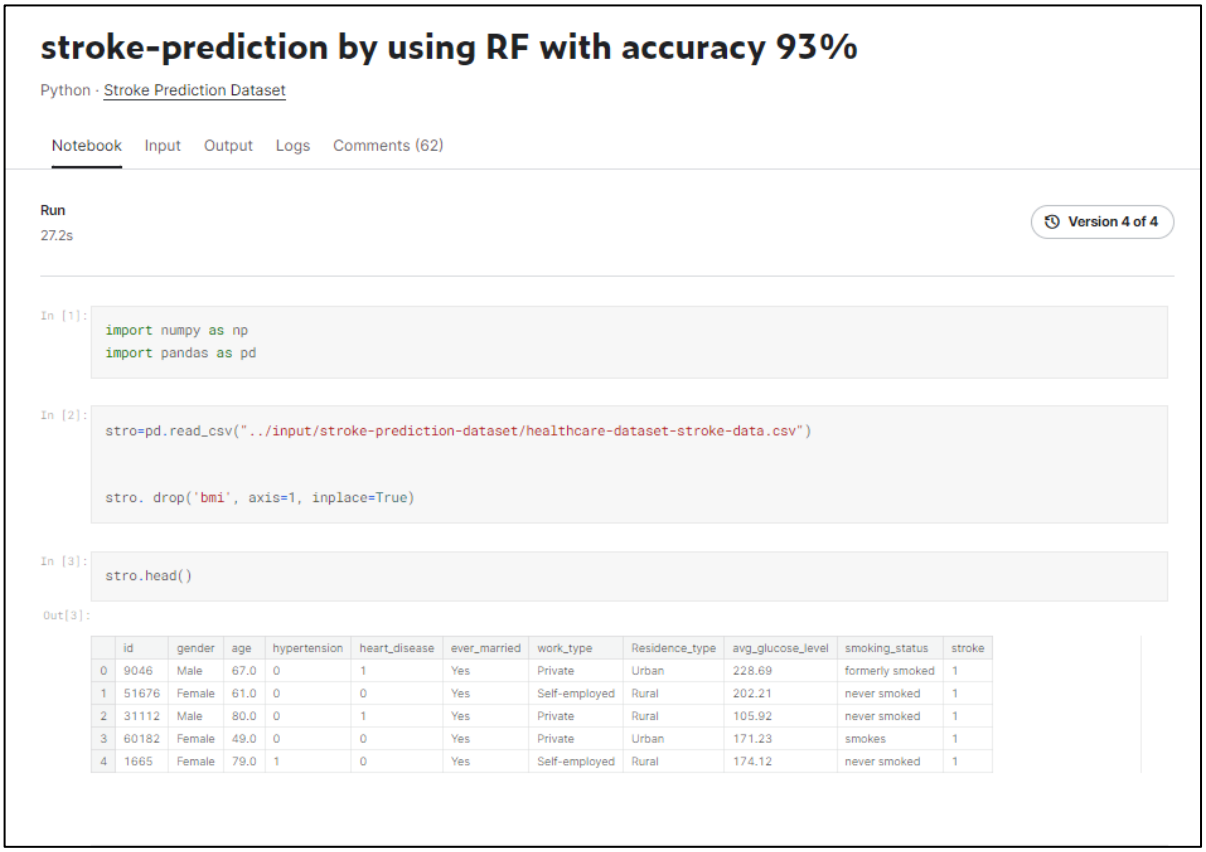


Рисунок 1.7 – Сторінка готового рішення «stroke-prediction by using RF with accuracy 93%»

З огляду виконаної роботи ми можемо побачити, що було використано лише одну модель Random Forest Classifier, яка відображає задовільні результати (рис 1.8).

RandomForestClassifierModel Test Score is : 0.9333333333333333

```
RandomForestClassifierModel = RandomForestClassifier(criterion = 'gini',n_estimators=100,max_depth=5,random_state=33) #criterion can
be also : entropy
RandomForestClassifierModel.fit(x_train, y_train)

#Calculating Details
print('RandomForestClassifierModel Train Score is : ' , RandomForestClassifierModel.score(x_train, y_train))
print('RandomForestClassifierModel Test Score is : ' , RandomForestClassifierModel.score(x_test, y_test))
print('RandomForestClassifierModel features importances are : ' , RandomForestClassifierModel.feature_importances_)
print('----------------------------------------------------')

#Calculating Prediction
y_pred = RandomForestClassifierModel.predict(x_test)
y_pred_prob = RandomForestClassifierModel.predict_proba(x_test)
print('Predicted Value for RandomForestClassifierModel is : ' , y_pred[:10])
print('Prediction Probabilities Value for RandomForestClassifierModel is : ' , y_pred_prob[:10])
```

```
RandomForestClassifierModel Train Score is :  0.9515748031496063
RandomForestClassifierModel Test Score is :  0.9333333333333333
```

Рисунок 1.8 – Код та результат побудови моделей машинного навчання.

Хоча і результат є успішним, але є простір для покращення результатів в точності. Також можна добавити ще декілька моделей для порівняння результатів. Окремі моделі можуть бути більш точними для певних наборів даних або для певних видів прогнозів. Використовуючи декілька моделей, можна отримати більш точний прогноз, ніж за допомогою однієї моделі.

Важливо зазначити, що відсутній будь який опис роботи, що ускладнює аналіз та розуміння. При наявності інформації можна було б оцінити рішення та зробити більш обґрунтовані висновки.

Для отримання більш точних і надійних результатів дослідження можна використовувати кілька різних моделей машинного навчання та порівняти їхні прогнози.

Одна із найкраще реалізованих технологій є дослідження під назвою «Brain Stroke Analysis Accuracy 96.03%» (рис 1.9). Проект відзначається інформативністю, гарними результатами. Було використано кілька моделей для прогнозування.

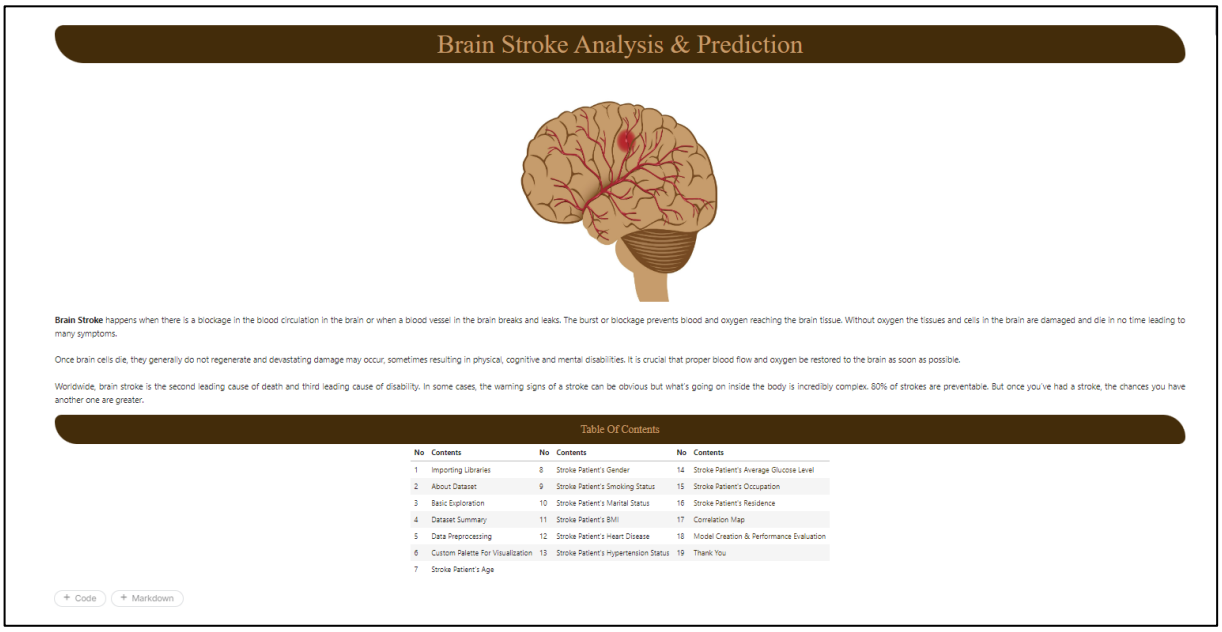


Рисунок 1.9 – Сторінка готового рішення «Brain Stroke Analysis Accuracy 96.03%»

Результат моделі Logistic Regression на рівні 0.97 свідчить про високу точність прогнозування. Важливо зазначити, що в даному вирішенні відсутня діаграма важливості ознак. Недолік цієї аналітики може призвести до того, що ключові аспекти, що впливають на прогнозування, не будуть розглянуті належним чином [14].

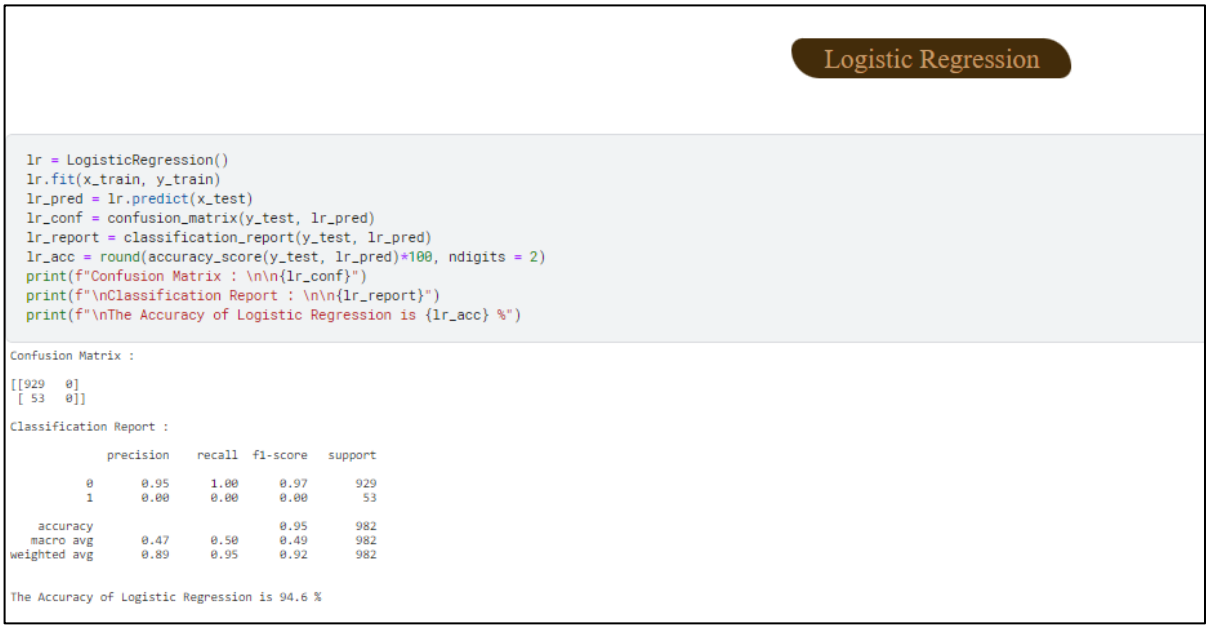


Рисунок 1.10 – Код та результат побудови моделей машинного навчання.

На рисунку 1.11 зображено готове рішення під назвою «Strokes Dataset - EDA & Prediction - 84% Recall» .

В ньому використовувалась тільки одна модель Linear SVC з найкращим результатом 84%, використовуючи тільки вік людей. Але при використання більше атрибутів точність результатів тільки погіршується [14].

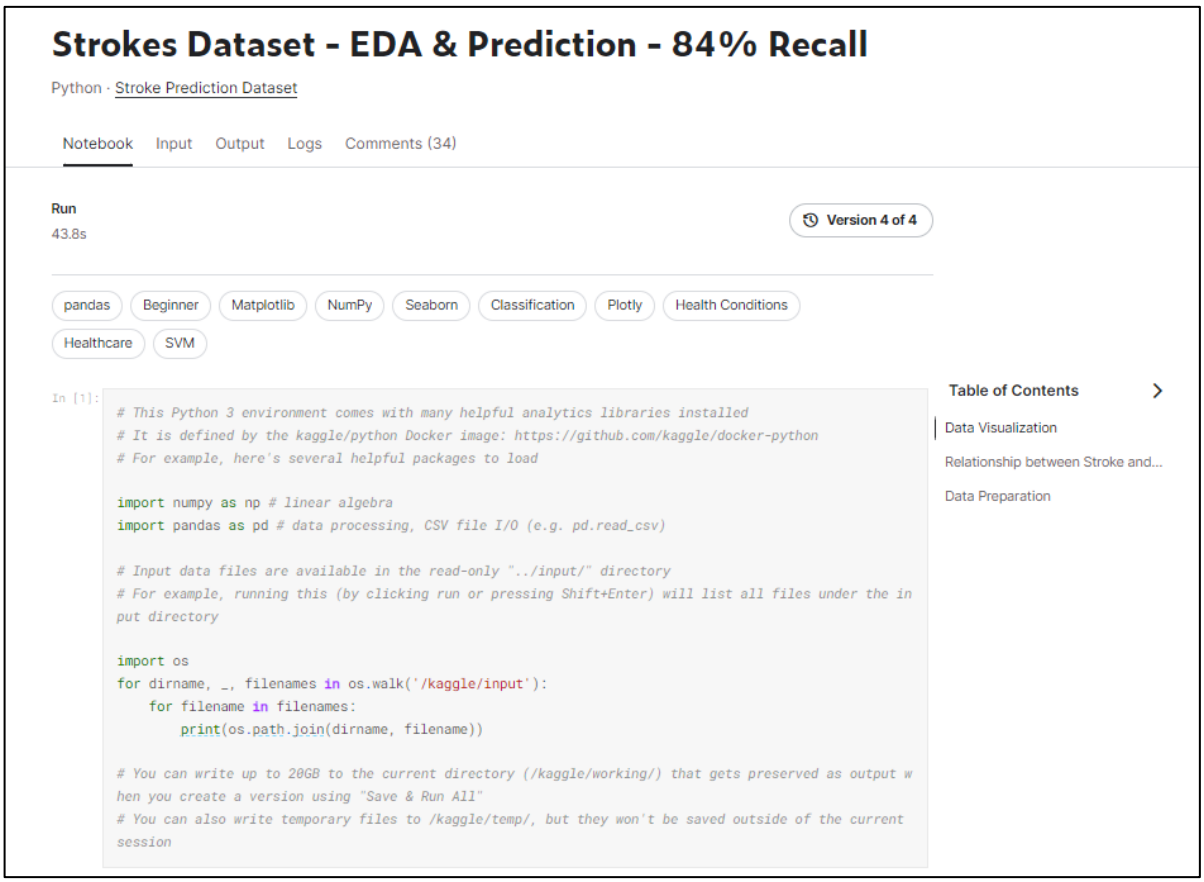


Рисунок 1.11 – Сторінка готового рішення Strokes Dataset - EDA & Prediction - 84% Recall

Причиною низької продуктивності моделі при включенні всіх ознак може бути той факт, що важливі фактори ризику, такі як гіпертонія і серцеві захворювання, виникають з початком старіння (рис. 1.12).

Рисунок 1.12 – Код та результат побудови моделей машинного навчання

Єдиним способом покращити продуктивність моделі є включення більш корисних ознак - не будь-яких, а тих, які не мають жодної кореляції з віком. Для отримання більш точних і надійних результатів дослідження можна використовувати кілька різних моделей машинного навчання та порівняти їхні прогнози [15].

На рисунку 1.13 зображено готове рішення під назвою «Decision Tree Classifier | F1-score 91%».

З огляду виконаної роботи ми можемо побачити, що було використано лише одну модель Decision Tree Classifier, яка відображає задовільні результати 0.9109751784206975

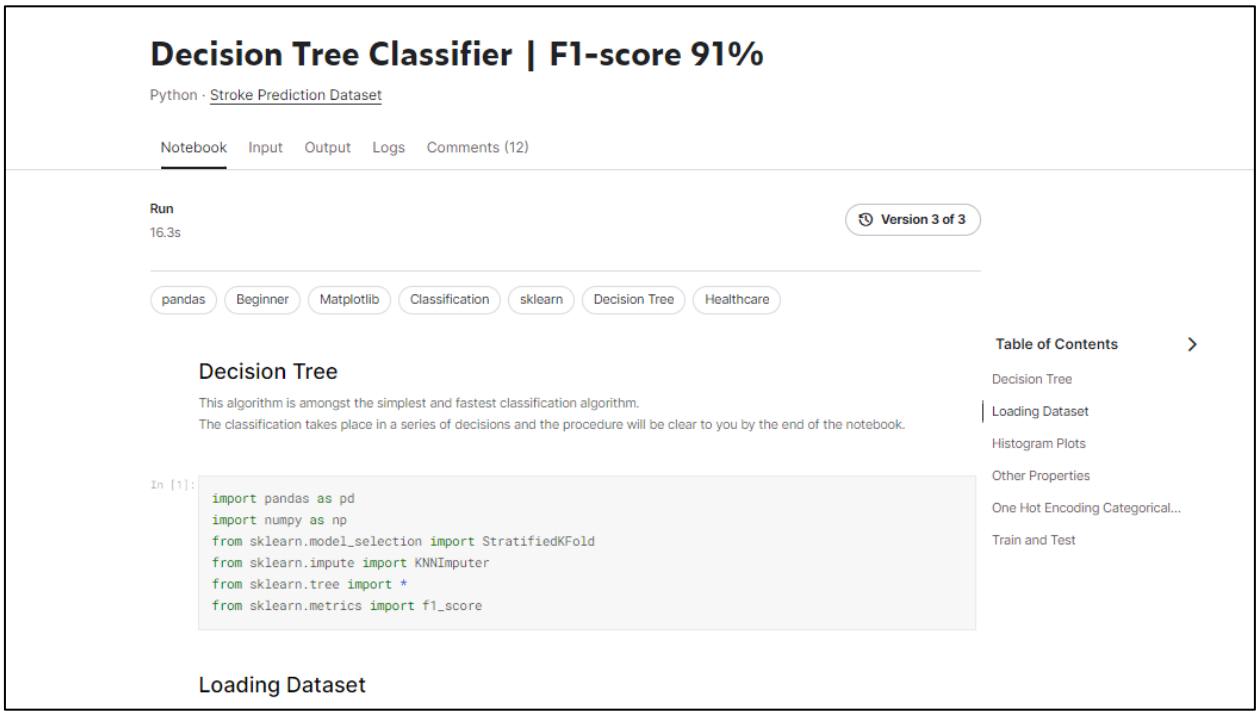


Рисунок 1.13 – Сторінка готового рішення «Decision Tree Classifier | F1-score 91%»

З огляду роботи ми можемо побачити, що було використано лише одну модель Decision Tree Classifier. При наявності додаткових даних можна було б більш точно оцінити рішення, та зробити більш ґрунтовні висновки (рис. 1.14) [16].

```
In [13]:
X = data.drop('stroke',axis=1).values
y = data['stroke'].values

In [14]:
skf = StratifiedKFold(n_splits=5)
skf.get_n_splits(X, y)

for train_index, test_index in skf.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    imputer = KNNImputer(n_neighbors=2)
    X_train = imputer.fit_transform(X_train)
    X_test = imputer.fit_transform(X_test)

    clf = DecisionTreeClassifier()
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    f = f1_score(y_true = y_test , y_pred = y_pred,average = 'weighted')

    print(f)

0.9079179018398901
0.9175035151995882
0.9083771460305464
0.9109751784206975
0.9180007397873157
```

Рисунок 1.14 – Код та результат побудови моделей машинного навчання.

З огляду роботи можна побачити, що він мало інформативний, що ускладню аналіз роботи та розуміння. Також можна додати ще декілька моделей для моделювання та порівняння результатів. Порівняння моделей дозволяє оцінити їхню продуктивність у різних контекстах і виявити найкращі рішення для конкретних проблем [16].

Розглянемо рішення під назвою «EDA + Classification 81% AUC 75.6% Accuracy» (рис. 1.15)

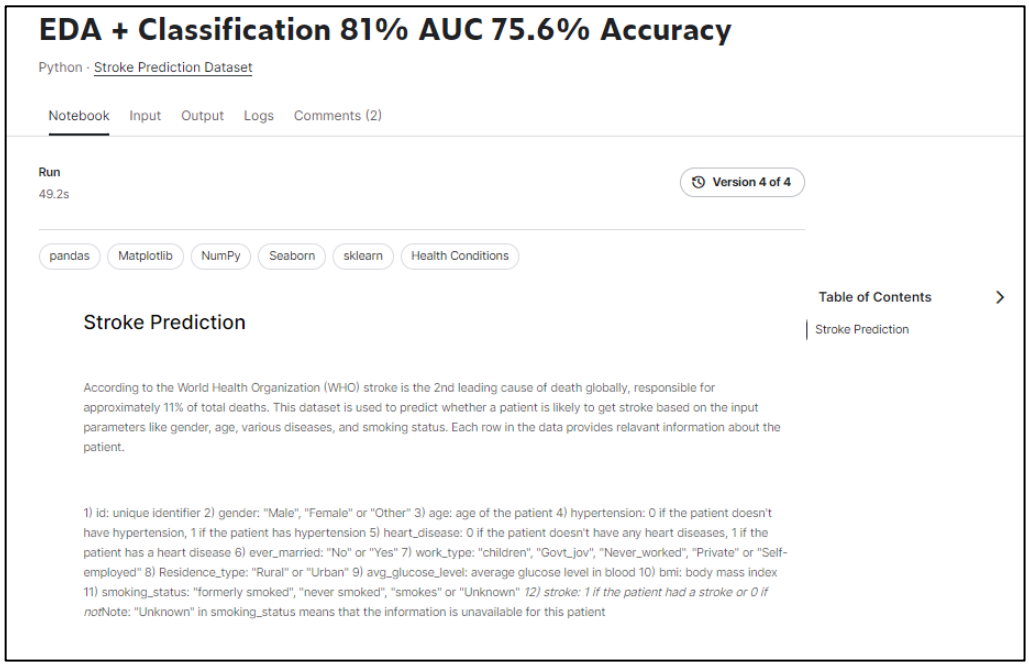
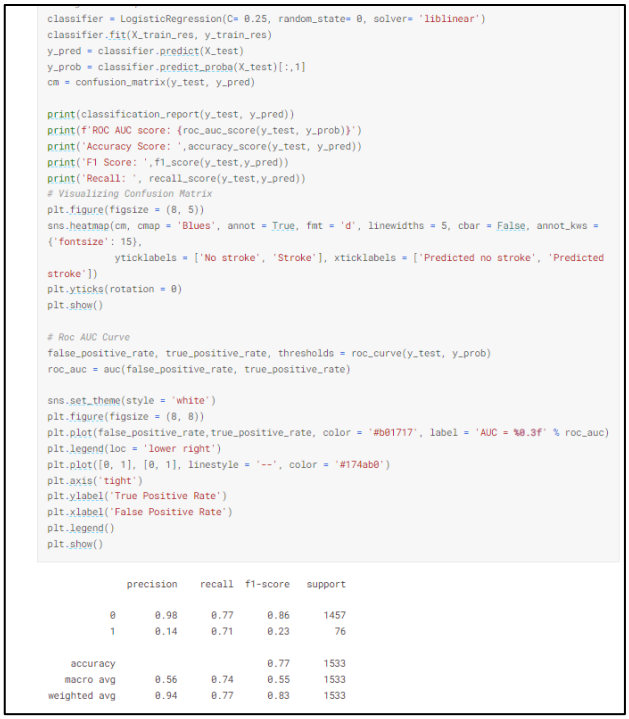


Рисунок 1.15 – Сторінка готового рішення EDA + Classification 81% AUC 75.6% Accuracy

Розглядаючи готове рішення можна побачити, що воно мало інформативне, використовується мало моделей, а найкращий результат всього 75.6% (рис.1.16).



```
classifier = LogisticRegression(C= 0.25, random_state= 0, solver= 'liblinear')
classifier.fit(X_train_res, y_train_res)
y_pred = classifier.predict(X_test)
y_prob = classifier.predict_proba(X_test)[:,1]
cm = confusion_matrix(y_test, y_pred)

print(classification_report(y_test, y_pred))
print(f'ROC AUC score: {roc_auc_score(y_test, y_prob)}')
print('Accuracy Score: ',accuracy_score(y_test, y_pred))
print('F1 Score: ',f1_score(y_test,y_pred))
print('Recall: ', recall_score(y_test,y_pred))
# Visualizing Confusion Matrix
plt.figure(figsize = (8, 5))
sns.heatmap(cm, cmap = 'Blues', annot = True, fmt = 'd', linewidths = 5, cbar = False, annot_kws =
{'fontsize': 15},
            yticklabels = ['No stroke', 'Stroke'], xticklabels = ['Predicted no stroke', 'Predicted
stroke'])
plt.yticks(rotation = 0)
plt.show()

# Roc AUC Curve
false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(false_positive_rate, true_positive_rate)

sns.set_theme(style = 'white')
plt.figure(figsize = (8, 8))
plt.plot(false_positive_rate,true_positive_rate, color = '#b01717', label = 'AUC = %0.3f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1], linestyle = '--', color = '#174ab0')
plt.axis('tight')
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend()
plt.show()
```

	precision	recall	f1-score	support
0	0.98	0.77	0.86	1457
1	0.14	0.71	0.23	76
accuracy			0.77	1533
macro avg	0.56	0.74	0.55	1533
weighted avg	0.94	0.77	0.83	1533

Рисунок 1.16 – Код та результат побудови моделей машинного навчання

Суттєвим недоліком є використання всього лише дві моделі, що приводить до того , що це не лише обмежує, а й зводить на невеликий рівень розвідувальний аналіз [16].

Останнім розглянутим готовим рішенням буде готове під назвою «Decision Tree | Stroke Prediction | Test – 92.15%» (рис. 1.17).

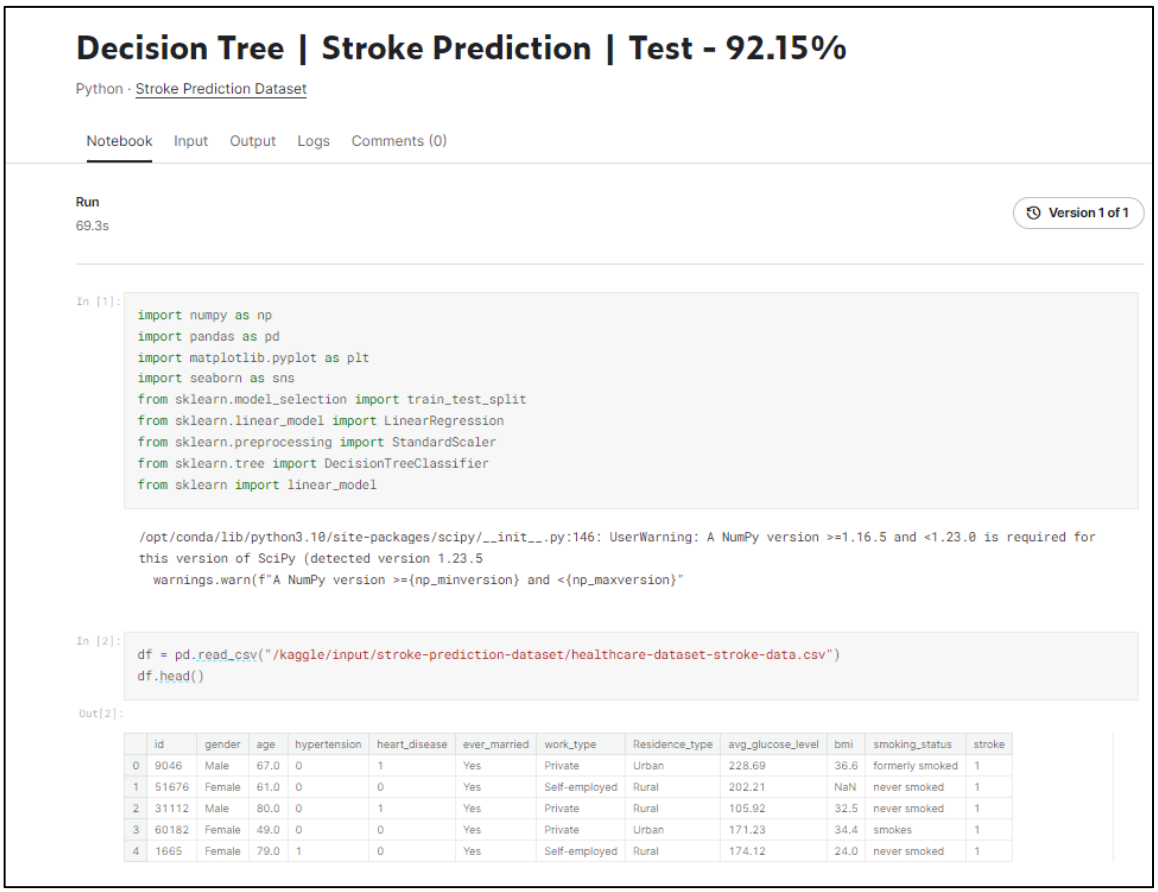


Рисунок 1.17 – Сторінка готового рішення «Decision Tree | Stroke Prediction | Test – 92.15%»

В готовому рішені використовується модель машинного навчання DecisionTreeClassifier, яка надає точність на рівні 0.92, було виконано очищення данних, що призвело до покращення результатів, ніж у попередників (рис. 1.18).

Рисунок 1.18 – Код та результат побудови моделей машинного навчання

Ретельний аналіз цієї інформаційної технології підтвердив, що в ній, як і в попередньому дослідженні, використовується лише одна модель машинного навчання – Decision Tree Classifier, якщо завдання вимагає більшої складності або у випадках, коли Decision Tree Classifier не є оптимальним рішенням. Врахування різноманітності моделей може поліпшити загальну роботу системи та забезпечити кращі результати для різноманітних типів даних та завдань.

1.5 Висновки

В цьому розділі досліджено предмету область хворих на інсульт, проаналізовано ключові характеристики та переваги різних підходів до передбачення хворих на інсульт методами машинного навчання, що включають у себе методи класифікації та кластеризації даних. Після аналізу існуючих рішень було виявлено, що вони часто не зосереджуються на попередній обробці даних, використанні декількох прогностичних моделей та виявленні аномалій. Ці аспекти будуть взяті до уваги при проектуванні нашої інформаційної технології.

2 РОЗВІДУВАЛЬНИЙ АНАЛІЗ ДАНИХ

2.1 Вибір середовища розробки

Для виконання поставленого завдання було обрано платформу Kaggle, це рішення було обрано через визнану ефективність у вирішені схожих завдань завдяки великій спільноті фахівців зі всього світу, які можуть надати підтримку та допомогу у вирішенні завдань, має доступ до потужних обчислювальних ресурсів. Це дозволяє швидко і ефективно обробляти великі набори даних, має інтегровані інструменти для обробки даних. Ці інструменти полегшують роботу з даними і дозволяють швидко отримати потрібні результати. Kaggle надає можливість участі в конкурсах [17].

Ці конкурси є відмінним способом обміну ідеями та розвитку навичок у сфері аналізу даних. Отже, Kaggle є стабільною і надійною платформою. Це гарантує, що проект буде виконаний успішно.

Kaggle - це відома онлайн-платформа для змагань у сфері аналізу даних, машинного навчання та штучного інтелекту. Заснована в 2010 році, Kaggle стала центром спільноти для досліджувачів, науковців та професіоналів, які зацікавлені у вирішенні реальних проблем та завдань, пов'язаних з аналітикою даних (рис. 2.1) [17].

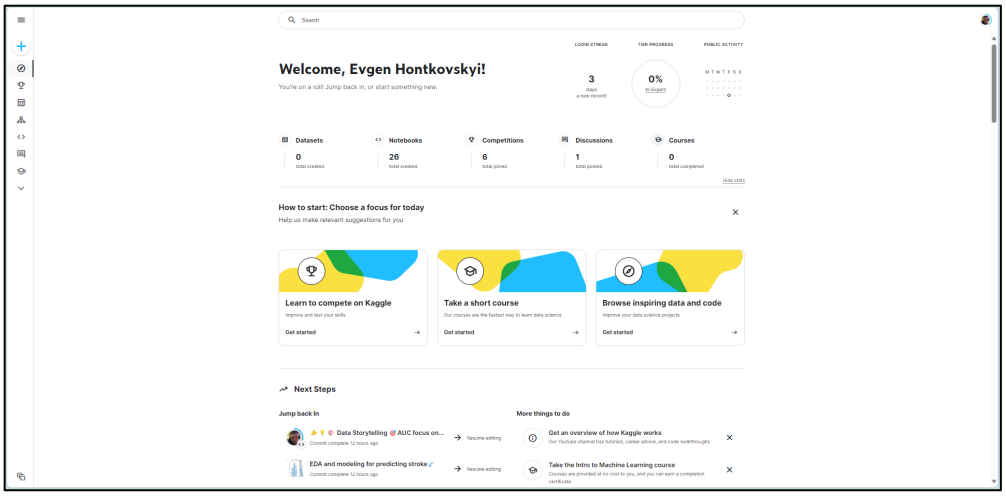


Рисунок 2.1 – Головна сторінка Kaggle

Основні особливості Kaggle:

Датасети:

– Платформа містить розмаїття даних, які можуть бути використані для аналізу та моделювання. Це дозволяє учасникам експериментувати та створювати моделі на різних типах даних (рис. 2.2);

– Користувачі можуть завантажувати власні датасети для спільної роботи або використання в конкурсах. Це сприяє обміну знаннями та взаємодії спільноти [17]

– Kaggle надає ресурси для навчання, які допомагають учасникам покращити свої навички в аналізі даних та машинному навчанні;

– Учасники можуть отримати визнання за досягнення та навички, які вони отримують через участь у змаганнях.

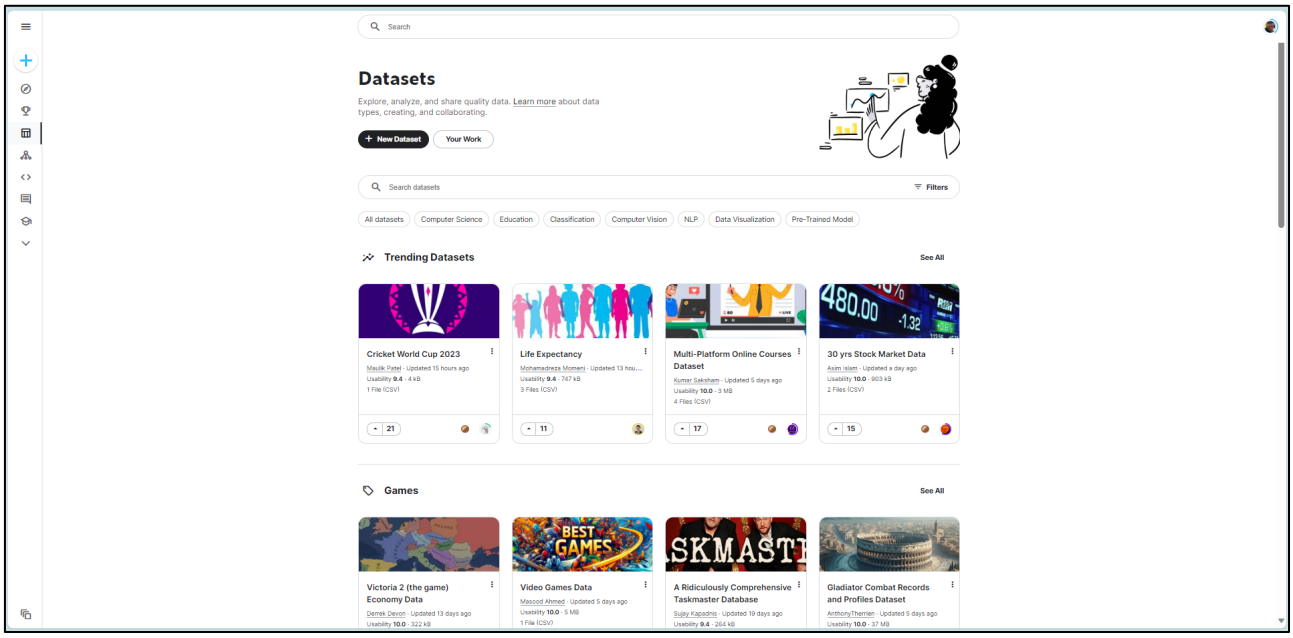


Рисунок 2.2 – Сторінка з датасетами у Kaggle

Змагання та завдання:

– Kaggle пропонує змагання та завдання в різних галузях, таких як медицина, фінанси, комп'ютерне бачення, обробка природних мов, та інші. Це

дозволяє учасникам спробувати свої сили в різних областях та розвивати широкий спектр навичок;

– Завдання на Kaggle часто базуються на реальних проблемах, з якими стикаються компанії та організації. Вирішення цих завдань може мати практичний вплив на суспільство та бізнес;

Код та ноутбуки:

– Учасники можуть ділитися своїм кодом, ноутбуками та результатами своїх досліджень. Це стимулює відкритий обмін ідеями та кращими практиками [17];

– Користувачі можуть відзначати корисні внески та визначати їхній внесок у розв'язанні конкретних завдань. Це важливо для визнання та підтримки спільноти;

Форум та обговорення:

– Kaggle має активну спільноту, де учасники можуть обговорювати свої ідеї, задавати питання та ділитися досвідом;

– Учасники можуть звертатися за допомогою та порадами до інших учасників або експертів у відповідних галузях (рис. 2.3);

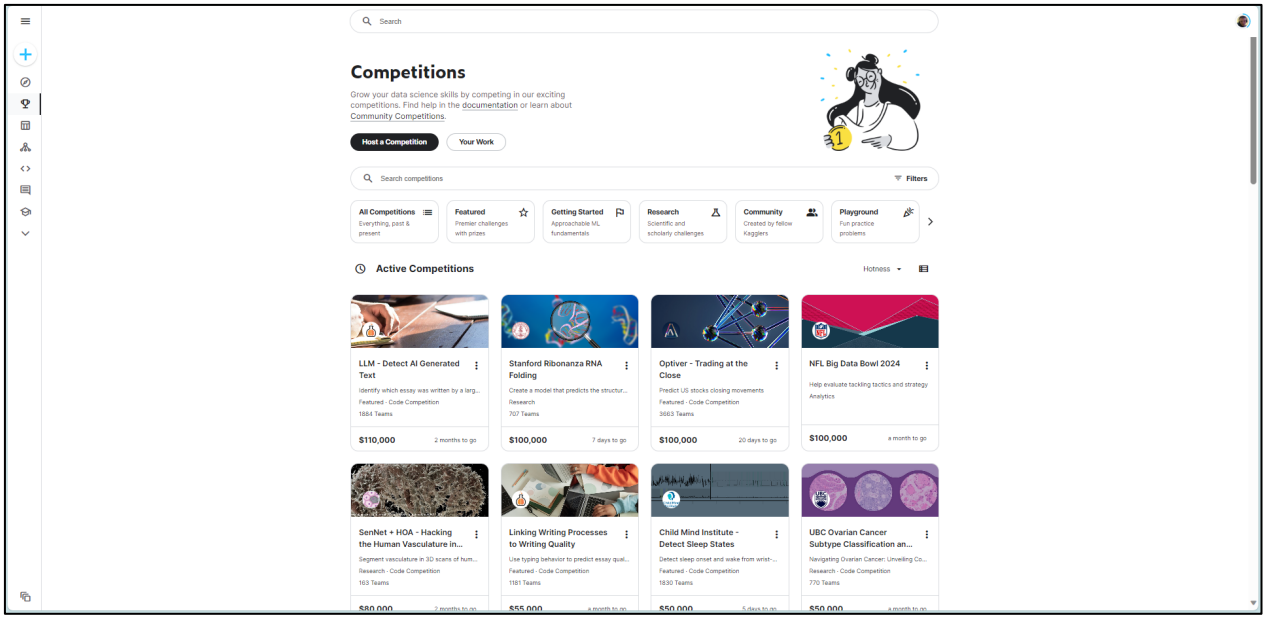


Рисунок 2.3 – Сторінка конкурсів у Kaggle

2.2 Обробка початкових даних

Перед тим, як розпочати виконання завдання, слід провести необхідну підготовку даних та імпортувати відповідні бібліотеки. У початковому етапі було здійснено імпорт усіх необхідних бібліотек для належного виконання завдання (рис. 2.4).

```
import warnings
warnings.filterwarnings('ignore')

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import sklearn

import matplotlib.pyplot as plt
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import plotly.figure_factory as ff


from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV

from sklearn.metrics import accuracy_score, classification_report, roc_curve,precision_recall_curve, auc,confusion_matrix
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.svm import SVC
from sklearn.impute import KNNImputer

from xgboost import XGBClassifier

from catboost import CatBoostClassifier
```

Рисунок 2.4 – Імпорт бібліотек

На рисунку 2.4 зображено з імпортування декількох бібліотек.

NumPy є важливою бібліотекою для мови програмування Python, призначеною для числових обчислень та обробки масивів даних. Ця бібліотека надає високоефективні масиви, що дозволяють легко вирішувати завдання з обчислення, що вимагають обробки великих обсягів даних [20].

В основі NumPy знаходиться об'єкт ndarray, що представляє собою багатовимірний масив. Це спрощує операції над даними, оскільки дозволяє використовувати векторизовані операції, тобто виконувати операції на цілих масивах даних одночасно.

Використання NumPy полегшує роботу з математичними функціями, що оперують на масивах, включаючи тригонометричні функції, логарифми та інші. Це важливо у великих обчисленнях та наукових дослідженнях [20].

Окрім того, NumPy надає зручний механізм для індексації та роботи зі зрізами масивів, що спрощує вибір та редагування певних елементів даних.

Бібліотека також має вбудовані функції для виконання операцій лінійної алгебри, що робить її невід'ємною для вирішення задач, пов'язаних із матрицями та векторами.

Узагальнюючи, NumPy забезпечує зручні та потужні інструменти для роботи з числовими даними в мові програмування Python, що робить його ключовою складовою для наукових досліджень та інженерних завдань [20].

Pandas є потужною бібліотекою для мови програмування Python, розробленою для обробки та аналізу даних у табличному форматі. Основними структурами даних в Pandas є Series і DataFrame. Series є одновимірним масивом даних з індексами, тоді як DataFrame представляє собою двовимірну табличну структуру даних, схожу на електронну таблицю чи базу даних [21].

Основні можливості Pandas включають:

– Головна структура для представлення та маніпуляції табличними даними. Вона забезпечує інтуїтивний інтерфейс для роботи з рядками та стовпцями, і дозволяє ефективно вирішувати завдання аналізу даних;

– Одновимірна структура даних, яка є базовим елементом для побудови більш складних структур, таких як DataFrame. Series містить дані та асоційований індекс [21].;

– Pandas підтримує читання та запис даних у різних форматах, таких як CSV, Excel, SQL-запити та інші. Це робить його інтеграційною частиною процесу обробки даних з різних джерел;

– Pandas надає зручні методи для індексації та вибору певних частин даних в структурах DataFrame і Series. Це включає у себе вибір по назві стовпців, рядків чи певних умов;

– Здатність групувати дані за певним критерієм та застосовувати агрегуючі функції (сума, середнє значення, тощо) для отримання загальної інформації.

Pandas став невід'ємним інструментом для роботи з даними в областях науки про дані, фінансів, аналізу великих даних та інших галузях. Його зручний та потужний функціонал робить його першим вибором для великої кількості завдань, пов'язаних з обробкою та аналізом табличних даних [21].

Matplotlib - це бібліотека для Python, спеціалізована на візуалізації даних. Забезпечує інструменти для створення різноманітних графічних представлень, таких як лінійні графіки та діаграми. Зручний інтерфейс дає можливість створювати наочні графіки з легкістю. Matplotlib використовується в багатьох галузях, включаючи аналіз даних, наукові дослідження та веб-розробку. Його гнучкість у налаштуванні вигляду графіків дозволяє точно відобразити дані. Ця бібліотека стала стандартом для візуалізації даних у середовищі Python і є невід'ємною частиною аналізу та представлення інформації [22].

Seaborn - це бібліотека для візуалізації даних у мові програмування Python, яка базується на бібліотеці Matplotlib. Вона надає високорівневий інтерфейс для створення привабливих та інформативних статистичних графіків. Seaborn дозволяє легко втілювати складні візуалізації даних з кількома строками коду, і вона часто використовується в областях аналізу даних та машинного навчання.

Основні риси Seaborn: високорівневий інтерфейс, легкість створення графіків, привабливий дизайн, підтримка статистичних моделей, робота з DataFrame.

Scikit-learn (sklearn) - це важлива бібліотека для мови програмування Python у сфері машинного навчання та аналізу даних. Вона дозволяє вчити моделі для класифікації, регресії, кластеризації та інших задач. За допомогою scikit-learn можна використовувати різноманітні алгоритми, такі як метод опорних векторів, дерева рішень, а також реалізації ансамблевих методів [23].

Бібліотека забезпечує інструменти для обробки та підготовки даних перед їх використанням для навчання моделей. Це включає в себе широкий спектр функцій для масштабування, видалення аномалій, обрізки та інші операції.

Оцінка та налаштування моделей також стають легкими завдяки інтерфейсу scikit-learn. Завдяки його функціональності можна проводити перехресну перевірку, визначати важливість ознак та вибирати найкращі параметри моделей.

Усі ці можливості роблять scikit-learn незамінною бібліотекою для розробників та дослідників у галузі машинного навчання, забезпечуючи широкий спектр інструментів для ефективної роботи з даними та розробки моделей [23].

Plotly - це бібліотека для створення високоякісних інтерактивних графіків та візуалізації даних у мові програмування Python. Вона підтримує різноманітні типи графіків, включаючи лінійні, точкові, стовпчасті, кругові, теплові карти та інші. Plotly також дозволяє створювати інтерактивні графіки, що робить її корисною для веб-сервісів та додатків, де користувачі можуть взаємодіяти з графіками.

Після цього було проведено завантаження даних та ознак. Здійснено виведення загальної інформації про датасет (рис. 2.5) [25].

[2]:
```
df = pd.read_csv('../input/stroke-prediction-dataset/healthcare-dataset-stroke-data.csv')
```

[3]:
```
df.head()
```

[3]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Рисунок 2.5 – Завантаження та огляд датасету

У вказаному наборі даних можна побачити наявність 12 ознак і одна бінарна мета. Проведено аналіз та розшифрування абревіатур, що використовуються у наданому наборі даних:

– id: Ідентифікатор особи;
– gender: Стать особи. Може бути “Male” або “Female”;
– age: Вік особи в роках;
– hypertension: Чи є у особи гіпертонія;
– heart disease: Чи є у особи серцеві захворювання;
– ever married: Чи була особа коли-небудь одружена;
– work type: Тип роботи особи. Може бути “Private”, “Self-employed” та інші;
– Residence type: Тип проживання особи. Може бути “Urban” (міський) або “Rural” (сільський);
– avg glucose level: Середній рівень глюкози в крові особи;
– bmi: Індекс маси тіла особи;
– smoking status: Статус куріння особи. Може бути “formerly smoked” (раніше курили), “never smoked” (ніколи не курили) та інші;
– stroke: Чи був у особи інсульт. 0 означає “Ні”, 1 означає “Так”.

На рисунку 2.6 зображено кругову діаграму, яка відображає кількість випадків інсульту в наборі даних. Діаграма має два сектори:

Синій сектор: Відсоток осіб, які не мали інсульту. Цей сектор позначено як 0 та становить 95.1% від загальної кількості осіб.

Червоний сектор: Відсоток осіб, які мали інсульт. Цей сектор позначено як 1 та становить 4.9%.

Ця діаграма допомагає візуалізувати розподіл випадків інсульту в наборі даних, з розподілу стало зрозуміло, що кожні 5 зі 100 осіб страждають від інсультів

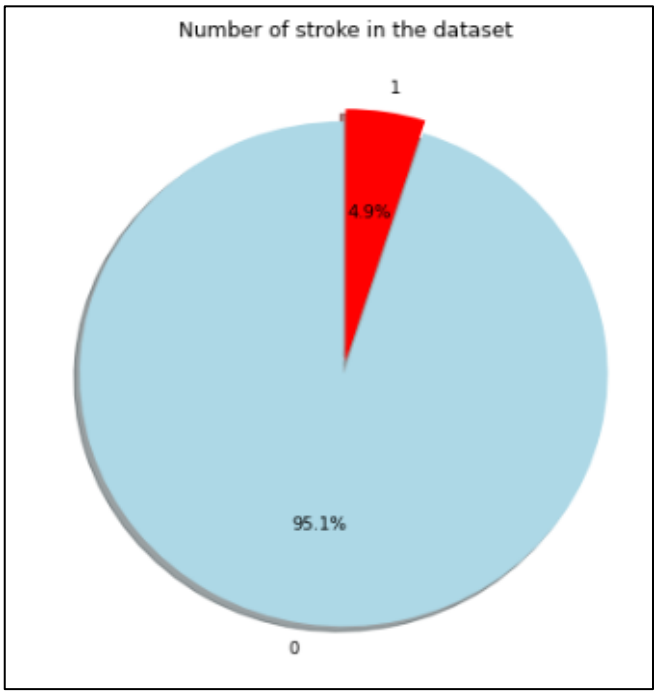


Рисунок 2.6 – Розподіл випадків інсульту в наборі даних

2.3 Розвідувальний аналіз даних

Розвідувальний аналіз даних (Exploratory Data Analysis, EDA) - це важливий етап обробки та аналізу даних. Його основна мета - виявити основні закономірності, тенденції, характер та властивості даних для аналізу, а також закони розподілу величин. Цей аналіз займається попереднім експрес-аналізом даних шляхом їх перетворення та/або представлення у зручному вигляді: графічному, табличному, схем, діаграм і т.д. Він використовується для знаходження зв'язків між змінними в ситуаціях, коли відсутні (або недостатні) апріорні уявлення щодо природи цих зв'язків.

Основні цілі розвідувального аналізу включають максимальне «проникнення» в дані, виявлення основних структур, вибір найвагоміших змінних, виявлення відхилень та аномалій, перевірка основних гіпотез (припущень) та розробка початкових моделей.

До основних методів розвідувального аналізу даних відносять: кластерний аналіз, факторний аналіз, аналіз дискримінантних функцій,

багатомірне шкалювання, логлінійний аналіз, канонічні кореляції, покрокова лінійна та нелінійна регресія, аналіз відповідностей, аналіз часових рядів, дерева класифікації.

Результати розвідувального аналізу не використовуються для вироблення управлінських рішень. Їхнє призначення — допомога в розробці найкращої стратегії поглибленого аналізу, висування гіпотез, уточнення особливостей застосування тих чи інших математичних методів та моделей.

На рисунку 2.7 зображена гістограма яка відображає розподіл індекс маси тіла (BMI). Найвища щільність спостерігається при рівні BMI близько 40.

Це означає, що чим вище BMI тим більший ризик інсульту і доводить, що інсульти найбільш поширені у людей з надмірною вагою.

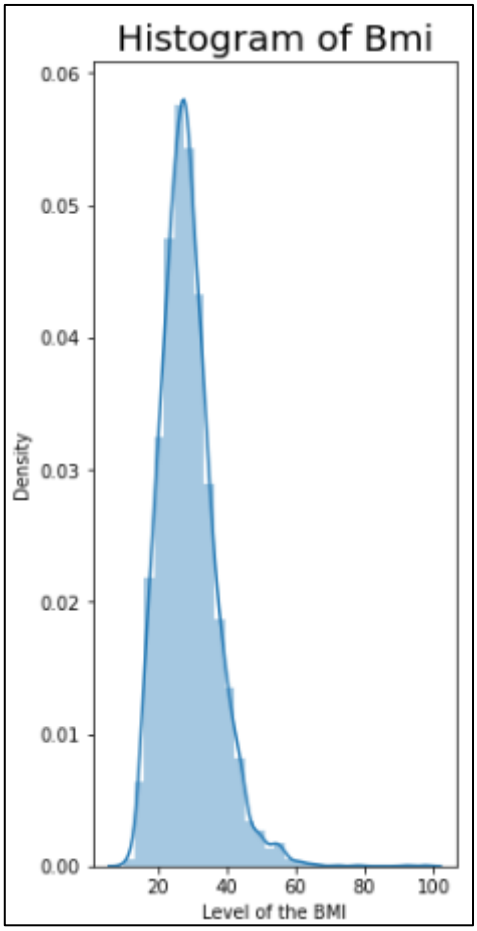


Рисунок 2.7 – Гістограма розподілу за індексом маси тіла

Було побудовано гістограму з розподілом людей за віком, які мають більший ризик перенести інсульт. З нього зрозуміло, що найвища частота спостерігається при віці близько 40 років (рис. 2.8).

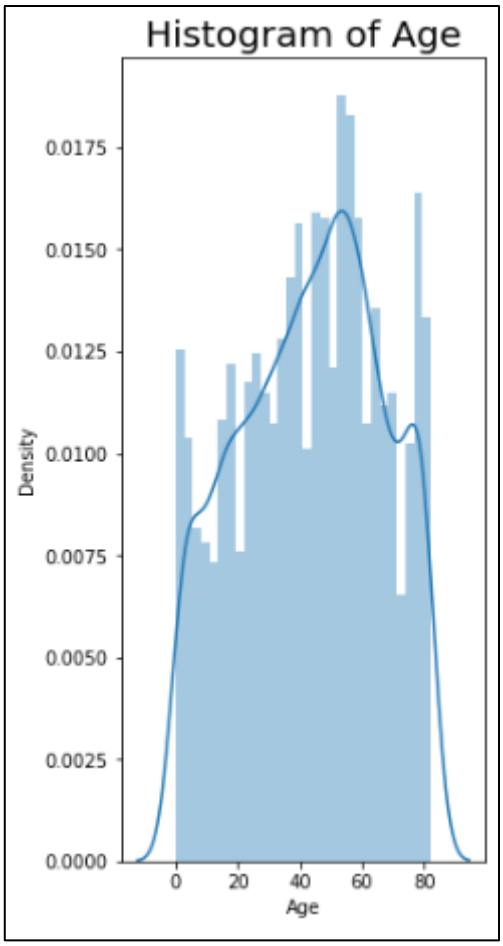


Рисунок 2.8 – Гістограма розподілу за віком

Було побудовано гістограму, яка відображає розподіл рівнів сироваткового креатиніну в крові. Більшість розподілу падає між 50 та 150 мг/дл. (рис. 2.9).

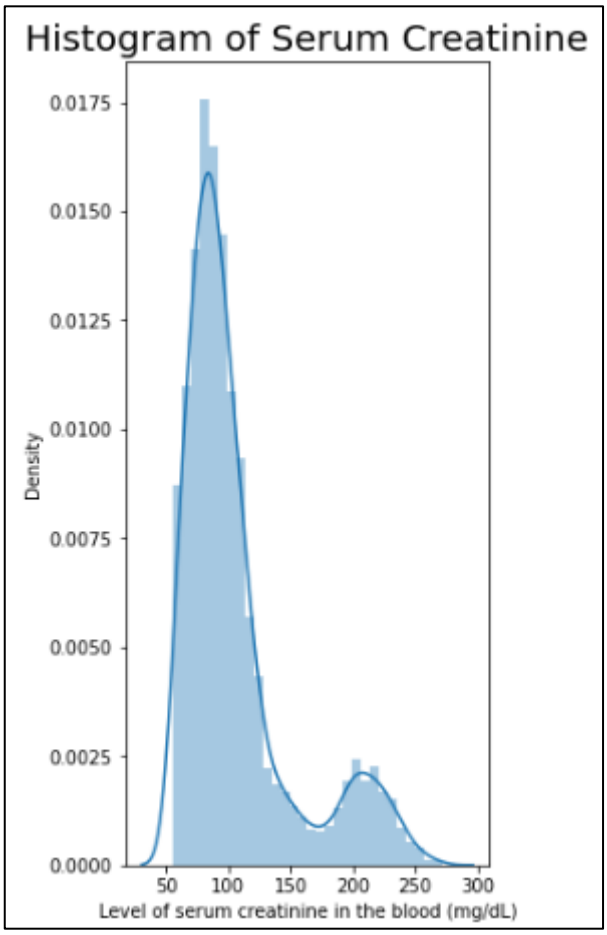


Рисунок 2.9 – Гістограма відображає розподіл рівнів креатиніну в крові

На рисунку 2.10 зображена стовпчаста діаграма, яка відображає кількість людей, які перенесли інсульт, залежно від типу їх роботи. Стовпці розфарбовані залежно від того, чи переносила людина інсульт чи ні. Сині стовпці представляють людей, які не переносили інсульт, а помаранчеві стовпці представляють людей, які перенесли інсульт. З неї ми можемо зробити висновки, що серед приватних підприємців та самозайнятих осіб кількість людей, які перенесли інсульт, є приблизно однаковою. Однак люди, які працюють у державному секторі, мають більше шансів не мати інсульту порівняно з обома першими категоріями, до того ж діти не дуже схильні до інсульту.

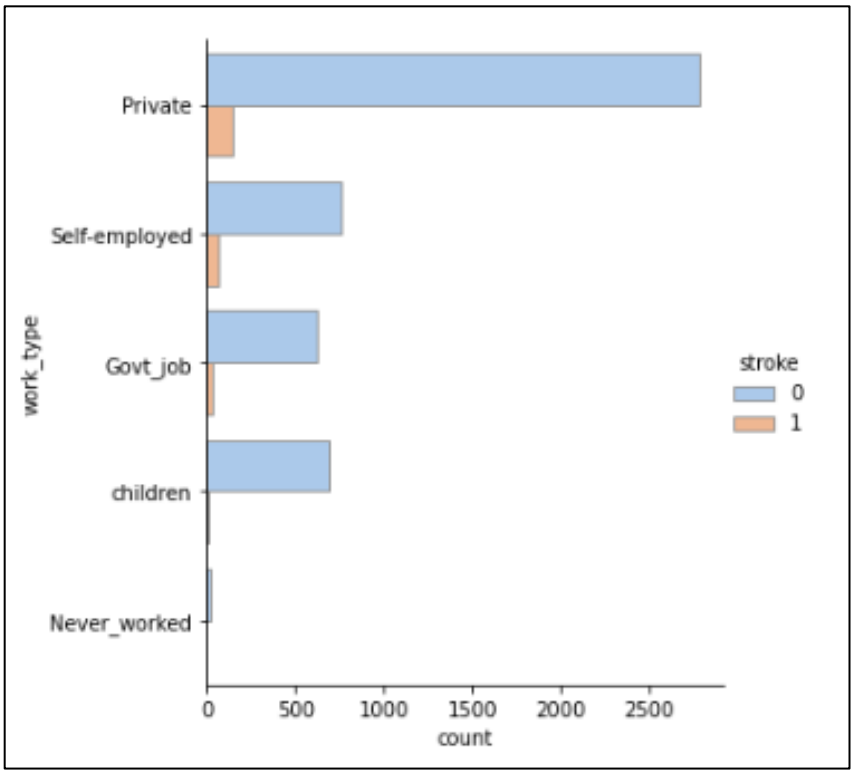


Рисунок 2.10 – Розподіл людей за їх працевлаштуванням

Було побудовано стовпчасту діаграму, яка відображає кількість людей, які перенесли інсульт залежно від їх статусу куріння. Статуси куріння включають: раніше курили, ніколи не курили, курять та невідомо.

Стовпці розфарбовані залежно від того, чи переносила людина інсульт чи ні. Сині стовпці відображають людей, які не переносили інсульт, а помаранчеві стовпці відображають людей, які перенесли інсульт.

На рисунку 2.11 видно, що інсульт не дуже пов'язаний з курцями, оскільки частка людей, які перенесли інсульт, є досить однаковою серед різних статусів куріння.

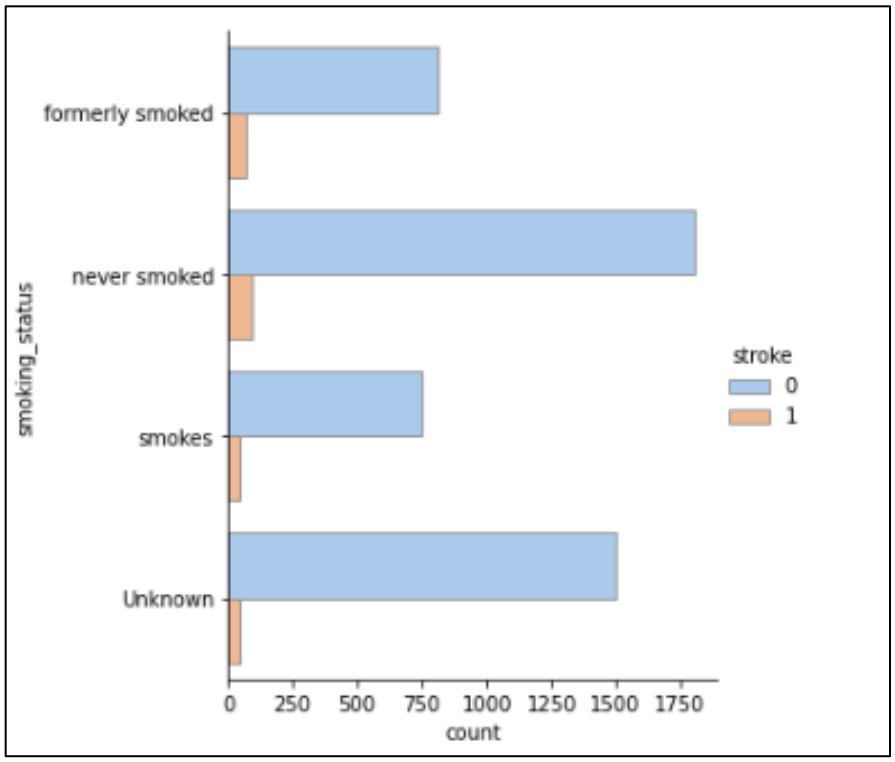


Рисунок 2.11 – Розподіл людей за їх статусом куріння

Було побудовано діаграму яка відображає відсоток людей з хворобами серця за їх статтю. З цього ми можемо зробити висновок, що ризик однаковий (рис. 2.12).

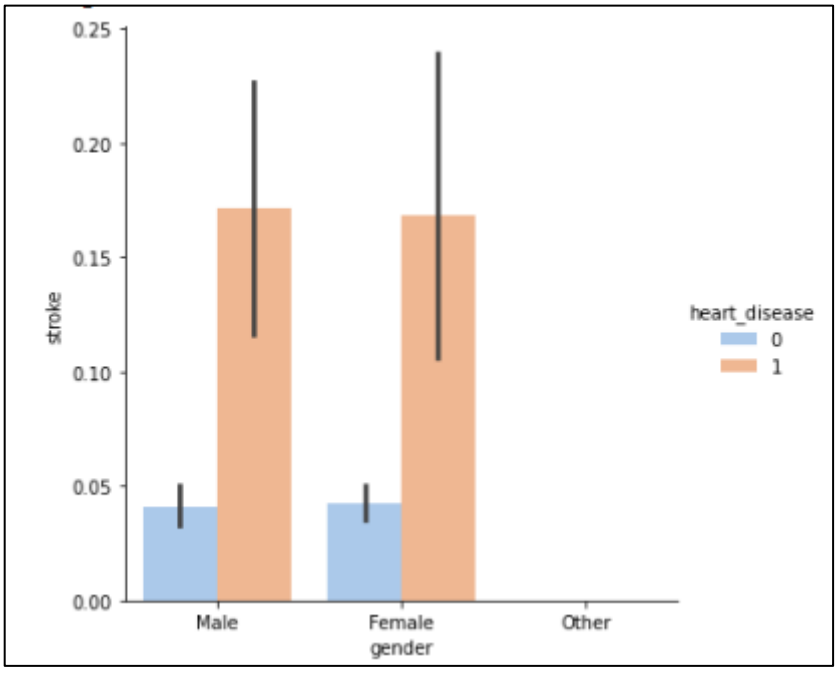


Рисунок 2.12 – Розподіл людей за статтю і хворобами серця

На рисунку 2.13 зображено діаграму, яка відображає розподіл людей за статтю та їх місцем проживання. На ній видно, що місце розташування будинку має значення. Сільські жителі менш схильні до інсульту, ніж жителі великих міст, це свідчить про те, що забруднення навколишнього середовища корелює з інсультом.

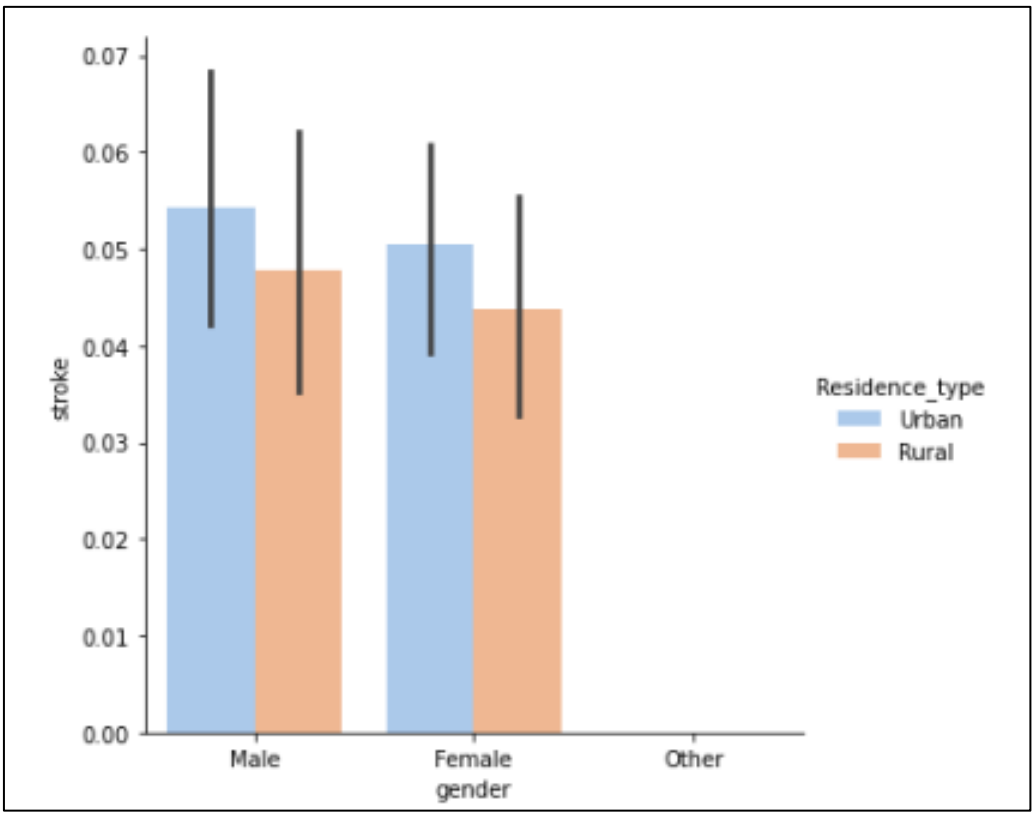


Рисунок 2.13 – Діаграма розподілу людей за їх місцем проживання

Було побудовано діаграму, яка відображає відношення між статтю людини та гіпертонією у зв'язку з інсультом. З цього можна зробити висновок, що ризик виникнення інсульту у людей з гіпертонією набагато більший ніж у здорової людини (рис 2.14).

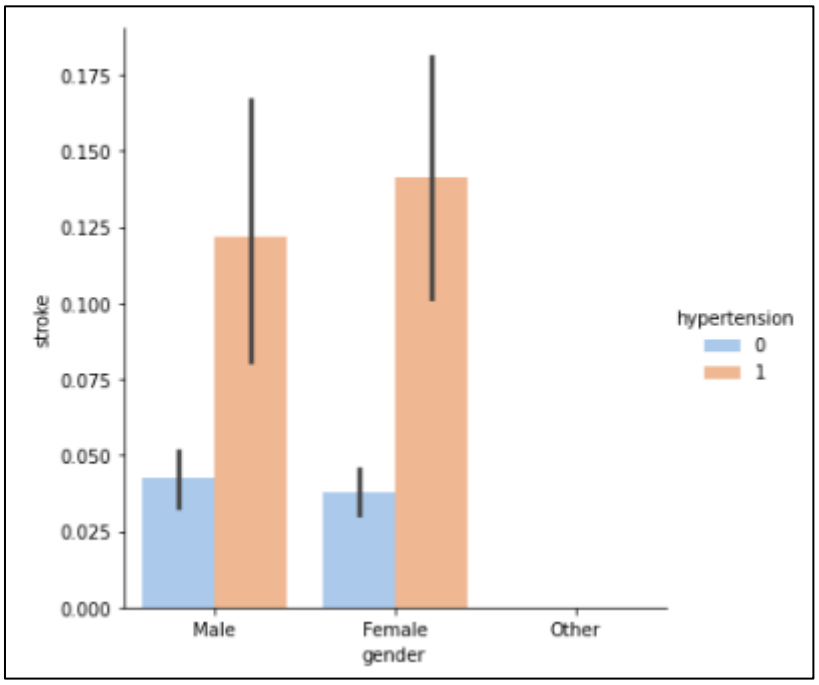


Рисунок 2.14 – Діаграма відношення між статтю та гіпертонією

Кругова діаграма, відображає розподіл випадків інсульту за статтю. Кожен сектор відображає різну групу: чоловіки з інсультом, жінки з інсультом, здорові чоловіки та здорові жінки (рис. 2.15).

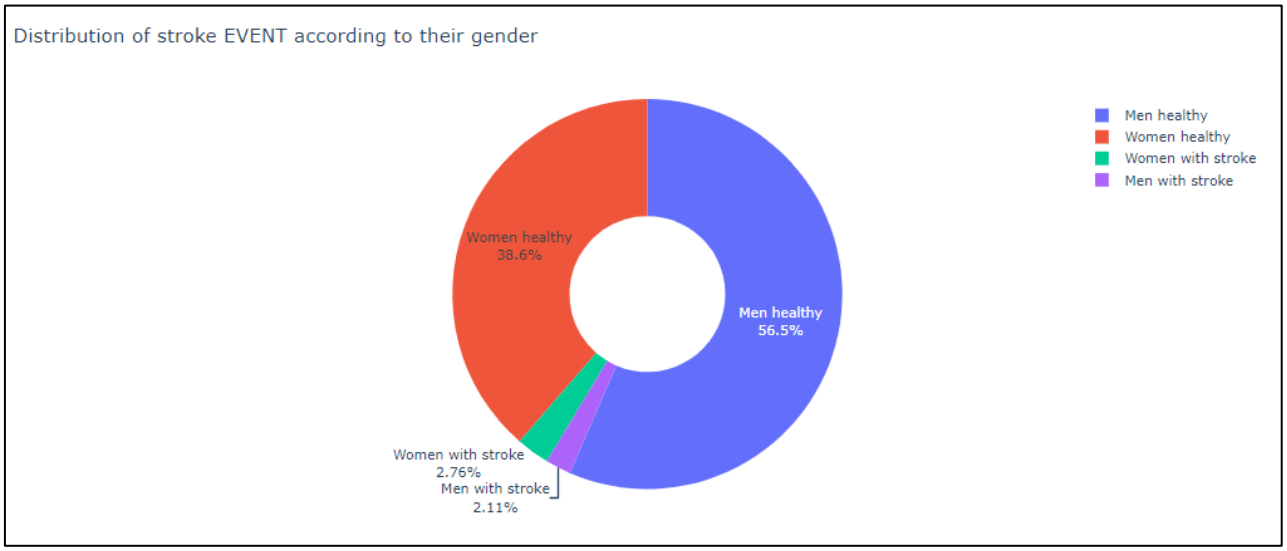


Рисунок 2.15 – Розподіл випадків інсульту за статтю

Було побудовано паралельний координатний графік, який має 8 вертикальних осей, кожна з яких представляє різну змінну: стать, вік,

гіпертонію, хвороби серця, шлюб, тип роботи, тип місця проживання, куріння та інсульт. Графік має два набори ліній, один для чоловіків, а другий - для жінок. Колір лінії буде вказувати на наявність чи відсутність інсульту. Лінії з'єднують різні змінні, показуючи взаємозв'язок між ними. За допомогою цього графіка можна вивчити взаємозв'язок між різними змінними та їх вплив на інсульт. (рис. 2.16)

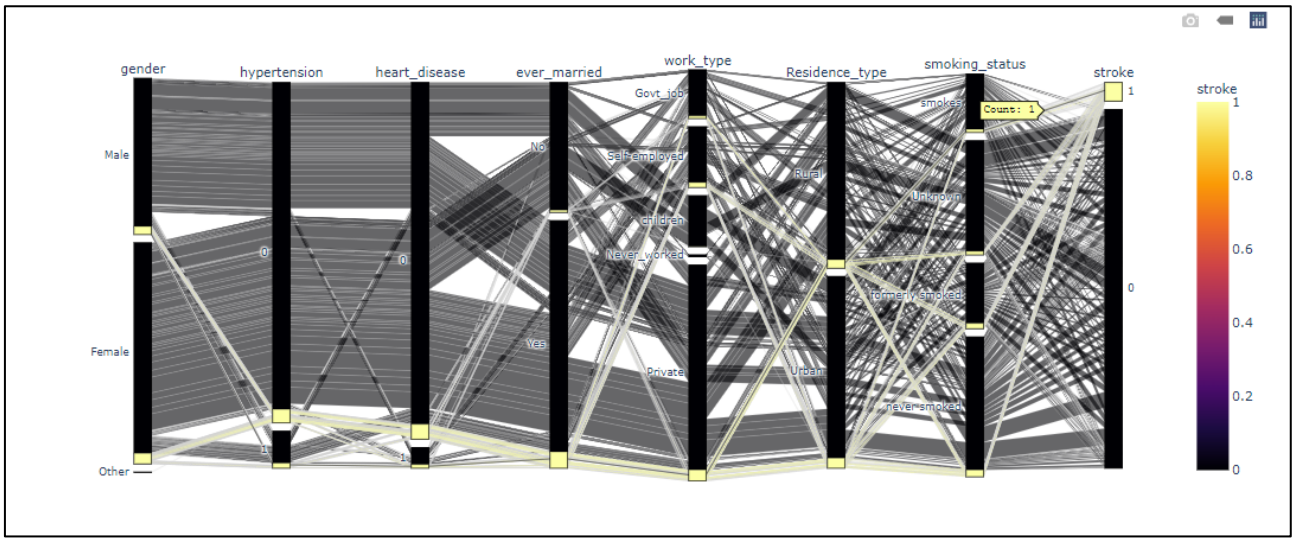


Рисунок 2.16 – Паралельний координатний графік

Було побудовано стовпчасту діаграму, яка відображає кореляцію між різними категоріальними характеристиками та цільовою змінною. Характеристики включають: вік, хвороби серця, гіпертонію, шлюб, статус куріння, проживання, стать та тип роботи. На діаграмі зображено, що кореляція між віком та цільовою змінною є позитивною, що означає, що зі збільшенням віку ймовірність захворювання на серцево-судинне захворювання також збільшується.

Кореляція між наявністю серцево-судинного захворювання та цільовою змінною також є позитивною, що означає, що наявність серцево-судинного захворювання є фактором ризику для інших серцево-судинних захворювань. (рис. 2.17).

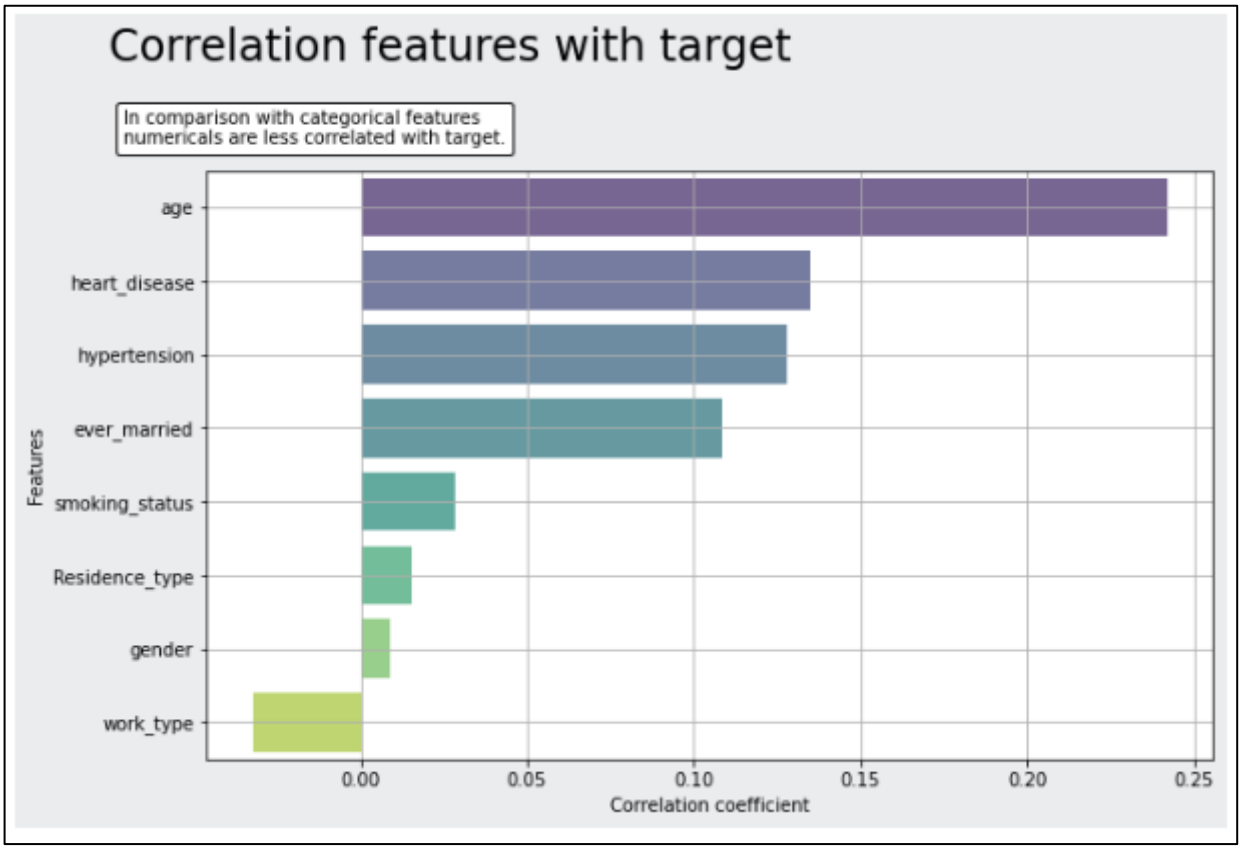


Рисунок 2.17 – Діаграма кореляції

На рисунку 2.18 зображено теплову карту, яка показує кореляцію між різними змінними. Кожен квадрат відображає кореляцію між двома змінними. Колір кожного квадрата вказує на силу та напрям кореляції. Характеристики, які мають значний зв'язок з діагнозом інсульт, включають:

– Люди старшого віку мають більшу ймовірність діагностики серцево-судинного захворювання, ніж молодші люди;

– Чоловіки мають більшу ймовірність діагностики серцево-судинного захворювання, ніж жінки;

– Люди з гіпертонією мають більшу ймовірність діагностики серцево-судинного захворювання, ніж люди без гіпертонії;

– Люди з високим рівнем глюкози в крові мають більшу ймовірність діагностики серцево-судинного захворювання, ніж люди з нормальним рівнем глюкози в крові;

– Люди з надмірною вагою мають більшу ймовірність діагностики серцево-судинного захворювання, ніж люди з нормальною вагою.

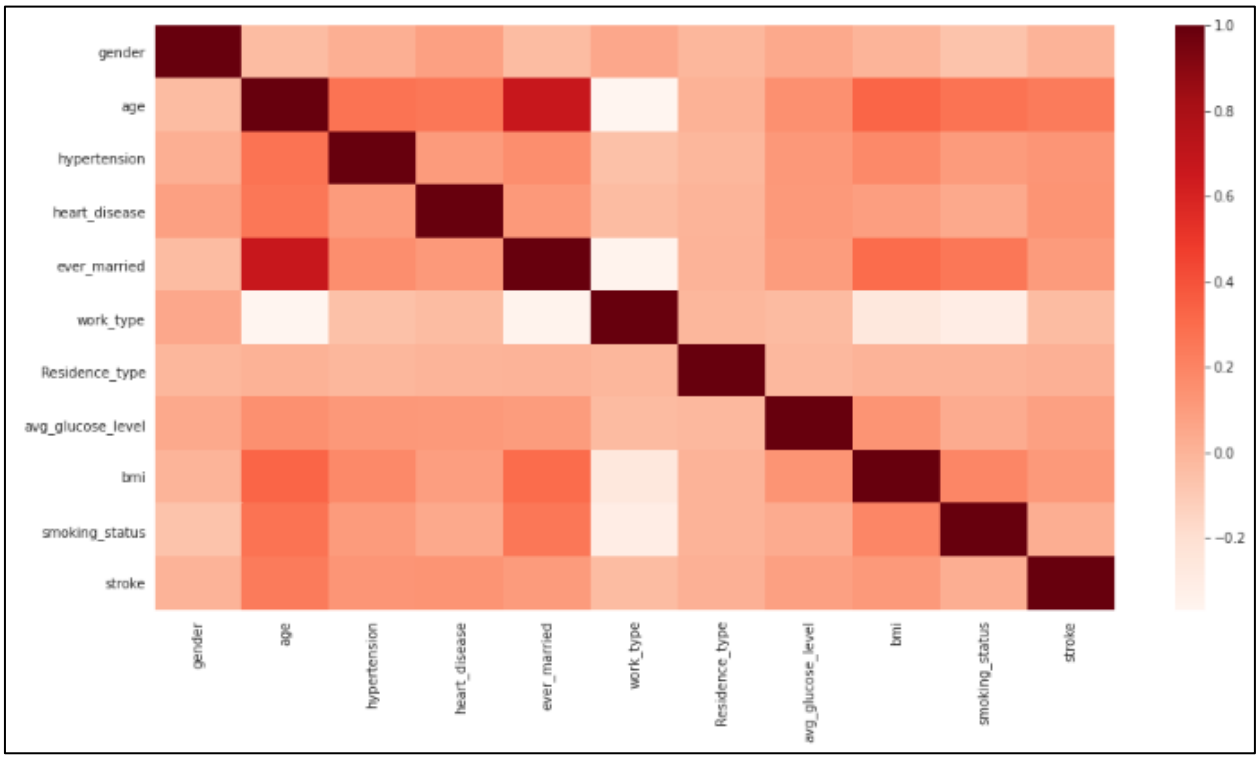


Рисунок 2.18 – Теплова карта

Було побудовано діаграму, яка відображає важливість характеристик для моделі машинного навчання. Найвища важливість - у характеристики 1 з важливістю 0,681038. Найнижча важливість - у характеристики 4 з важливістю 0,024185 (рис. 2.19).

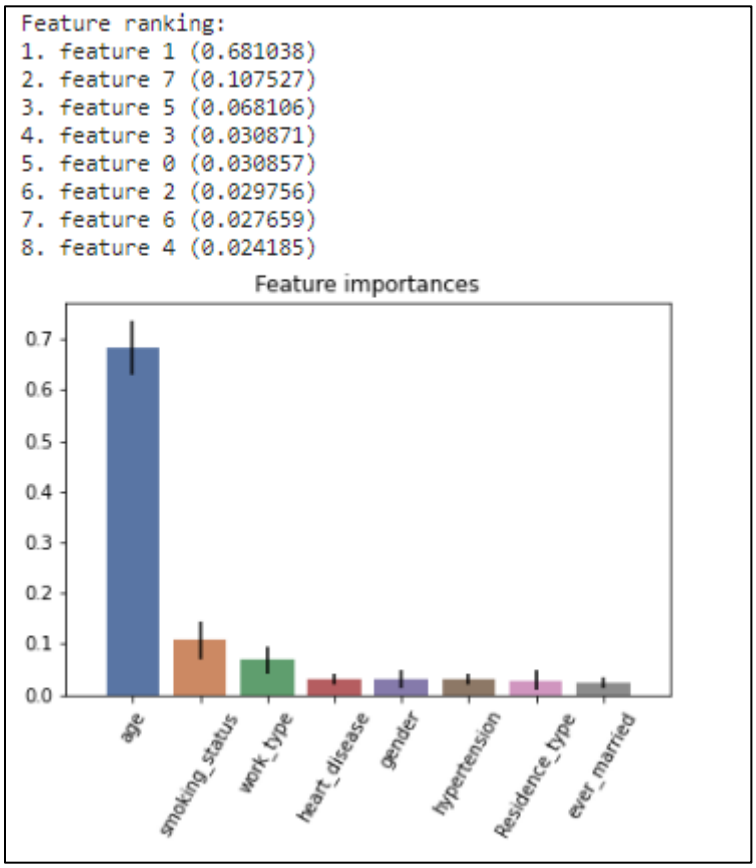


Рисунок 2.19 – Діаграма важливості характеристик для моделі машинного навчання

Для того, щоб зробити набір даних збалансованим, було використано пакет SMOTE для надмірної вибірки. Для виправлення пропущених значень у стовпчиках BMI ми використаємо техніку імплікації на основі KNN.

Техніка імплікації на основі k-найближчих сусідів (KNN) є методом машинного навчання, який використовується для вирішення завдань класифікації та регресії. KNN вважається методом навчання з учителем, і його ідея полягає в тому, щоб призначити новий приклад класу або значення на основі його "найближчих сусідів" в просторі ознак (рис 2.20).

Пакет SMOTE (Synthetic Minority Over-sampling Technique) представляє собою інноваційний метод балансування класів у задачах машинного навчання, зокрема в сферах, де виникає проблема нерівноваги між класами, така як виявлення шахрайства, медична діагностика або будь-яка інша ситуація, де кількість прикладів в класах відрізняється значно.

SMOTE вирішує проблему нерівноваги шляхом створення синтетичних прикладів для меншинового класу. Основна ідея полягає у тому, щоб випадковим чином обрати екземпляри з меншинового класу та створити нові штучні приклади, використовуючи їхні атрибути. Це досягається шляхом вибору випадкового прикладу з меншинового класу та його використання для генерації нового, шляхом врахування випадкового вектора ваг.

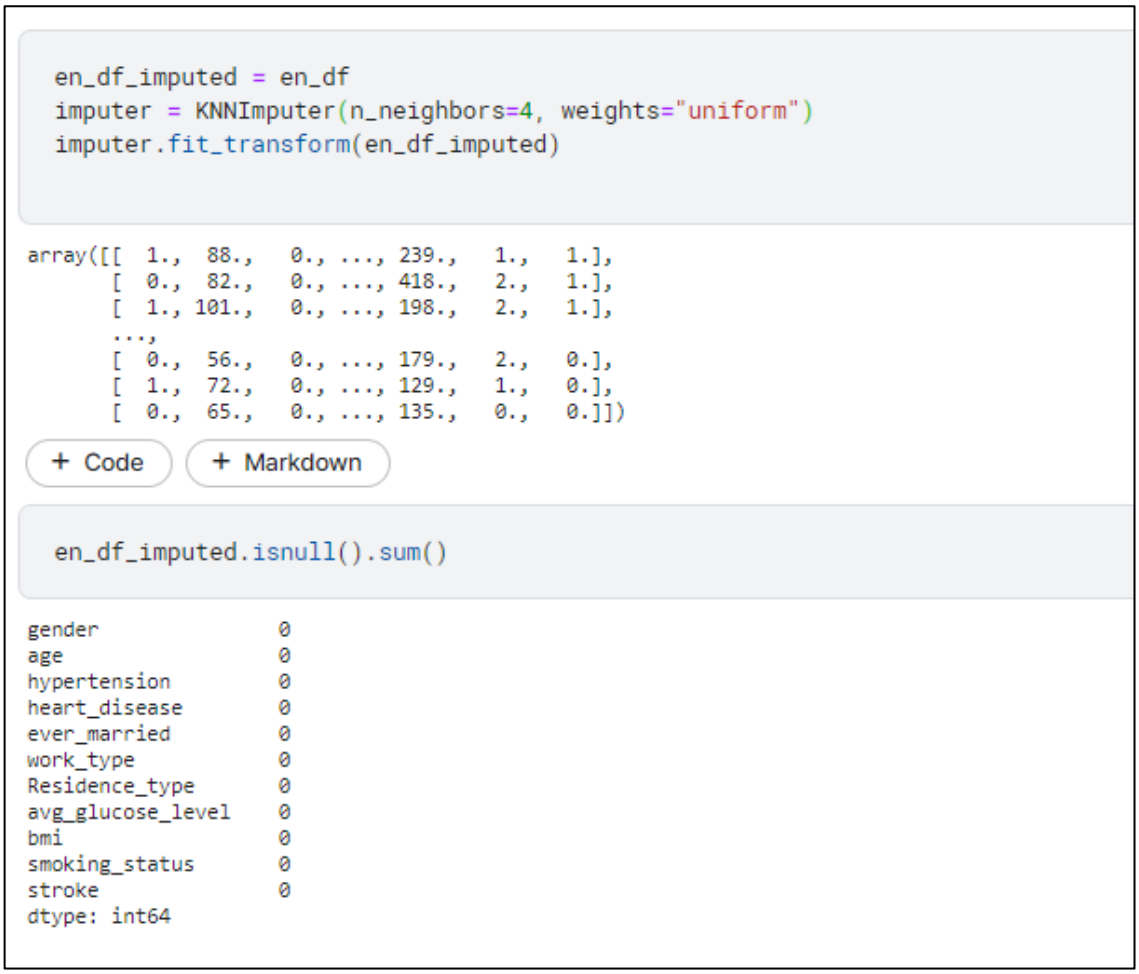


Рисунок 2.20 – KNNImputer для заміни відсутніх значень

Алгоритм SMOTE виконується в кілька етапів:

– Спочатку обирається конкретний приклад з меншинового класу.

– Вибирається один чи кілька сусідів для обраного прикладу. Це може бути випадковим чином вибране підмножина найближчих сусідів;

– Для кожного вибраного сусіда створюється новий синтетичний приклад шляхом врахування різниці між вибраним прикладом та його сусідом.

Це може виконуватися за допомогою формул:

Новий приклад = Обраний приклад + Вага × (Обраний приклад−Сусід)

де вага - це випадкове число між 0 і 1.

SMOTE є потужним інструментом для усунення проблеми нерівноваги класів, проте, як і будь-який метод, він має свої обмеження. Наприклад, він може викликати перенавчання у деяких випадках, і важливо враховувати це під час його використання. Також варто відзначити, що SMOTE не враховує зміну глобальної структури даних і може призводити до надмірного збагачення деяких регіонів простору ознак (рис. 2.21) [30].

```
from imblearn.over_sampling import SMOTE
X , y = en_df_imputed[features],en_df_imputed["stroke"]
x_train,x_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=23)
sm = SMOTE()
X_res, y_res = sm.fit_resample(x_train,y_train)

print("Before OverSampling, counts of label '1': {}".format(sum(y==1)))
print("Before OverSampling, counts of label '0': {} \n".format(sum(y==0)))

print('After OverSampling, the shape of train_X: {}'.format(X_res.shape))
print('After OverSampling, the shape of train_y: {} \n'.format(y_res.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_res==1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_res==0)))
```

```
Before OverSampling, counts of label '1': 249
Before OverSampling, counts of label '0': 4861

After OverSampling, the shape of train_X: (7788, 8)
After OverSampling, the shape of train_y: (7788,)

After OverSampling, counts of label '1': 3894
After OverSampling, counts of label '0': 3894
```

Рисунок 2.21 – Використання SMOTE для збалансування даних

Отже, на основі проведеного аналізу даних встановлено, що інсультом частіше захворюють люди похилого віку, із зайвою вагою, курців та люди із вадами серця.

2.4 Висновки

В цьому розділі обрано середовище для розробки інформаційної технології передбачення хворих на інсульт. Виконано розвідувальний аналіз, який показав, що інсультом частіше хворіють люди похилого віку, із зайвою вагою, курців та люди із вадами серця. Також виконано балансування датасету з використанням методу SMOTE, що дозволило збалансувати датасет.

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

3.1 Розроблення математичної моделі

Опишемо моделі класифікації які було використано в дипломній роботі: Logistic Regression, KNearest Neighbors, Decision Tree Classifier, Random Forest Classifier, Ada Boost, Support Vector Machine (SVM), XGBoost, .

Логістична регресія (англ. Logistic Regression) — статистичний регресійний метод, що застосовують у випадку, коли залежна змінна є бінарною, тобто може набувати тільки двох значень (0 або 1).

Логістична регресія описує залежність між залежною змінною Y та незалежними змінними $X_1, X_2, \dots X_p$ за допомогою логістичної функції:

$$p(Y = 1 \mid X_1, X_2, \dots, X_p) = \frac{1}{1+e^{-\beta_0-\beta_1 X_1-\beta_2 X_2-\dots.-\beta_p X_p}} \quad (3.1)$$

де $\beta_0,\beta_1,\dots,\beta_p$ — параметри моделі, які оцінюються за допомогою методів регресійного аналізу.

Логістична функція відображає ймовірність того, що залежна змінна Y прийме значення 1, як функцію від значення суми незалежних змінних $X1,X2,\dots,Xp$.

Алгорим роботи Logistic regression зображено на рисунку 3.1.

Як видно з графіка, логістична функція має S-подібну форму. Це означає, що ймовірність того, що залежна змінна прийме значення 1, поступово зростає зі збільшенням значення незалежних змінних.

Наприклад, якщо незалежні змінні мають значення 0, то ймовірність того, що залежна змінна прийме значення 1, дорівнює 0,5. Якщо незалежні змінні мають значення, що наближається до нескінченності, то ймовірність того, що залежна змінна прийме значення 1, дорівнює 1.

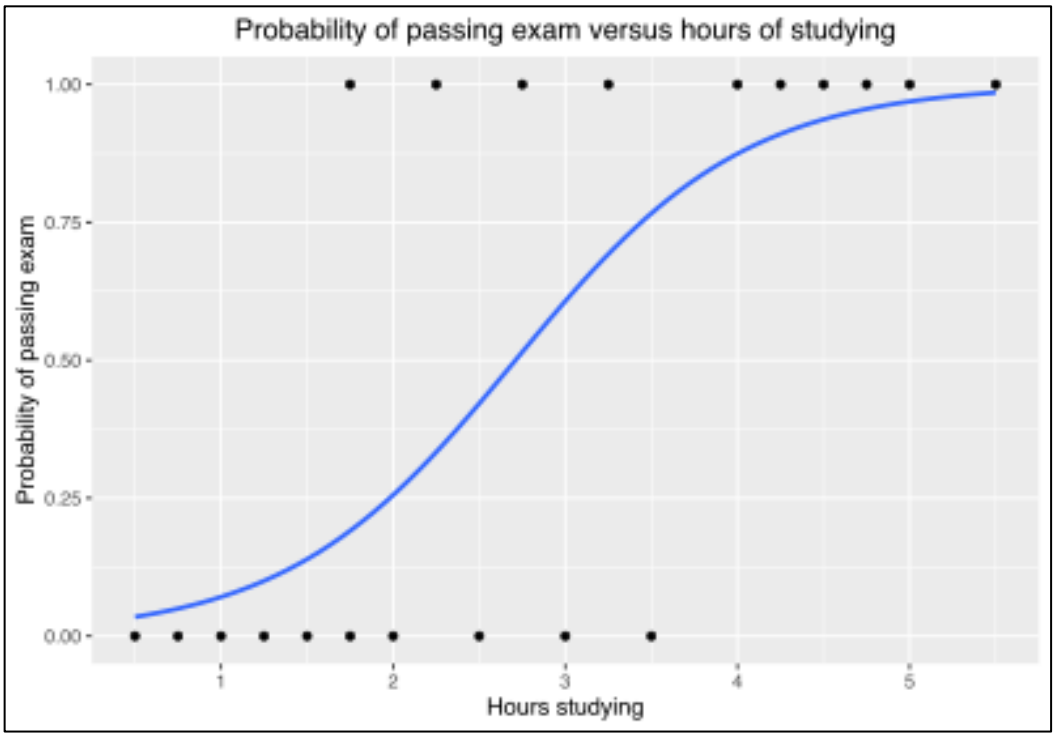


Рисунок 3.1 – алгоритм Logistic regression

AdaBoost (Adaptive Boosting) — це мета-алгоритм машинного навчання, який використовується для класифікації та регресії. AdaBoost працює шляхом послідовного створення слабких класифікаторів і адаптації їх до даних.

Алгоритм AdaBoost спрямований на поєднання кількох слабких класифікаторів з метою створення потужного класифікатора.

Слабкий класифікатор — це класифікатор, який має відносно низьку точність. У AdaBoost часто використовуються такі слабкі класифікатори, як дерева рішень з одним вузлом.

Розглянемо набір даних, що містить N точок або рядків в наборі даних…

$$x_i \in R^n, y^i \in \{-1, 1\}. \quad (3.2)$$

У цьому випадку:

– N визначає розмірність дійсних чисел або кількість атрибутів у нашому наборі даних.

– X представляє собою множину даних.

– Y є цільовою змінною, що приймає значення -1 або 1, оскільки це проблема бінарної класифікації, що вказує на перший або другий клас.

Спочатку всі точки даних матимуть однакову зважену вибірку w. Де N - це загальна кількість точок даних:

$$\omega = 1/\ N\ \epsilon\ [0, 1]. \tag{3.3}$$

У випадку зважених вибірок завжди рівні одиниці, тому значення кожної окремої ваги завжди буде в діапазоні від 0 до 1.

Для визначення фактичного впливу класифікатора при класифікації використовується формула:

$$\alpha_t = \frac{1}{2} ln \frac{(1-TotalError)}{TotalError}. \tag{3.4}$$

Ваги вибірки, які спочатку прийняті як 1/N для кожної точки даних, обчислюються за формулою:

$$\omega_i = \ \omega_{i-1} * \ e^{\pm\alpha}. \tag{3.5}$$

Другими словами, вага нового зразка буде рівна вазі старого зразка, помноженій на число Ейлера.

Дерево рішень (Decision Tree Classifier) — це алгоритм машинного навчання, який використовується для класифікації даних. Дерево рішень створюється шляхом поділу даних на два або більше підмножин на основі певних критеріїв. Цей процес повторюється до тих пір, поки кожна підмножина не буде однорідною.

Критерії поділу:

Існує кілька різних критеріїв, які можна використовувати для поділу даних на підмножини. Найпоширенішими критеріями є:

Інформаційність (information gain) — цей критерій міряє, скільки інформації дає нам поділ даних на два підмножини.

Індекс Джини (Gini index) — цей критерій міряє, наскільки однорідні підмножини, що утворюються в результаті поділу даних.

Математична формула інформаційності. Інформаційність (information gain) міряє, скільки інформації дає нам поділ даних на два підмножини. Вона обчислюється як різниця в ентропії між початковим набором даних і двома підмножинами:

$$\text{Information Gain(A)=Entropy(S) - [}Entropy(S\ left) * P(S\ left) + Entropy\ (S\ right) * P\ (S\ right)\text{]}, \quad (3.6)$$

де A - атрибут, який використовується для поділу даних;

S - початковий набір даних;

S_left і S_right - два підмножини даних;

P(S_left) і P(S_right) - ймовірності S_left і S_right, відповідно.

Математична формула ентропії(entropy) міряє невизначеність або хаос у наборі даних. Вона обчислюється як сума добутку ймовірності кожного класу та логарифму цієї ймовірності:

$$\text{Entropy(S)} = -\Sigma\ P(c) * \log2(P(c)), \quad (3.7)$$

де S - набір даних;

c - мітка класу;

P(c) - ймовірність класу.

Математична формула індексу Джині (Gini index) - ще один мірник нечистоти у наборі даних. Він обчислюється як сума квадратів ймовірностей кожного класу.

$$\text{Gini Index(S)} = 1 - \Sigma\ P(c)^2 \quad (3.8)$$

де S - набір даних;

c - мітка класу;

P(c) - ймовірність класу.

Алгоритм Decision Tree Classifier вибирає атрибут, який максимізує інформаційність або мінімізує індекс Джині. Це гарантує, що розділення є максимально інформативним і призводить до більш однорідних підмножин даних.

На рисунку 3.2 зображено алгоритм роботи Decision Tree Classifier

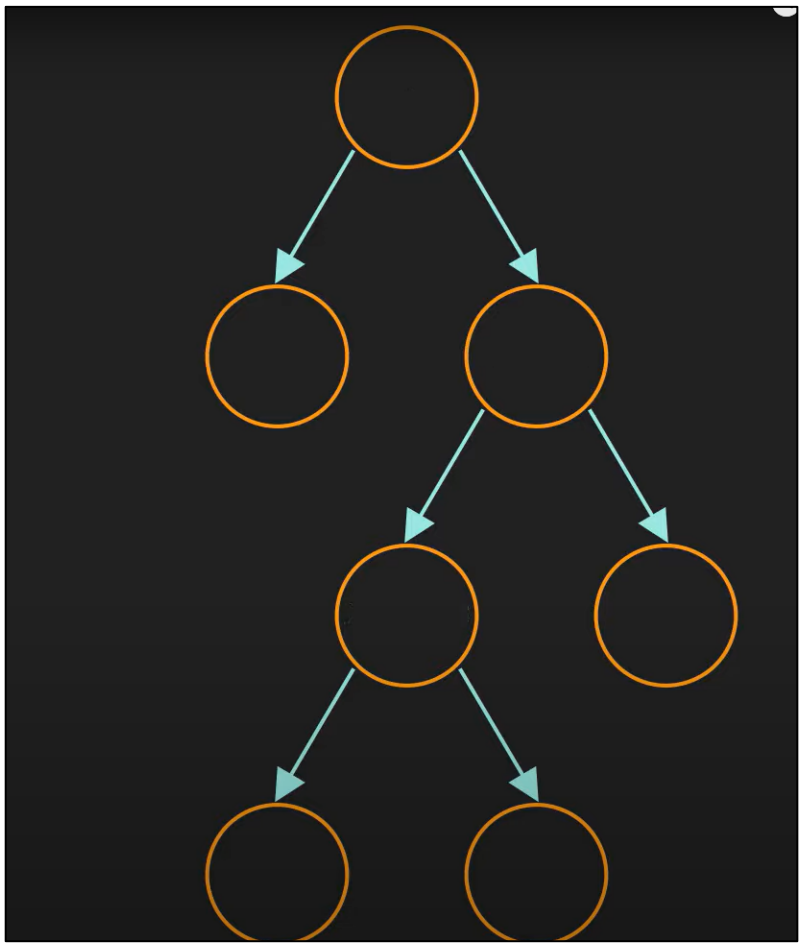


Рисунок 3.2 – Алгоритм роботи Decision Tree Classifier

XGBoost - це алгоритм машинного навчання, який використовується для класифікації та регресії. Він заснований на методі градієнтного бустингу, який використовує ансамбль слабких класифікаторів для прогнозування.

Принцип роботи

XGBoost працює, ґрунтуючись на наступному принципі:

– Спочатку створюється слабкий класифікатор;

– Потім оцінюється похибка слабкого класифікатора;

– Похибка слабкого класифікатора використовується для створення сильного класифікатора;

– Процес повторюється до тих пір, поки не буде досягнута заданої точності.

XGBoost має вбудовану підтримку паралельної обробки, що робить можливим навчання моделей на великих наборах даних за розумний час.

XGBoost можна використовувати в різних застосуваннях, включаючи змагання Kaggle, системи рекомендацій та прогнозування коефіцієнта клікабельності, серед іншого.

Основним принципом XGBoost є оптимізація функції втрат через градієнтний спуск. Алгоритм використовує ансамбль дерев рішень та додає їх з врахуванням градієнту функції втрат. До цього можуть бути додані регуляризаційні компоненти для контролю складності моделі.

Функція об'єктиву (функція втрат і регуляризації) на ітерації t, яку ми повинні мінімізувати, є наступною

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} l\,(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \qquad (3.9)$$

де l - функція втрат,

y^i(t−1) - прогноз на попередньому кроці,

ft(xi) - нова модель, яку ми додаємо,

Ω - функція регуляризації.

Другий порядок апроксимації Тейлора для функції об'єктиву:

$$\text{Obj}^{(t)} \approx \sum_{i=1}^{n} [\, l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \qquad (3.10)$$

де gi=∂y^i(t−1)l(yi,y^i(t−1)) і hi=∂y^i(t−1)2l(yi,y^i(t−1)) - це перший і другий порядок градієнтних статистик функції втрат.

Ці формули використовуються для оптимізації моделі на кожному кроці алгоритму.

Random Forest Classifier — це універсальний і потужний алгоритм машинного навчання, який добре справляється з завданнями класифікації. Це метод ансамблевого навчання, що означає, що він поєднує кілька дерев рішень для прогнозування. Поєднуючи кілька дерев рішень, Random Forest Classifier може зменшити перенавчання та покращити загальну продуктивність моделі.

Алгоритм Random Forest Classifier працює за наступним принципом:

– Спочатку створюється випадкова підмножина навчальних даних;

– На основі цієї підмножини створюється дерево рішень;

– Повторюється пункт 1-2 для заданої кількості дерев;

– Для нового зразка даних робиться прогноз, шляхом голосування більшості серед усіх дерев.

Алгоритм Random Forest Classifier має ряд переваг порівняно з іншими алгоритмами класифікації:

– Висока точність: Random Forest Classifier постійно досягає високої точності в широкому діапазоні завдань класифікації;

– Роботоздатність до відхилень: Random Forest Classifier менш чутливий до відхилень і шумних даних, що робить його стійким у реальних сценаріях застосування;

– Важливість ознак: Random Forest Classifier надає міру важливості ознак, що вказує на відносний внесок кожної ознаки в прогнози моделі;

– Не вимагається масштабування ознак: Random Forest Classifier не вимагає масштабування ознак, що робить його простішим у реалізації та менш сприйнятливим до проблем масштабування;

– Розумність: Random Forest Classifier відносно зрозумілий, оскільки окремі дерева рішень надають уявлення про процес міркування моделі.

На рисунку 3.3 зображено алгоритм роботи Random Forest Classifier.

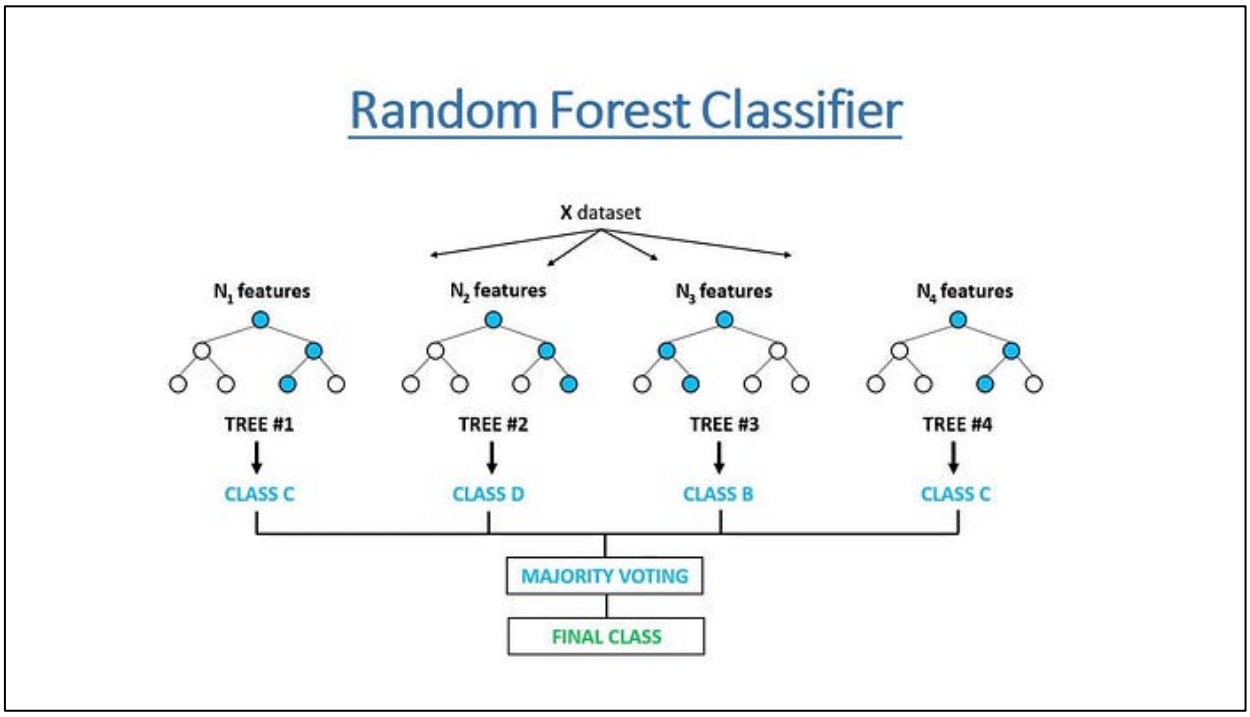


Рисунок 3.3 – Алгоритм роботи Random Forest Classifier

Support Vector Machine – це алгоритм навчання з учителем, який використовується для задач класифікації та регресії. Це один з найпопулярніших і ефективних алгоритмів для задач класифікації. SVM засновані на ідеї пошуку гіперплощини, яка найкраще розділяє два класи даних. Гіперплощина - це площина рішень, яка максимізує зазор між двома класами. Зазор - це відстань між гіперплощиною та найближчими точками даних з кожного класу, які називаються опорними векторами.

Алгоритм SVM може використовуватися для як для лінійної, так і для нелінійної класифікації. Для лінійної класифікації гіперплощина є прямою. Для нелінійної класифікації гіперплощина може мати будь-яку форму, і це досягається за допомогою функції ядра. Функція ядра - це функція, яка відображає точки даних з простору введення в простір з більшою розмірністю. У просторі ознак точки даних більш ймовірно є лінійно розділеними.

Алгоритм роботи Support Vector Machine (SVM)

– Попередня обробка даних;
– Навчання моделі;

– Оцінка точності моделі та використання моделі.

Модель SVM може бути використана для вирішення різних задач класифікації, таких як: розпізнавання образів, класифікація текстових даних, класифікація медичних даних.

Алгоритм роботи Gradient Boosting - це метод машинного навчання, який спробовує покращити прогнози моделі, додаваючи до неї нові моделі, які коригують помилки попередніх. Цей метод часто використовується для задач регресії та класифікації. Основна ідея полягає в тому, щоб побудувати послідовні моделі, кожна з яких намагається виправити помилки попередньої моделі.

Основний алгоритм Gradient Boosting можна подати так:

– Ініціалізуємо базову модель для передбачення середнього значення цільової змінної;

– Розраховуємо помилки між прогнозами базової моделі та фактичними значеннями;

– Будуємо нову модель, яка намагається виправити ці помилки. Для цього використовуємо градієнт функції втрати;

– Оновлюємо прогнози, додаючи до них прогнози нової моделі з вагою (зазвичай меншим за 1);

– Повторюємо кроки 2-4 декілька разів, додаючи нові моделі та коригуючи прогнози;

– Фінальний прогноз отримується шляхом додавання прогнозів всіх побудованих моделей.

На рисунку 3.4 зображено алгоритм роботи Gradient Boosting Classifier.

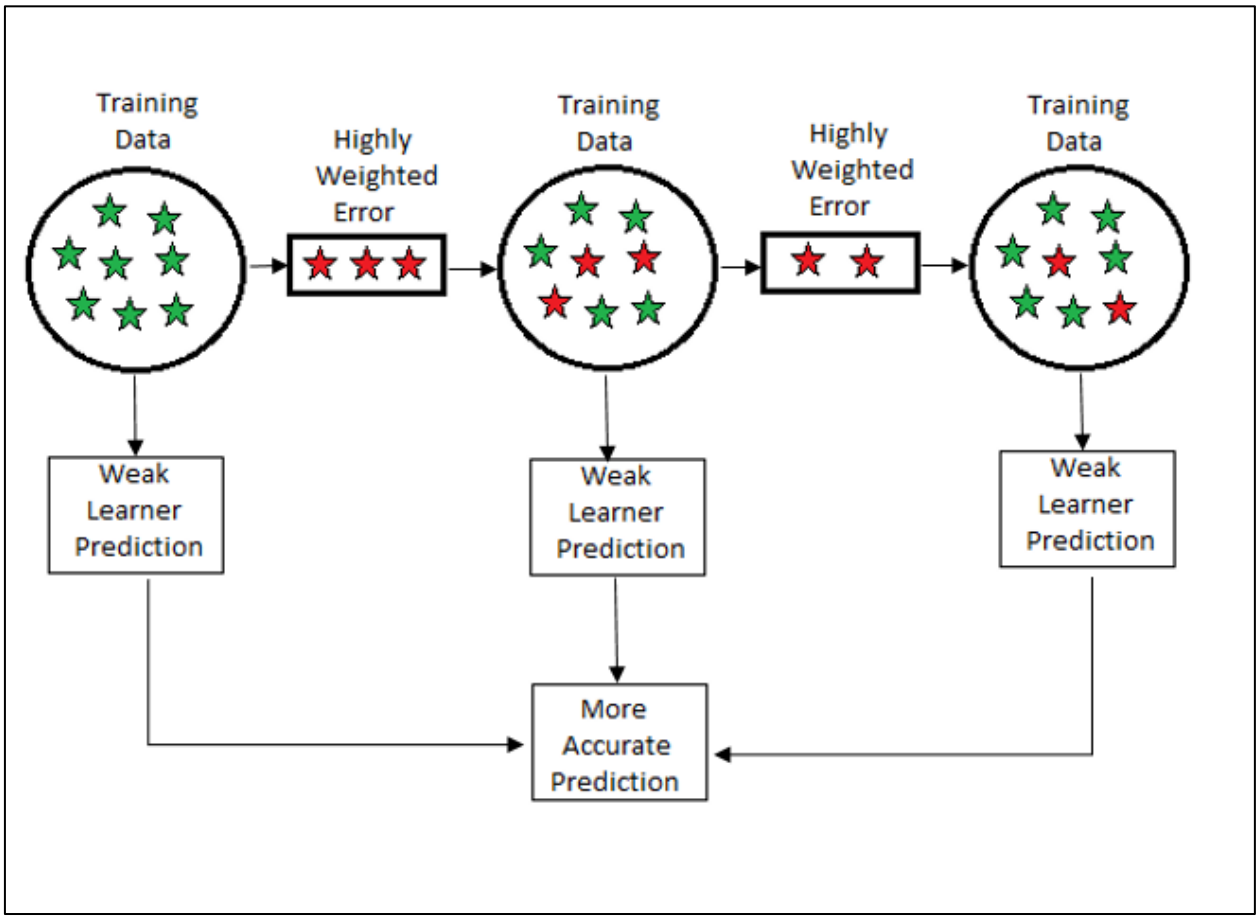


Рисунок 3.4 – Алгоритм роботи Gradient Boosting Classifier

3.2 Реалізація моделей для аналізу даних

Для початку моделювання та тестування було виконано розподіл даних та тестування

Конфузійна матриця - це інструмент для оцінювання якості роботи алгоритмів класифікації в машинному навчанні. Вона використовується для порівняння прогнозованих результатів класифікації з фактичними класами даних.

Конфузійна матриця виглядає як таблиця, в якій рядки представляють фактичні класи, а стовпці - прогнозовані класи. Кожна клітина матриці показує кількість прикладів, що належать конкретному класу за фактом і прогнозом. Зазвичай конфузійна матриця використовується для задач з бінарною або багатокласовою класифікацією.

Основні елементи конфузійної матриці:

– True Positives кількість прикладів, які належать певному класу і були правильно класифіковані як цей клас;

– True Negatives кількість прикладів, які не належать певному класу і були правильно класифіковані як інші класи або відсутні;

– False Positives кількість прикладів, які не належать певному класу, але були неправильно класифіковані як цей клас;

– False Negatives кількість прикладів, які належать певному класу, але були неправильно класифіковані як інші класи або відсутні.

На основі цих елементів можна обчислити різні метрики, такі як точність (accuracy), чутливість (recall), специфічність (specificity) та інші, які допомагають оцінити ефективність алгоритму класифікації.

На рисунку 3.5 зображено код який включає імпорт різних моделей машинного навчання з бібліотеки sklearn.

Створено список models, який містить екземпляри цих моделей з певними параметрами. Також є список model_names, який містить назви цих моделей для подальшого використання.

```python
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from catboost import CatBoostClassifier

# Models
models = [KNeighborsClassifier(n_neighbors=3),
          SVC(probability=True),
          DecisionTreeClassifier(random_state=2022),
          RandomForestClassifier(random_state=2022),
          XGBClassifier(random_state=2022),
          CatBoostClassifier(random_state=2022, verbose=0),
          LogisticRegression(random_state=2022),
          KNeighborsClassifier(),  # You can customize parameters for KNearest
          AdaBoostClassifier(random_state=2022)]

model_names = ['KNN', 'SVM', 'Decision Tree', 'Random Forest', 'XGBoost', 'CatBoost', 'Logistic Regression', 'KNearest', 'AdaBoost']
```

Рисунок 3.5 – Імпорт моделей

На рисунку 3.6 зображено код який включає функцію build_models, яка приймає на вхід набір моделей, їх назви, а також тренувальні та тестові дані. Функція тренує кожну модель на тренувальних даних, робить прогнози для тренувальних та тестових даних, обчислює метрики точності та ROC AUC, а також будує матрицю помилок для кожної моделі.

Результати зберігаються у вигляді списків, які потім об’єднуються в словник results. Цей словник перетворюється на DataFrame models_scores_df, який повертається як результат функції. Цей код дозволяє легко порівняти ефективність різних моделей машинного навчання на одних і тих самих даних. За допомогою цього можна визначити, яка модель найкраще працює для конкретного набору даних.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, roc_auc_score, plot_confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from tabulate import tabulate

# Models
models = [
    KNeighborsClassifier(n_neighbors=3),
    make_pipeline(StandardScaler(), SVC(probability=True)),
    DecisionTreeClassifier(random_state=2022),
    RandomForestClassifier(random_state=2022),
    XGBClassifier(random_state=2022),
    LogisticRegression(random_state=2022),
    KNeighborsClassifier(),  # You can customize parameters for KNearest
    AdaBoostClassifier(random_state=2022),
    GradientBoostingClassifier(random_state=2022)  # Adding Gradient Boosting Classifier
]

model_names = [
    'KNN', 'SVM', 'Decision Tree', 'Random Forest', 'XGBoost',
    'Logistic Regression', 'KNearest', 'AdaBoost', 'Gradient Boosting'  # Updating model_names
]

def build_models(models, model_names, X_train, X_test, y_train, y_test):
    # Let's create empty lists to append the results
    roc_auc_scores = []
    accuracy_scores_train = []  # to store training accuracy
    accuracy_scores_test = []  # to store testing accuracy
    trained_models = []  # to store trained models
    results = {}

    fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(18, 18))  # Increased figure size
    # Use enumerate() and zip() function to iterate the lists
    for idx, (ml_model_names, ml_models, ax) in enumerate(zip(model_names, models, axes.flatten())):
        clf = models[idx]
```

Рисунок 3.6 – Побудова конфузійної матриці

На рисунку 3.7 побудовано конфузійну матрицю з результатами моделі Logistic Regression.

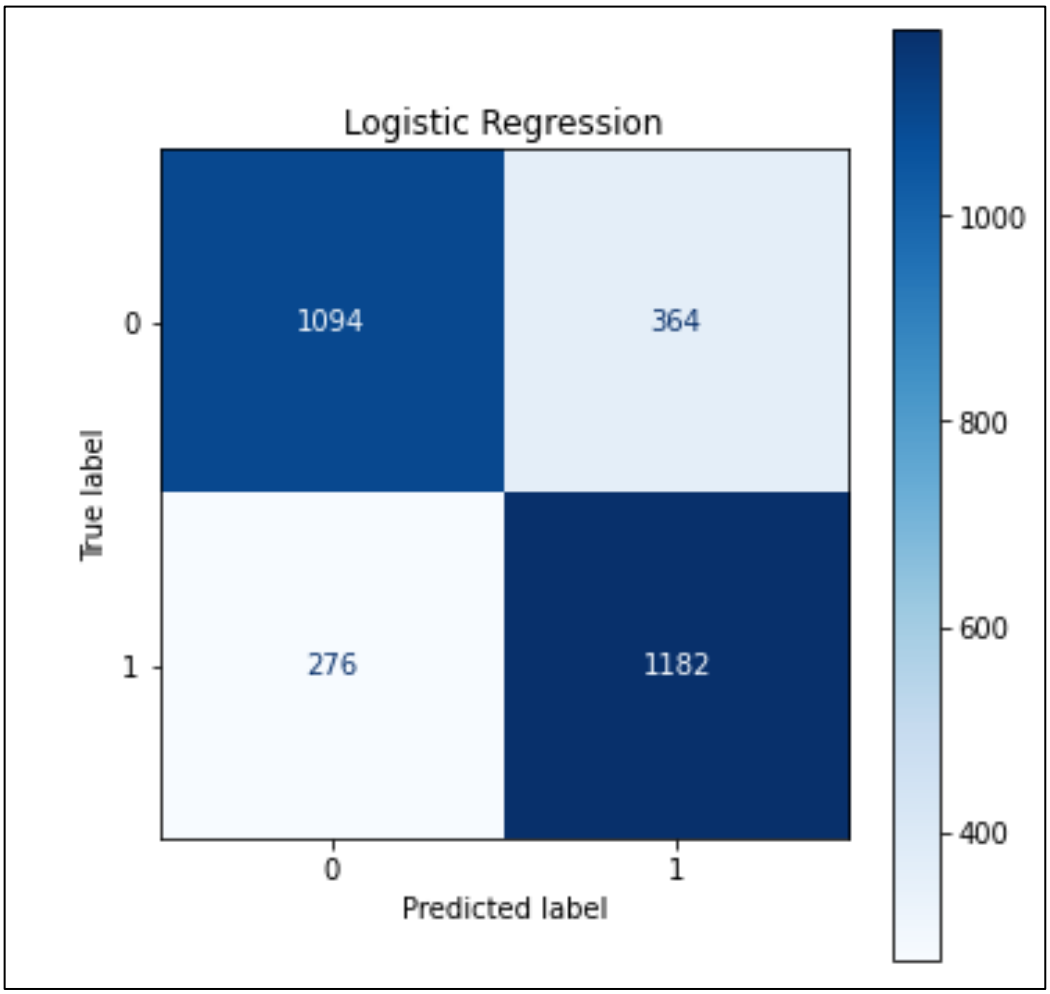


Рисунок 3.7 – Конфузійна матриця Logistic Regression

Було відображено тестові та треновані результати моделі Logistic Regression. У цьому випадку модель правильно передбачала результат у 94,6% випадків. Це означає, що модель є досить точною для прогнозування інсульту (рис. 3.8).

```
Model: Logistic Regression
Training Accuracy: 0.7855673133450911
Testing Accuracy: 0.782235939643347
Roc Auc Score: 0.8541343253531436
```

Рисунок 3.8 – Результати тестової та тренованої моделі

На рисунку 3.9 побудовано конфузійну матрицю з результатами моделі KNearest. Ця матриця допоможе оцінити вимірювання продуктивності моделі класифікації.

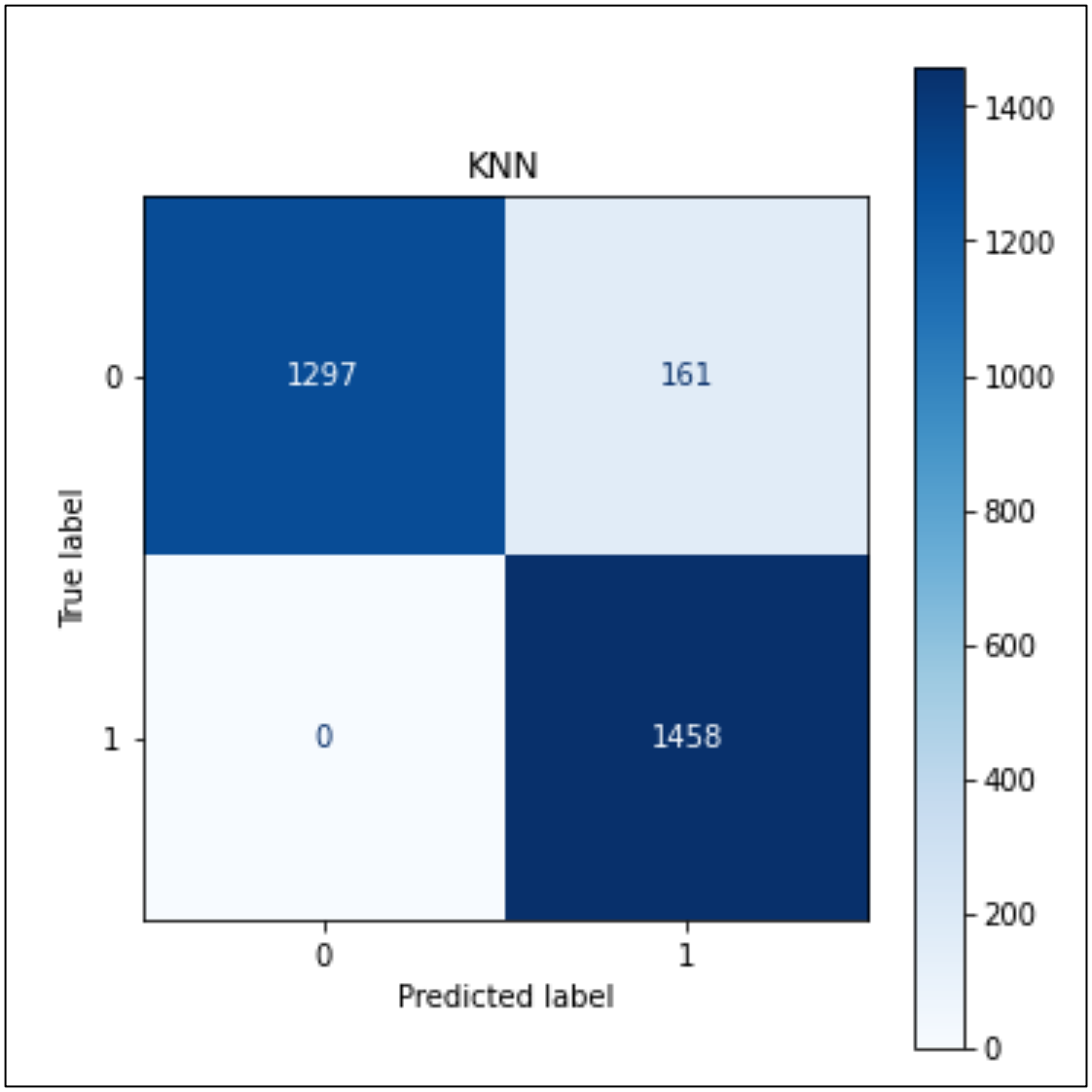


Рисунок 3.9 – Конфузійна матриця KNN

Було відображено тестові та треновані результати моделі KNearest показала хороші результати при класифікації даних. Загальна точність моделі на тестових даних становить 0,944. Це означає, що модель має високу ймовірність правильно класифікувати дані (рис. 3.10)

```
Model: KNearest
Training Accuracy: 0.9431216931216931
Testing Accuracy: 0.9204389574759945
Roc Auc Score: 0.9694641550049771
```

Рисунок 3.10 – Результати тестової та тренованої моделі

На рисунку 3.11 зображена конфузійна матриця для моделі Decision Tree Classifier.

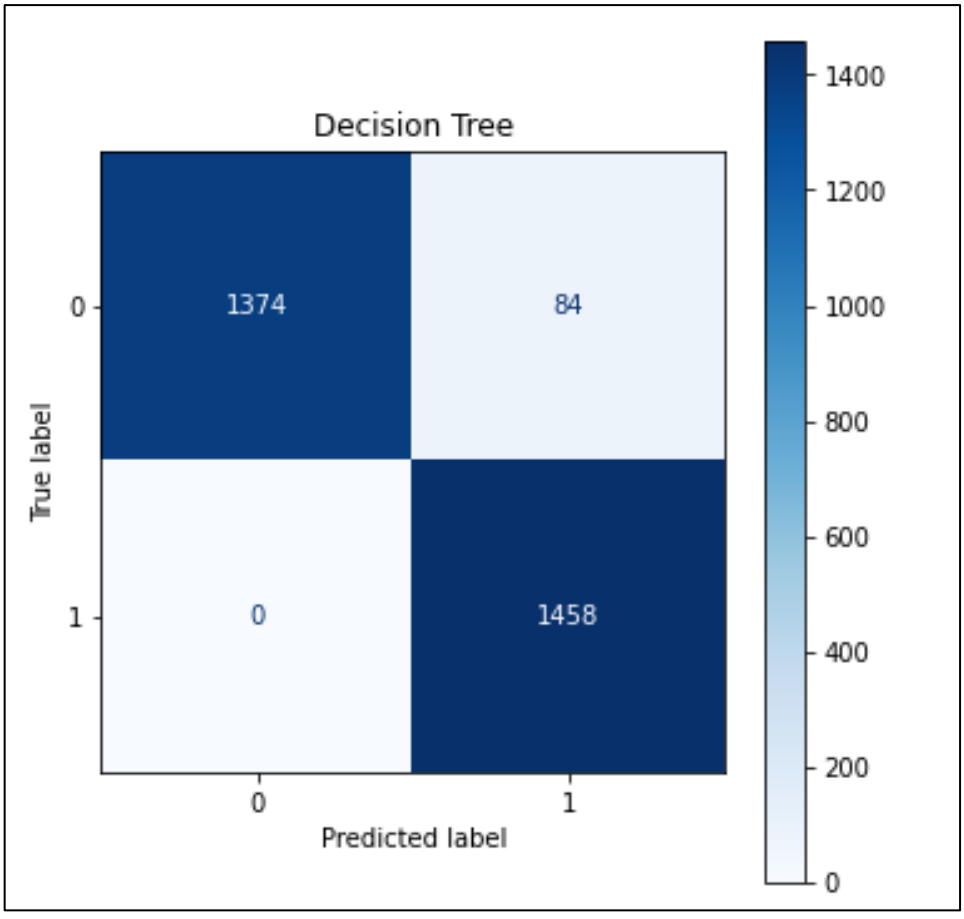


Рисунок 3.11 – Конфузійна матриця Decision Tree

Було відображено тестові та треновані результати моделі . Ці результати вказують на те, що модель є досить точною при навчанні, але її точність знижується при роботі з тестовими даними (рис. 3.12).

```
Model: Decision Tree
Training Accuracy: 0.9315108759553204
Testing Accuracy: 0.9190672153635117
Roc Auc Score: 0.9399716995865957
```

Рисунок 3.12 – Результати тестової та тренованої моделі

На рисунку 3.13 зображена матриця для моделі Random Forest, з результатами. На ній зображена кількість випадків коли модель правильно, або не правильно передбачила клас.

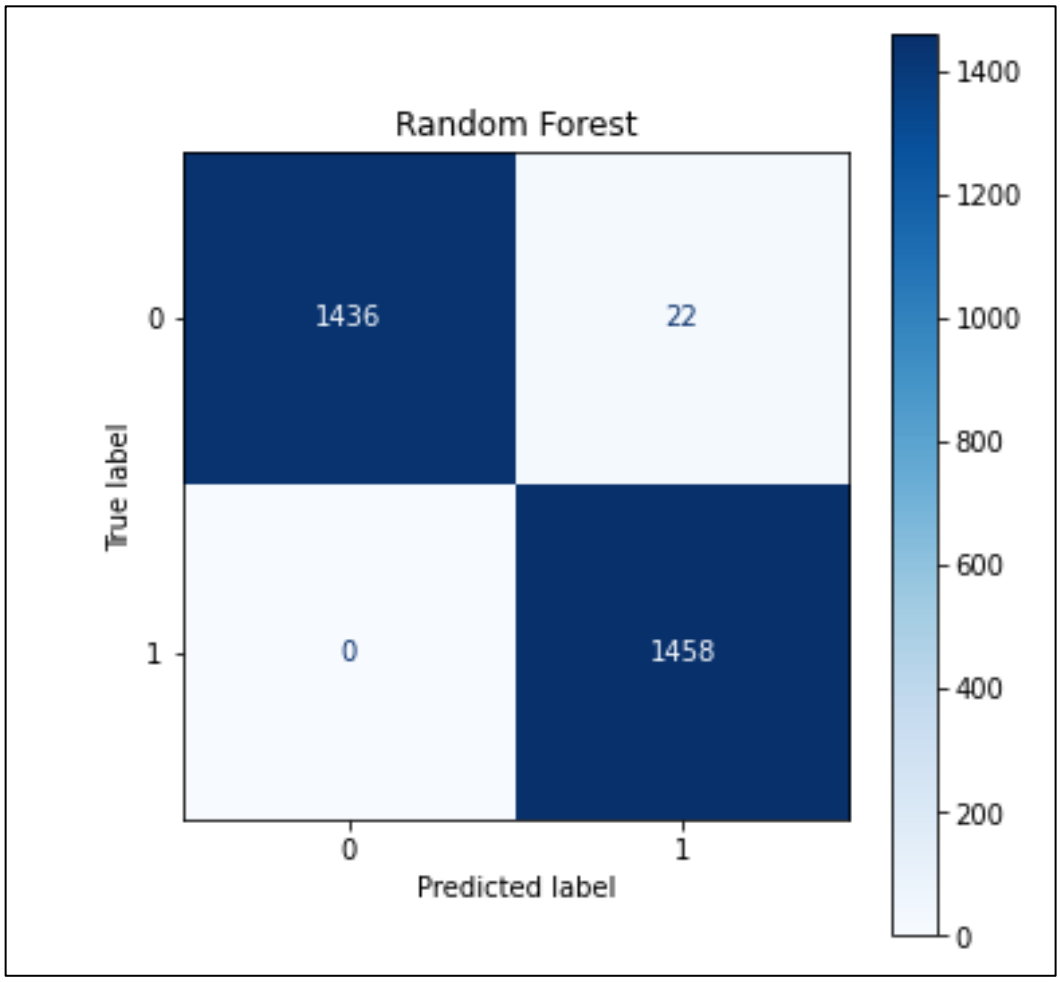


Рисунок 3.13 – Конфузійна матриця Random Forest Classifier

Були представлені результати тестування та тренування моделі. Ці результати вказують на те, що модель має дуже високу точність класифікації, але її точність знижується при роботі з тестовими даними (рис. 3.14).

```
Model: Random Forest
Training Accuracy: 0.9434156378600823
Testing Accuracy: 0.9266117969821673
Roc Auc Score: 0.988637026499649
```

Рисунок 3.14 – Результати тестової та тренованої моделі

На рисунку 3.15 зображена конфузійна матриця для моделі AdaBoost. Ця матриця допомагає оцінити ефективність моделі класифікації.

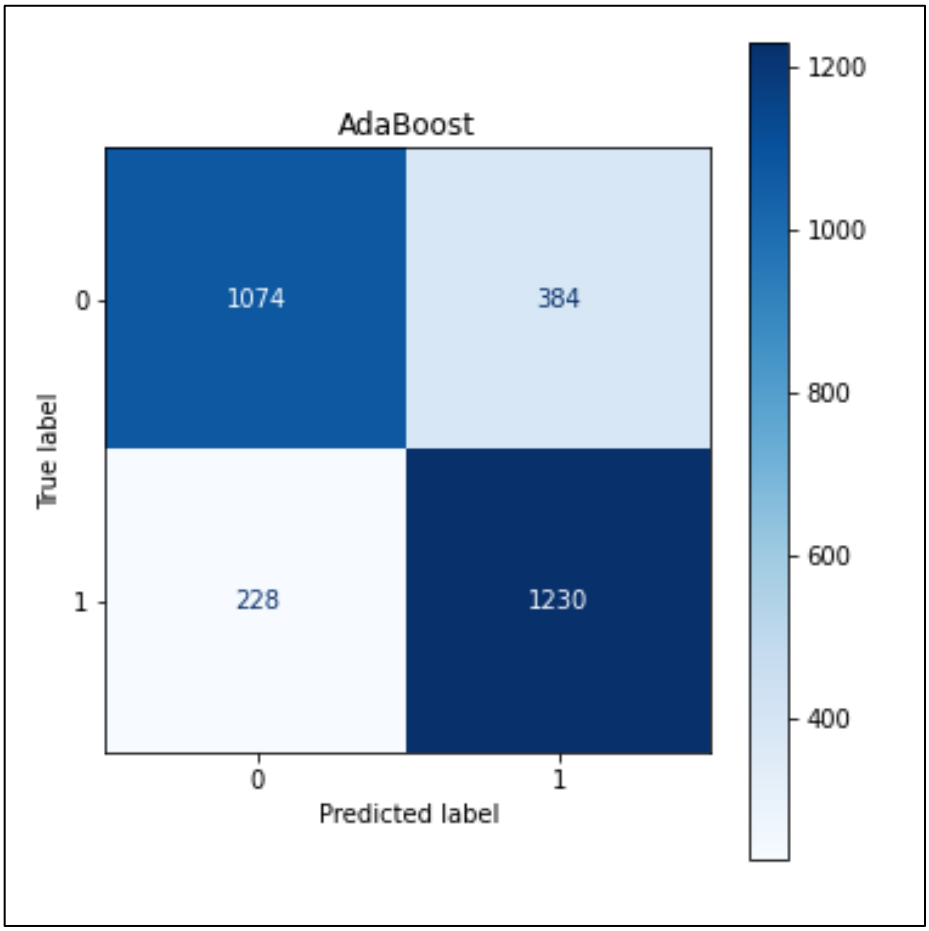


Рисунок 3.15 – Конфузійна матриця Ada Boost

Було відображено тестові та треновані результати моделі. Ці результати вказують на те, що модель має високу точність класифікації. (рис. 3.16).

```
Model: AdaBoost
Training Accuracy: 0.7970311581422692
Testing Accuracy: 0.7901234567901234
Roc Auc Score: 0.876449831684044
```

Рисунок 3.16 – Результати тестової та тренованої моделі

На рисунку 3.17 зображена матриця для моделі Support Vector Machine (SVM). Матриця допомагає оцінити ефективність моделі класифікації.

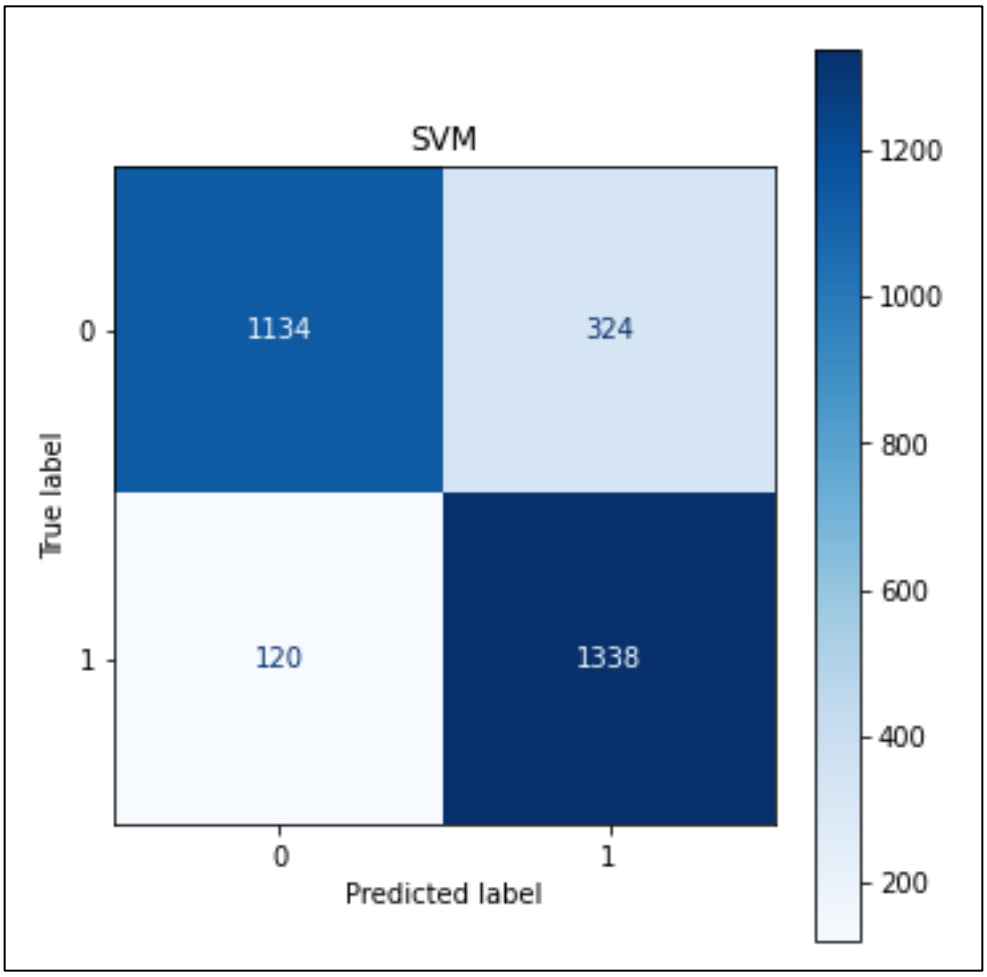


Рисунок 3.17 – Конфузійна матриця SVM

На рисунку 3.18 можна побачити, що точність моделі SVM становить 0.831. Це високий рівень точності, що свідчить про те, що модель є ефективною для прогнозування інсульту.

```
Model: SVM
Training Accuracy: 0.8495002939447384
Testing Accuracy: 0.8319615912208504
Roc Auc Score: 0.9091347863638674
```

Рисунок 3.18 – Результати тестової та тренованої моделі

На рисунку 3.19 зображена конфузійна матриця для моделі XGBoost.

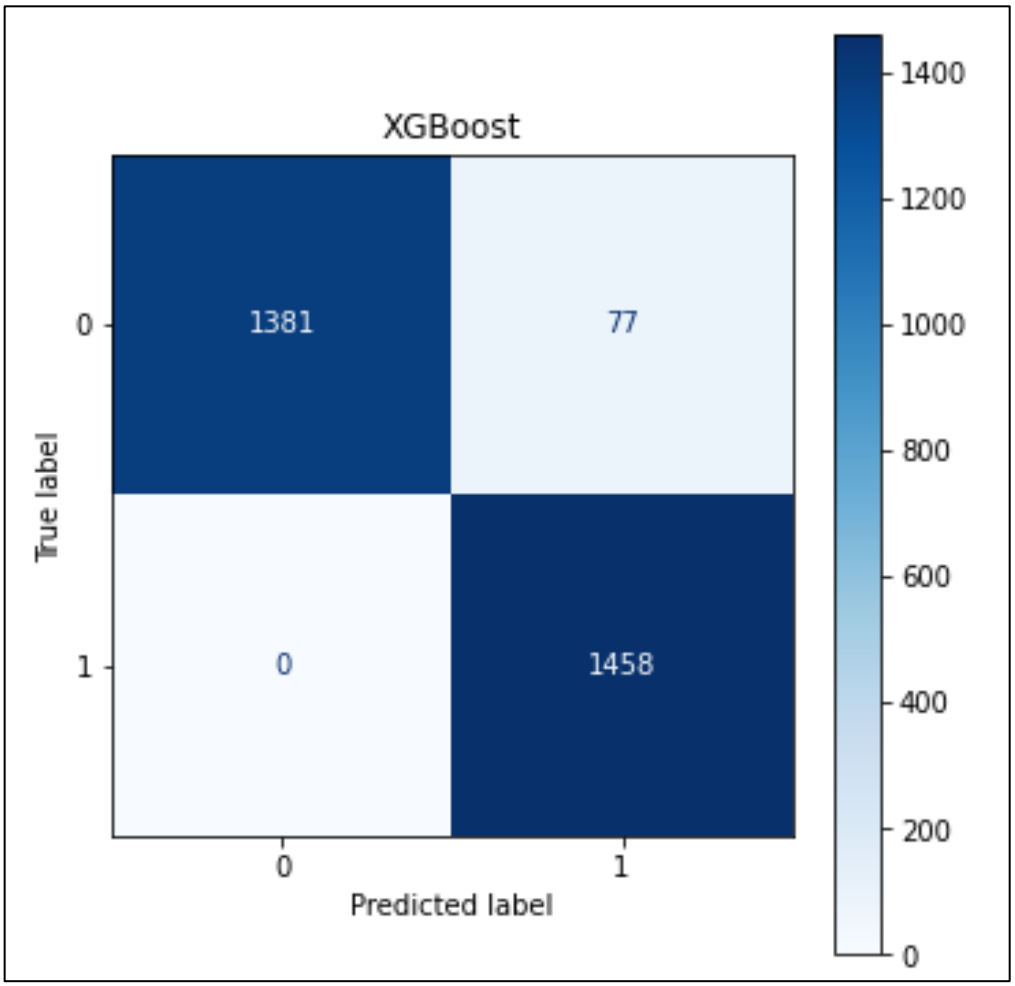


Рисунок 3.19 – Конфузійна матриця XG Boost

На рисунку 3.20 можна зробити висновок, що класифікатор, який використовується для прогнозування інсульту, має високу точність. Це високий рівень точності, що свідчить про те, що модель є ефективною для прогнозування.

```
Model: XGBoost
Training Accuracy: 0.9976484420928865
Testing Accuracy: 0.9735939643347051
Roc Auc Score: 0.9982241678756438
```

Рисунок 3.20 – Результати тестової та тренованої моделі

На рисунку 3.21 зображена матриця для моделі Gradient Boosting. Матриця допомагає оцінити ефективність моделі класифікації.

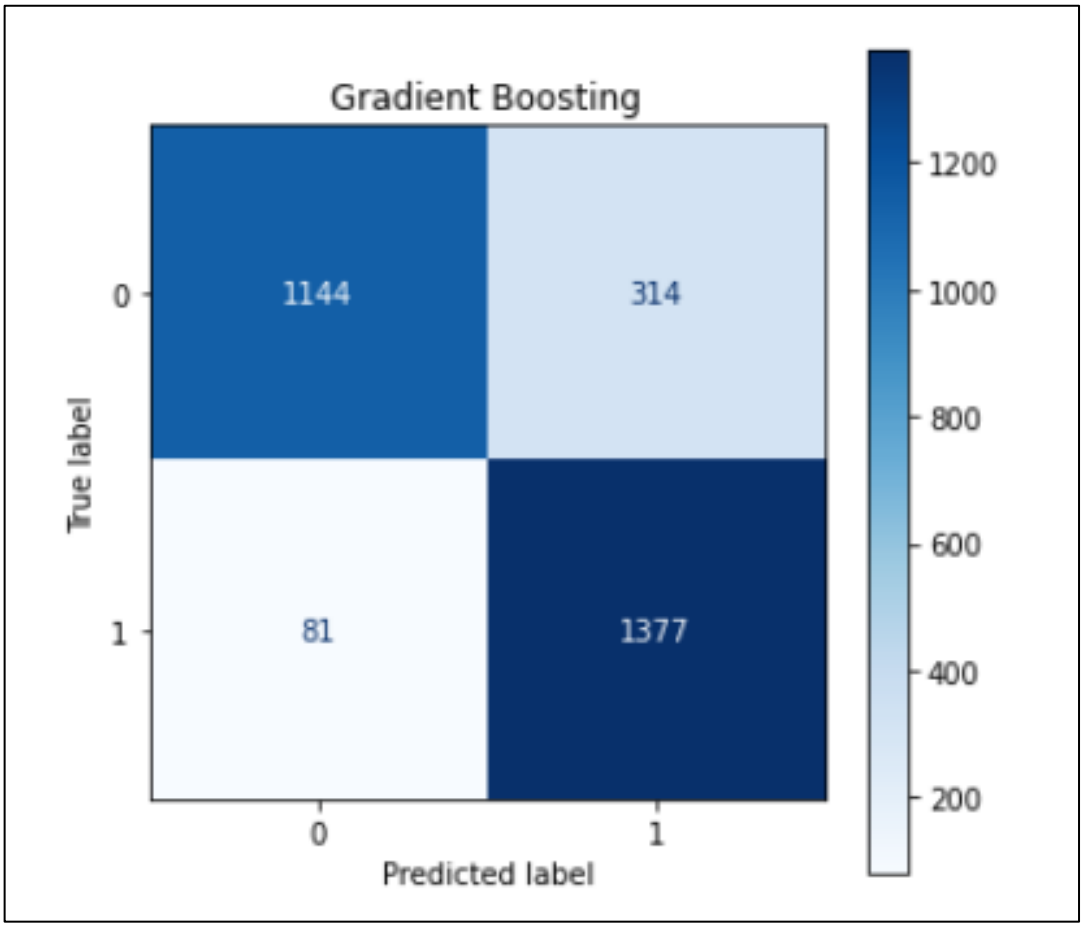


Рисунок 3.21 – Конфузійна матриця Gradient Boosting

На рисунку 3.22 можна побачити, що точність моделі Gradient Boosting становить 0.864. Це високий рівень точності, що свідчить про те, що модель є ефективною для прогнозування інсульту.

```
Model: Gradient Boosting
Training Accuracy: 0.8684597295708407
Testing Accuracy: 0.8645404663923183
Roc Auc Score: 0.9292640199006098
```

Рисунок 3.22 – Результати тестової та тренованої моделі

Було побудовано таблицю та діаграму точності усіх моделей класифікації на тестовому наборі даних. Він показує точність різних моделей

класифікації на тестовому наборі. Тестовий набір складається з набору даних, який не використовувався для навчання моделей класифікації. Цей набір даних використовується для оцінки точності моделей після їх навчання (рис. 3.23-3.24).

	Model	Training Accuracy	Testing Accuracy
4	XGBoost	0.998383	0.972908
0	KNN	0.969283	0.944787
3	Random Forest	0.959289	0.943416
6	KNearest	0.943269	0.920096
2	Decision Tree	0.938419	0.919410
8	Gradient Boosting	0.868460	0.864540
1	SVM	0.851852	0.844307
7	AdaBoost	0.801881	0.803498
5	Logistic Regression	0.767343	0.785665

Рисунок 3.23 – Таблиця з результатами кожної моделі

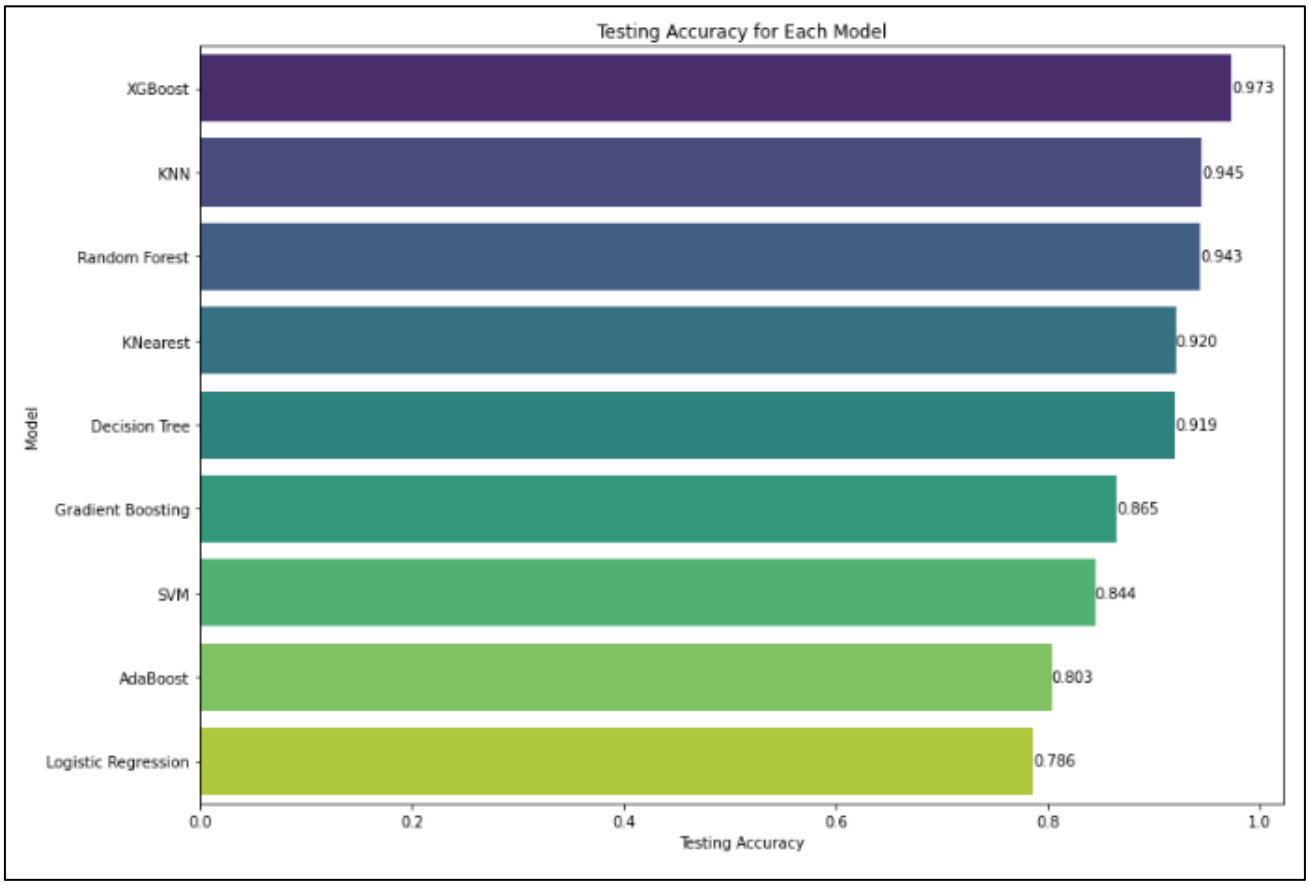


Рисунок 3.24 – Діаграма порівняння результатів кожної моделі.

Діаграма показує, що точність моделей класифікації варіюється залежно від моделі. Однак всі представлені моделі показали хорошу точність на тестовому наборі.

Як можна бачити, результат вийшов точнішим, тому використання SMOTE в цьому випадку підвищило ефективність класифікації. Найкраще себе показала модель XGBoost, оскільки її результати за метрикою accuracy_score 0.974.

Найменшу точність за метрикою accuracy_score 0.78 показала модель Logistic Regression.

3.3 Висновки

В цьому розділі була виконана розробка математичних моделей, виконана реалізація моделей для аналізу даних та розроблено інформаційну

технологію передбачення хворих на інсульт за допомогою таких моделей машинного навчання як: Gradient Boosting, Random Forest, Ada Boost, Logistic Regression, та SVM. За результатами аналізу можна визначити, що найвищу точність (0,974) за метрикою accuracy_score продемонструвала модель XGBoost.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Інформаційної технології передбачення хворих на інсульт» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія передбачення хворих на інсульт» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1 [31].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в

Продовження таблиці 4.1

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно

Продовження таблиці 4.1

Продовження Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій
12	Необхідна розробка та отримання дозволів на виробництво та продаж продукції.	Необхідно Отримання дозвільних документів для виробництва та продажу продукту вимагає значних витрат коштів і часу.	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	4	4
2. Ринкові переваги (наявність аналогів)	3	2	3
3. Ринкові переваги (ціна продукту)	3	4	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	2	2	2
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	2	2	2
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	34	36	35

Продовження таблиці 4.2

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
Середньоарифметична сума балів $СБ_c$	35,0		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [31].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ ,розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія передбачення хворих на інсульт» становить 35,0 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [32]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i , \tag{4.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *i-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *i-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{нi}}{I_{ai}} , \tag{4.2}$$

де $I_{нi}$ та $I_{на}$ – чисельні значення конкретного *i*-го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{нi}} ; \tag{4.3}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість використаних моделей машинного навчання	одиниць	4	8	2	0,2
Швидкість попередньої обробки та очистки даних	бал	6	9	1,5	0,25
Точність прогнозу	%	78	95	1,22	0,15
Кількість графіків розвідувального аналізу	одиниць	9	20	2,11	0,2
Кількість алгоритмів нормалізації даних	одиниць	1	2	2	0,2

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 2 \cdot 0,2 + 1,5 \cdot 0,25 + 1,22 \cdot 0,15 + 2,11 \cdot 0,2 + 2 \cdot 0,2 = 1,78.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,78 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія передбачення хворих на інсульт», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [31]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, (грн);

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 17150{,}00 \cdot 21 / 21 = 17150{,}00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник науково-технічної роботи	17150,00	816,67	21	17150,00
Інженер-аналітик (галузь - системний аналіз)	16540,00	787,62	21	16540,00
Консультант (лікар-невропатолог вищої категорії)	15250,00	726,19	7	5083,33
Всього				38773,33

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія передбачення хворих на інсульт» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=6700,00 (грн);

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду [31];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$$С_1 = 6700{,}00 \cdot 1{,}10 \cdot 1{,}35 / (21 \cdot 8) = 59{,}22 \text{ (грн)}.$$
$$З_{р1} = 59{,}22 \cdot 8{,}00 = 473{,}79 \text{ (грн)}.$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця дослідника-розробника інформаційної технології	8,00	2	1,10	59,22	473,79
Монтаж обчислювального обладнання та серверних блоків	6,00	4	1,50	80,76	484,55
Інсталяція програмного забезпечення розробки (моделювання) інформаційної технології аналізу	6,00	5	1,70	91,53	549,16
Всього					1507,50

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.7)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$$З_{дод} = (38773,33 + 1507,50) \cdot 10 / 100\% = 4028,08 \text{ (грн)}.$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%}, \quad (4.8)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (38773,33 + 1507,50 + 4028,08) · 22 / 100% = 9747,96 (грн).

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія передбачення хворих на інсульт».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (4.9)$$

де $Н_j$ – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = 1,1 … 1,15);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

$М_1$ = 2 · 200,00 · 1,04 - 0 · 0 = 416,00 (грн).

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний (А4)	200,00	2	0	0	416,00
Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір для заміток (А5)	110,00	4	0	0	457,60
Органайзер офісний	220,00	3	0	0	686,40
Начиння канцелярське	199,00	3	0	0	620,88
Картридж для принтера	1050,00	1	0	0	1092,00
Диск оптичний	27,00	4	0	0	112,32
USB-пам'ять	140,00	2	0	0	291,20
Всього					3676,40

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія передбачення хворих на інсульт», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \qquad (4.10)$$

де $Н_j$ – кількість комплектуючих *j*-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих j-го виду, грн;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = 1,1 … 1,15).

$$К_в = 1 \cdot 452,00 \cdot 1,04 = 470,08 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Концентратор Defender SEPTIMA SLIM (83505)	1	452,00	470,08
Найменування комплектуючих	Кількість, шт	Ціна за штуку	Сума, грн
Кабель для передачі даних USB to COM 1.0m Patron (CAB-PN-USB-COM)	1	463,00	481,52
Жорсткий диск 2.5" 2TB Seagate (STGD2000200)	1	3095,00	3218,80
Всього			4170,40

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (4.11)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$К_i$ – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($К_i$ = 1,10…1,12);

k – кількість найменувань устаткування.

$$В_{спец} = 3599,00 \cdot 1 \cdot 1,04 = 3742,96 \text{ (грн).}$$

Отримані результати зведемо до таблиці 4.9:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Маршрутизатор TP-LINK Archer AX55	1	3599,00	3742,96
Моноблок HP 205 G8 Starry White (6D455EA)	1	39460,00	41038,40
Серверне обладнання обробки (Комп'ютер Artline Gaming X66 v22 (X66v22) AMD Ryzen 5 5600X / RAM 16ГБ / HDD 2ТБ + SSD 480ГБ / nVidia GeForce RTX 3060 Ti 8ГБ)	1	35499,00	36918,96
Всього			81700,32

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.12)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$К_i$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($К_i$ = 1,10…1,12);

k – кількість найменувань програмних засобів.

$$В_{прг} = 2350,00 \cdot 1 \cdot 1,04 = 2444,00 \text{ (грн)}.$$

Отримані результати зведемо до таблиці:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Stroke Prediction dataset	1	2350,00	2444,00
Доступ до мережі Internet (високошвидкісний) грн/місяць	2	235,00	488,80
Платформа Kaggle	1	4350,00	4524,00
Прикладне програмне забезпечення (Мова програмування Python та її бібліотеки)	1	7850,00	8164,00
Всього			15620,80

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.13)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (26950{,}00 \cdot 2) / (4 \cdot 12) = 1122{,}92 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Мережеве сховище	26950,00	4	2	1122,92

Продовження таблиці 4.11

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Дослідницька лабораторія	430000,00	25	2	2866,67
Місце оператора спеціалізоване	8150,00	5	2	271,67
Офісна оргтехніка Samsung TM	10200,00	4	2	425,00
Пристрої виведення інформації	6699,00	5	2	223,30
Програмне забезпечення Microsoft Windows, Offise 2021	9280,00	2	2	773,33
Програмно-обчислювальний комплекс розробки системи аналізу даних ARTLINE Gaming X57WHITE v41) (X57WHITEv41) Intel Core i5-12400F / RAM 16ГБ / HDD 2ТБ + SSD 480ГБ / nVidia GeForce RTX 3060 Ti 8 ГБ	39999,00	2	2	3333,25
Всього				9016,13

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{впі}}{\eta_i}, \quad (4.14)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e = 7{,}50$ грн;

$K_{впі}$ – коефіцієнт, що враховує використання потужності, $K_{впі} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0{,}08 \cdot 160{,}0 \cdot 7{,}50 \cdot 0{,}95 / 0{,}97 = 96{,}00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Мережеве сховище Synology 4BAY DS923+	0,08	160,0	96,00
Офісна оргтехніка Samsung TM	0,52	2,2	8,58
Пристрої виведення інформації	0,30	3,2	7,20
Програмно-обчислювальний комплекс розробки системи аналізу даних ARTLINE Gaming X57WHITE v41) (X57WHITEv41) Intel Core i5-12400F / RAM 16ГБ / HDD 2ТБ + SSD 480ГБ / nVidia GeForce RTX 3060 Ti 8 ГБ	0,36	160,0	432,00
Маршрутизатор TP-LINK Archer AX55	0,02	160,0	24,00
Моноблок HP 205 G8 Starry White (6D455EA)	0,20	160,0	240,00

Продовження таблиці 4.12

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Серверне обладнання обробки (Комп'ютер Artline Gaming X66 v22 (X66v22) AMD Ryzen 5 5600X / RAM 16ГБ / HDD 2ТБ + SSD 480ГБ / nVidia GeForce RTX 3060 Ti 8ГБ)	0,40	160,0	480,00
Всього			1287,78

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія передбачення хворих на інсульт» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов’язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з’їзди, конференції, наради, пов’язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{H_{св}}{100\%}, \quad (4.15)$$

де $H_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$$В_{св} = (38773,33 + 1507,50) \cdot 20 / 100\% = 8056,17 \text{ (грн)}.$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (4.16)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 30%.

$В_{сп}$ = (38773,33 + 1507,50) · 30 / 100% = 12084,25 (грн).

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.17)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 55%.

$І_в$ = (38773,33 + 1507,50) · 55 / 100% = 22154,46 (грн).

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків;

витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{H_{нзв}}{100\%}, \quad (4.18)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$В_{нзв}$ = (38773,33 + 1507,50) · 100 / 100% = 40280,83 (грн).

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія передбачення хворих на інсульт» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + M + K_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + I_в + B_{нзв}. \quad (4.19)$$

$В_{заг}$= 38773,33 +1507,50 +4028,08 +9747,96 +3676,40 +4170,40 +81700,32 + 15620,80 + 9016,13 +1287,78 +8056,17 +12084,25 +22154,46 +40280,83 = 252104,42 (грн).

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.20)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,95.

ЗВ = 252104,42 / 0,95 = 265373,07 (грн).

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія передбачення хворих на інсульт» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

На таблиці 4.13 зображно збільшення споживачів.

Таблиця 4.13 – Збільшення споживачів.

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	1100	1600	1000	800

N– кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 21000 осіб;

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 720,00 грн;

$\pm\Delta Ц_o$ – зміна вартості послуги від впровадження результатів, приймемо 161,60 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [31]:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.21)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =40%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1$ =(161,60·21000,00+881,60·1100)·0,83·0,4·(1-0,18/100%)=1187881,13 грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2$ =(161,60·21000,00+881,60·2700)·0,83·0,4·(1-0,18/100%)=1571891,98 грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3$ =(161,60·21000,00+881,60·3700)·0,83·0,4·(1-0,18/100%)=1811898,76 грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4$ =(161,60·21000,00+881,60·4500)·0,83·0,4·(1-0,18/100%)=2003904,19 грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.22)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,15;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 1187881,13/(1+0,15)^1+1571891,98/(1+0,15)^2+1811898,76/(1+0,15)^3+$$
$$+2003904,19/(1+0,15)^4=1032940,11+1188576,17+1191352,85+1145738,73=$$
$$=4558607,85 \text{ (грн).}$$

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \tag{4.23}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2,1;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 265373,07 (грн).

$$PV = k_{інв} \cdot ЗВ = 2,1 \cdot 265373,07 = 557283,45 \text{ (грн).}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{4.24}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 4558607,85 грн;

PV – теперішня вартість початкових інвестицій, 557283,45 (грн).

$$E_{абс} = ПП - PV = 4558607,85 - 557283,45 = 4001324,40 \text{ (грн)}.$$

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.25)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 4001324,40 грн;

PV – теперішня вартість початкових інвестицій, 557283,45 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+4001324,40/557283,45)^{1/4} = 0,69.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (4.26)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,12;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,34.

$\tau_{мін}$= 0,12+0,34 = 0,46 < 0,69 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у

впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія передбачення хворих на інсульт» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{4.27}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,69 = 1,45 року.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія передбачення хворих на інсульт» становить 35,0 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,78 рази.

Також термін окупності становить 1,45 року, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може

спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія передбачення хворих на інсульт».

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробці інформаційної технології передбаченню хворих на інсульт.

Проведено аналіз предметної області, проаналізовано ключові характеристики та переваги різних підходів до машинного навчання, що включають у себе методи класифікації та кластеризації даних, ретельно проаналізовано існуючі аналоги, звертаючи увагу на їхні переваги та недоліки з метою виокремлення прогалин, які потрібно заповнити у цьому дослідженні.

Проведено розвідувальний аналіз, де визначено, яка група людей є найбільш вразливою до інсульту. Також надано характеристику збалансованості датасету, використовуючи алгоритм SMOTE. У результаті проведеного кореляційного аналізу виконано візуалізацію отриманих результатів.

Розроблено моделі класифікації інсульту. За результатами аналізу можна визначити, що найвищу точність (0,974) за метрикою accuracy_score продемонструвала модель XGBoost.

Економічна частина даної роботи містить розрахунок рівня комерційного потенціалу розробку нового програмного продукту, що становить 35 бала. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат.

Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,78 рази. Термін окупності складе становить 1,45 років, що менше 3 років, що свідчить про комерційну привабливість науково-технічної розробки.

За результатами даної роботи опубліковано тези на конференції LIII Всеукраїнській науково-технічній конференції підрозділів ВНТУ: НТК факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2023-2024 рр.), на тему «Інформаційна технологія передбачення хворих на інсульт» [1].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гонтковський Є.Ю., Козачко О.М. Інформаційна технологія передбачення хворих на інсульт. LIII Всеукраїнській науково-технічній конференції підрозділів ВНТУ: НТК факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2023-2024 рр.). Вінниця, 2024. URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/19794/16385 (дата звернення 19.09.2023).

2. Stroke, 2020 URL: https://www.mayoclinic.org/diseases-conditions/stroke/symptoms-causes/syc-20350113 (дата звернення 10.09.2023).

3. Інтелектуальний аналіз даних, 2020 URL: https://ela.kpi.ua/bitstream/123456789/24971/1/Komp_prakt.pdf (дата звернення 11.09.2023).

4. Machine learning 2021 URL: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (дата звернення 11.09.2023).

5. Інтелектуальний аналіз даних 2020 URL: https://ela.kpi.ua/bitstream/123456789/24971/1/Komp_prakt.pdf (дата звернення 11.09.2023).

6. Python Introduction URL: https://www.w3schools.com/python/python_intro.asp (дата звернення 11.09.2023).

7. Logistic Regression 2020 URL: https://towardsdatascience.com/logistic-regression-detailed-overview (дата звернення 11.09.2023).

8. KNearest Neighbors 2018 URL: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/ (дата звернення 11.09.2023)

9. Decision Tree Classification Algorithm 2020 URL: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (дата звернення 11.09.2023)

10. Random Forest Classification with Scikit-Learn 2020 URL: https://www.datacamp.com/tutorial/random-forests-classifier-python (дата звернення 11.09.2023)

11. AdaBoost Algorithm: Understand, Implement and Master AdaBoost 2021 URL: https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/ (дата звернення 11.09.2023)

12. Support Vector Machine (SVM) Algorithm 2020 URL: https://www.geeksforgeeks.org/support-vector-machine-algorithm/ (дата звернення 11.09.2023)

13. XGBoost 2020 URL: https://www.geeksforgeeks.org/xgboost/ (дата звернення 11.09.2023)

14. Gradient Boosting in ML 2020 URL: https://www.geeksforgeeks.org/ml-gradient-boosting/ (дата звернення 11.09.2023)

15. Strokes Dataset - EDA & Prediction - 84% Recall 2021 URL: https://www.kaggle.com/code/navkanthjyothi/strokes-dataset-eda-prediction-84-recall (дата звернення 23.09.2023)

16. Brain Stroke Analysis & Accuracy 96.03% 2021 URL: https://www.kaggle.com/code/hasibalmuzdadid/brain-stroke-analysis-accuracy-96-03 (дата звернення 23.09.2023)

17. Decision Tree Classifier | F1-score 91% 2021 URL: https://www.kaggle.com/code/kabirnagpal/decision-tree-classifier-f1-score-91 (дата звернення 23.09.2023)

18. Kaggle 2023 URL: https://www.kaggle.com/ (дата звернення 23.09.2023)

19. What is Python? 2020. URL: https://opensource.com/resources/python (дата звернення 23.09.2023)

20. Stroke Prediction Dataset 2021. URL: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset (дата звернення 23.09.2023)

21. NumPy: the absolute basics for beginners, 2023. URL: https://numpy.org/doc/stable/user/absolute_beginners.html (дата звернення 23.09.2023)

22. Pandas - Python Data Analysis Library, 2023. URL: https://pandas.pydata.org/ (дата звернення 23.09.2023)

23. Matplotlib: Visualization with Python, 2012. URL: https://matplotlib.org/ (дата звернення 23.09.2023)

24. Seaborn: Everything you need to know about the Python data visualization tool 2023 URL: https://datascientest.com/en/seaborn-everything-you-need-to-know-about-the-python-data-visualization-tool (дата звернення 23.09.2023)

25. Scikit Learn: Discover the Python library dedicated to Machine Learning, 2023. URL: https://datascientest.com/en/scikit-learn-discover-the-python-library-dedicated-to-machine-learning (дата звернення 23.09.2023)

26. Plotly : 2022. URL: https://plotly.net (дата звернення 23.09.2023)

27. Навчання з підкріпленням URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning (дата звернення 23.09.2023)

28. Класифікація методів навчання на основі внутрішнього логічного шляху засвоєння знань. URL: http://surl.li/celjq (дата звернення 23.09.2023)

29. Характеристика методів навчання на основі внутрішнього логічного шляху засвоєння знань учнів. 2020 URL: https://allref.com.ua/uk/skachaty/Harakteristika_metodiv_navchannya_na_osnovi_vnutrishn-ogo_logichnogo_shlyahu_zasvoyennya_znan-_uchniv (дата звернення 23.09.2023)

30. Machine Learning - Машинне навчання URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning (дата звернення 23.09.2023)

31. SMOTE for Imbalanced Classification with Python 2021 URL: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/ (дата звернення 23.09.2023)

32. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця : ВНТУ, 2021. 42 с.

33. Кавецький В.В. Економічне обґрунтування інноваційних рішень: практикум / за ред. В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с.

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«___» _____ 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ХВОРИХ НА ІНСУЛЬТ»
08-34.МКР.012.00.000.ТЗ

Керівник: к.т.н., доц.
_____ Олексій КОЗАЧКО
«___» _____ 2023 р.

Розробив: студент гр. 2ІСТ-22м
_____ Євгеній ГОНТКОВСЬКИЙ
«___» _____ 2023 р.

Вінниця 2023

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2023 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2023 р.

2. Джерела розробки:

1. Machine learning 2021 URL: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (date of access: 23.09.2023)
2. SMOTE for Imbalanced Classification with Python 2021 URL: http://surl.li/okfur (date of access: 23.09.2023)

3. Мета і призначення роботи:

Метою дослідження є підвищення точності передбачення хворих на інсульт за рахунок використанні методів машинного навчання.

4. Вихідні дані для проведення робіт:

Датасет Kaggle «Stroke Prediction Dataset» з даними для передбачення ймовірності інсульту у пацієнтів.

5.Методи дослідження:

Підготовка даних, розвідувальний аналіз, прогнозування даних;

6. Етапи роботи і терміни їх виконання:

1. Аналіз предметної області _____ – _____
2. Вибір оптимальних технологій та проведення розвідувального аналізу _____ – _____
3. Розроблення моделей класифікації _____ – _____
4. Економічна частина _____ – _____
5. Оформлення матеріалів до захисту МКР. _____ – _____

7. Очікувані результати та порядок реалізації:

Очікуваним результатом є розроблення інформаційної технології передбачення хворих на інсульт та побудова моделей класифікації, які дадуть високу точність передбачення хворих на інсульт.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання магістерських кваліфікаційних робіт для студентів спеціальності 126 «Інформаційні системи та технології» (освітня програма «Інформаційні технології аналізу даних та зображень»).

9. Порядок приймання роботи

Публічний захист «____» _____ 2023 р.

Початок розробки «____» _____ 2023 р.

Граничні терміни виконання МКР «____» _____ 2023 р.

Розробив студент групи 2ІСТ-22м _____ Євгеній ГОНТКОВСЬКИЙ

Додаток Б
Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія передбачення хворих на інсульт»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 90,45% Схожість 9,55%

Аналіз звіту подібності (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Євгеній ГОНТКОВСЬКИЙ

Керівник роботи _____ (підпис) Олексій КОЗАЧКО

Додаток В
Лістинг програми

```
import warnings
warnings.filterwarnings('ignore')

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import sklearn

import matplotlib.pyplot as plt
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import plotly.figure_factory as ff


from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV

from sklearn.metrics import accuracy_score, classification_report,
roc_curve,precision_recall_curve, auc,confusion_matrix
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.svm import SVC
from sklearn.impute import KNNImputer

from xgboost import XGBClassifier

from catboost import CatBoostClassifier

en_df_imputed = en_df
imputer = KNNImputer(n_neighbors=4, weights="uniform")
imputer.fit_transform(en_df_imputed)
```

```
en_df_imputed.isnull().sum()

from imblearn.over_sampling import SMOTE
X , y = en_df_imputed[features],en_df_imputed["stroke"]
x_train,x_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=23)
sm = SMOTE()
X_res, y_res = sm.fit_resample(x_train,y_train)

print("Before OverSampling, counts of label '1': {}".format(sum(y==1)))
print("Before OverSampling, counts of label '0': {} \n".format(sum(y==0)))

print('After OverSampling, the shape of train_X: {}'.format(X_res.shape))
print('After OverSampling, the shape of train_y: {} \n'.format(y_res.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_res==1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_res==0)))

def plot_cm(cm,title):
    z = cm
    x = ['No stroke', 'stroke']
    y = x
    # change each element of z to type string for annotations
    z_text = [[str(y) for y in x] for x in z]

    # set up figure
    fig = ff.create_annotated_heatmap(z, x=x, y=y, annotation_text=z_text,
colorscale='deep')

    # add title
    fig.update_layout(title_text='<i><b>Confusion matrix {}</b></i>'.format(title),
                      #xaxis = dict(title='x'),
                      #yaxis = dict(title='x')
                     )

    # add custom xaxis title
    fig.add_annotation({'font':{'color':"black",'size':14},
                        'x':0.5,
                        'y':-0.10,
                        'showarrow':False,
                        'text':"Predicted value",
                        'xref':"paper",
                        'yref':"paper"})

    fig.add_annotation({'font':{'color':"black",'size':14},
```

```
                    'x':-0.15,
                    'y':0.5,
                    'showarrow':False,
                    'text':"Real value",
                    'textangle':-90,
                    'xref':"paper",
                    'yref':"paper"})


    # adjust margins to make room for yaxis title
    fig.update_layout(margin={'t':50, 'l':20},width=750,height=750)



    # add colorbar
    fig['data'][0]['showscale'] = True
    fig.show()



import plotly.graph_objects as go
import matplotlib.pyplot as plt
import seaborn as sns

def hist_score(score):
    models_names = [
        'Logistic Regression',
        'KNearest Neighbor',
        'Decision Tree Classifier',
        'Random Forest Classifier',
        'Ada Boost',
        'SVM',
        'XG Boost',
      ]

    sns.set_style('darkgrid')

    fig = go.Figure()

    for model, acc in zip(models_names, score):
        fig.add_trace(go.Bar(x=[acc], y=[model], orientation='h', name=model))

    layout_dict = {
        'barmode': 'group',
```

```
        'xaxis': {'title': '% of Accuracy'},
        'yaxis': {'title': 'Classifier Models', 'categoryorder': 'total ascending'},
        'title': 'Accuracy of different Classifier Models on test set',
        'width': 1024,
        'height': 1024
    }
    fig.update_layout(**layout_dict)

    fig.show()

def run_exp_on_feature(x_train, y_train, x_test, y_test):
    models = [
        ['Logistic Regression ', LogisticRegression()],
        ['KNearest Neighbor ', KNeighborsClassifier()],
        ['Decision Tree Classifier ', DecisionTreeClassifier()],
        ['Random Forest Classifier ', RandomForestClassifier()],
        ['Ada Boost ', AdaBoostClassifier()],
        ['SVM ', SVC()],
        ['XG Boost', XGBClassifier()]
    ]

    models_score = []
    for name, model in models:
        model = model
        model.fit(x_train, y_train)
        model_pred = model.predict(x_test)
        cm_model = confusion_matrix(y_test, model_pred)

        # Заміна колірної палітри для heatmap
        plt.figure(figsize=(15, 15))
        sns.heatmap(cm_model, annot=True, fmt='g', cmap='plasma', xticklabels=[0,
1], yticklabels=[0, 1])
        plt.xlabel('Predicted')
        plt.ylabel('True')
        plt.title(f'Confusion Matrix - {name}model')
        plt.show()

        models_score.append(accuracy_score(y_test, model.predict(x_test)))

        print(name)
        print('Validation Accuracy: ', accuracy_score(y_test, model.predict(x_test)))
        print('Training Accuracy: ', accuracy_score(y_train, model.predict(x_train)))
        print('##################################################')
```

```
        fpr, tpr, thresholds = roc_curve(y_test, model_pred)

        fig = px.area(
            x=fpr, y=tpr,
            title=f'ROC Curve (AUC={auc(fpr, tpr):.4f})',
            labels={'x': 'False Positive Rate', 'y': 'True Positive Rate'},
            width=700, height=500
        )
        fig.add_shape(
            type='line', line={'dash': 'dash'},
            x0=0, x1=1, y0=0, y1=1
        )

        fig.update_yaxes(scaleanchor="x", scaleratio=1)
        fig.update_xaxes(constrain='domain')
        fig.show()

    return models_score

models_score = run_exp_on_feature(x_train,y_train,x_test,y_test)
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ХВОРИХ НА ІНСУЛЬТ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2023 р.

Вінниця 2023

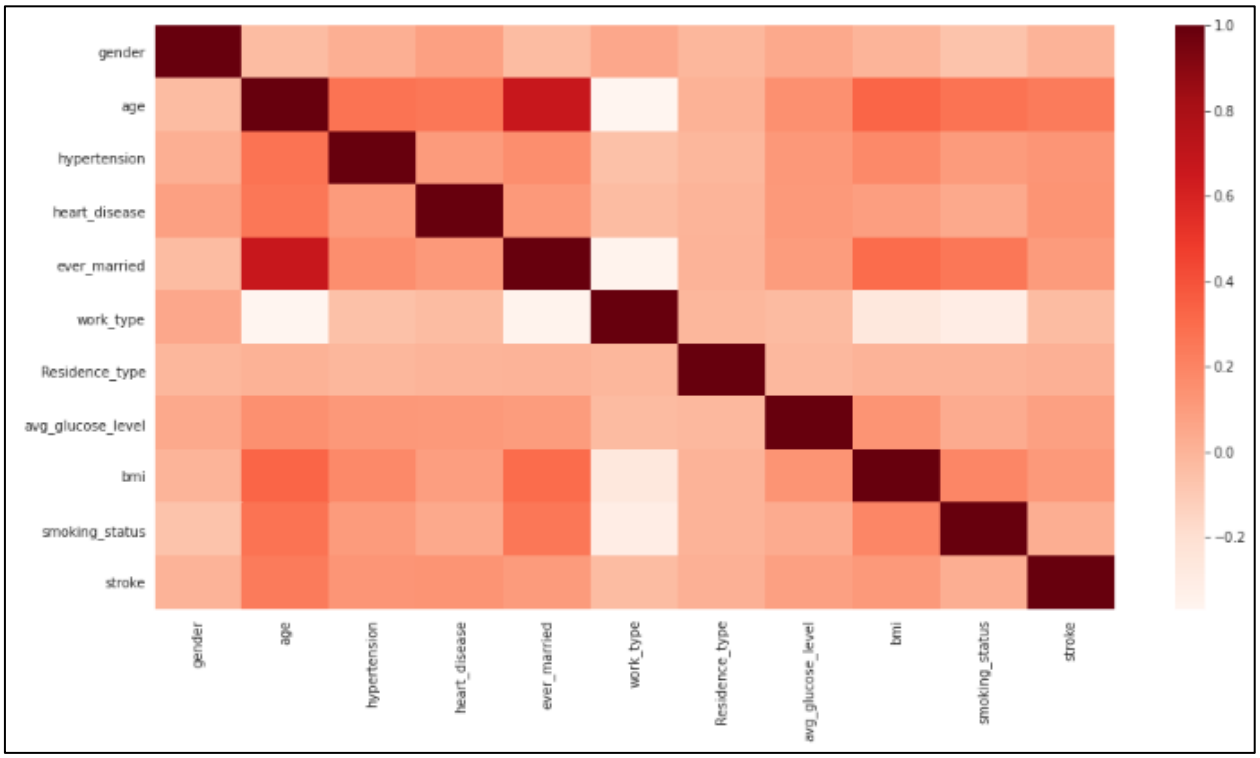


Рисунок Г.1 – Кореляційна матриця показників.

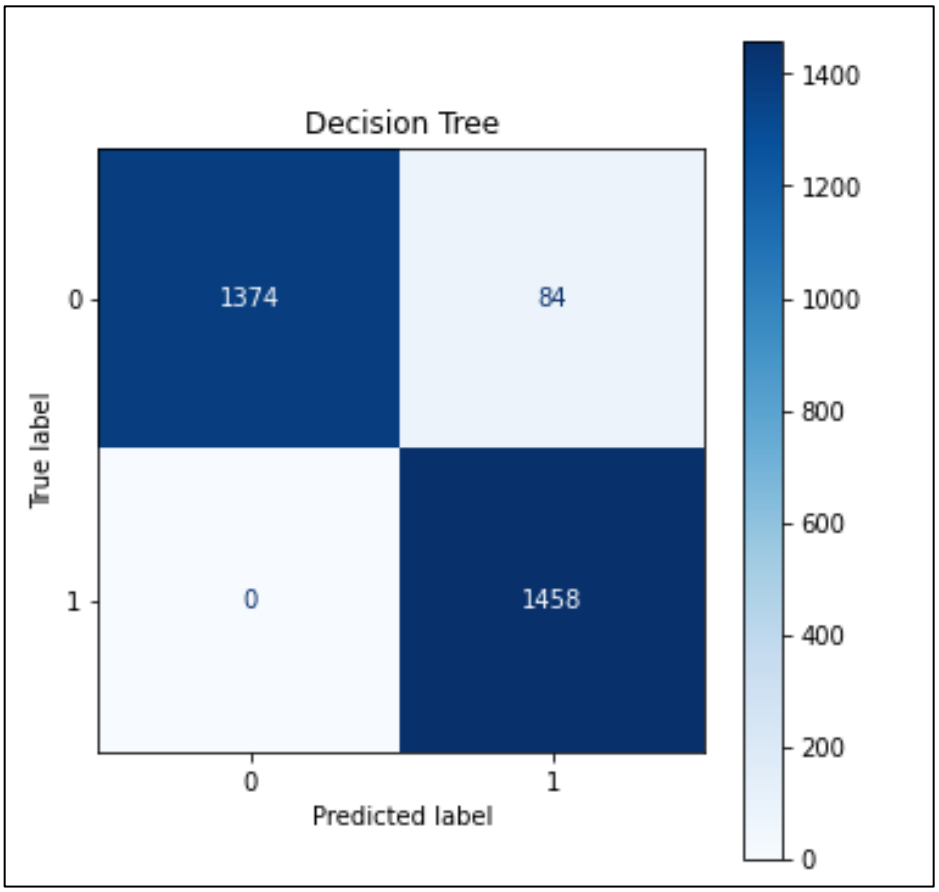


Рисунок Г.2 – Конфузійна матриця Decision Tree

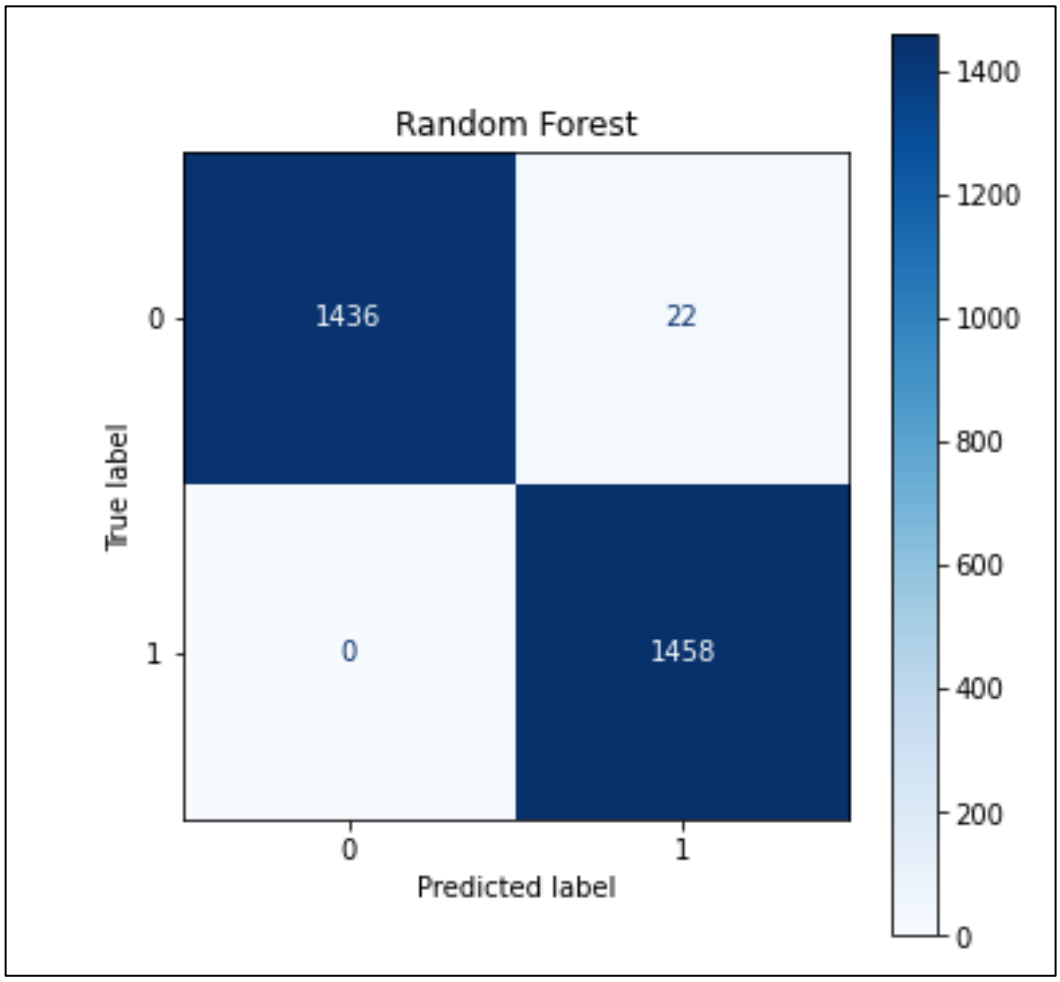


Рисунок Г.3 – Конфузійна матриця Random Forest

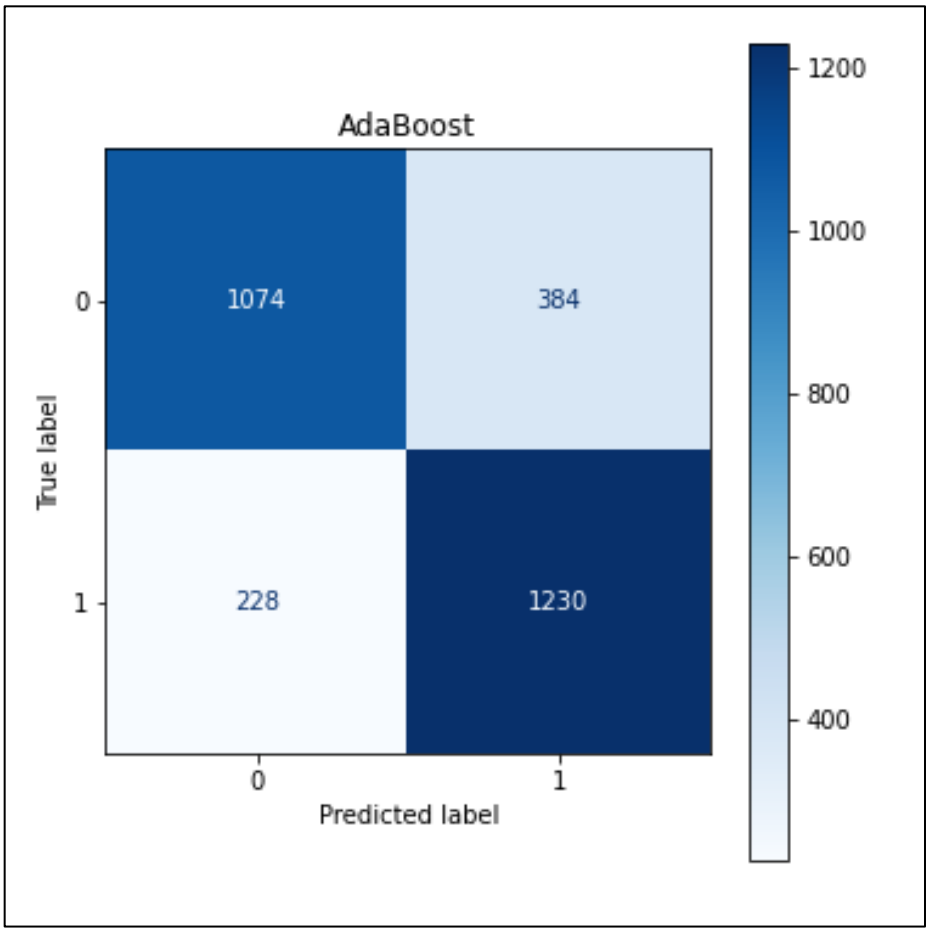


Рисунок Г.4 – Конфузійна матриця Ada Boost

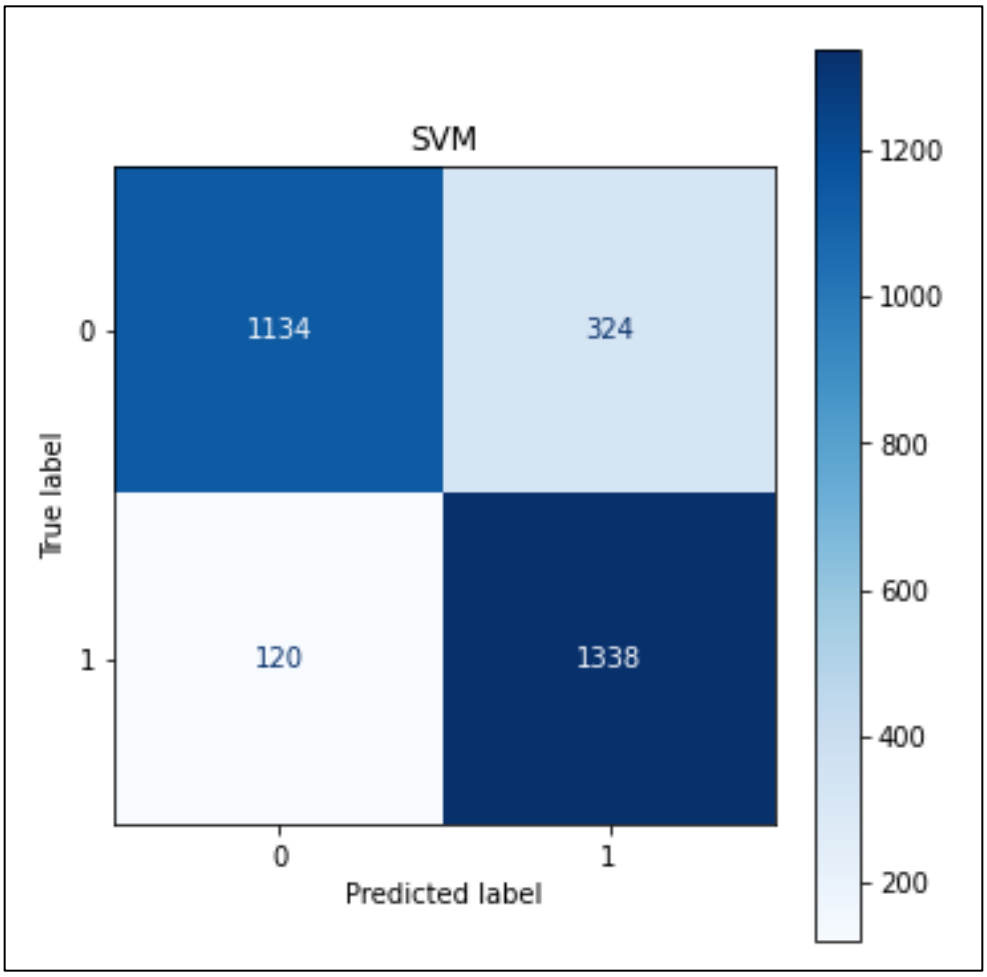


Рисунок Г.5 – Конфузійна матриця SVM

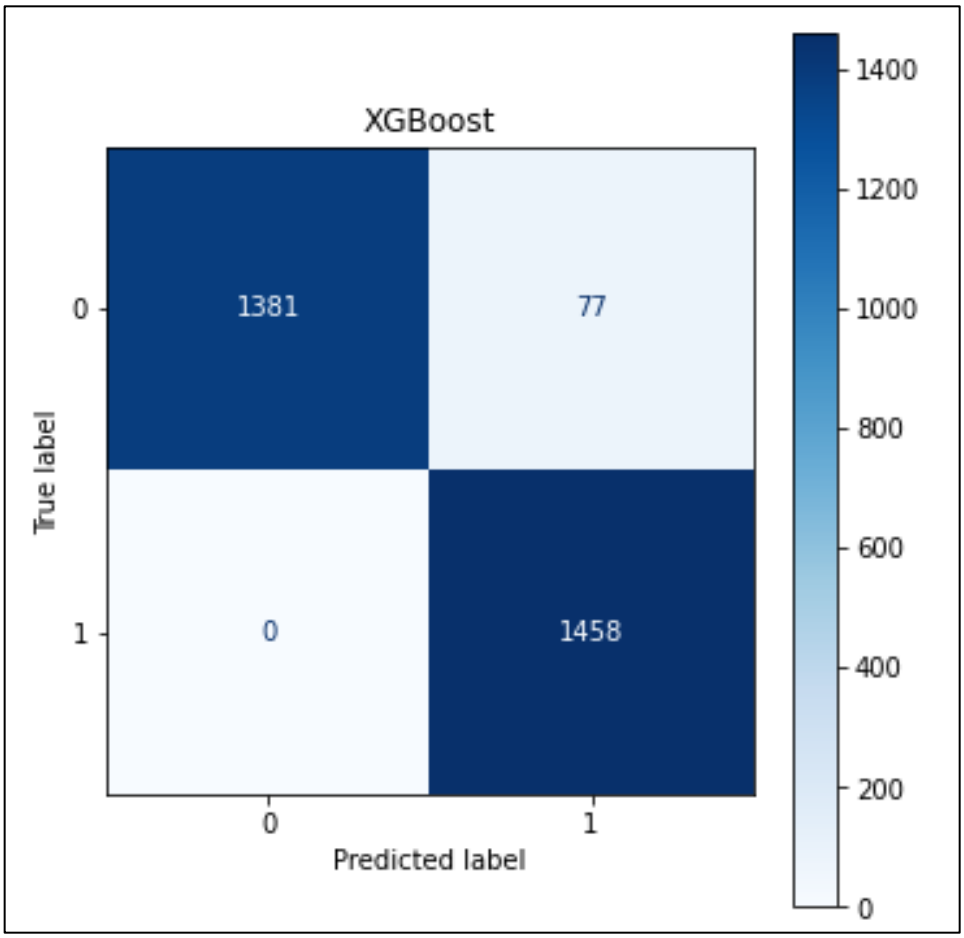


Рисунок Г.6 – Конфузійна матриця XG Boost

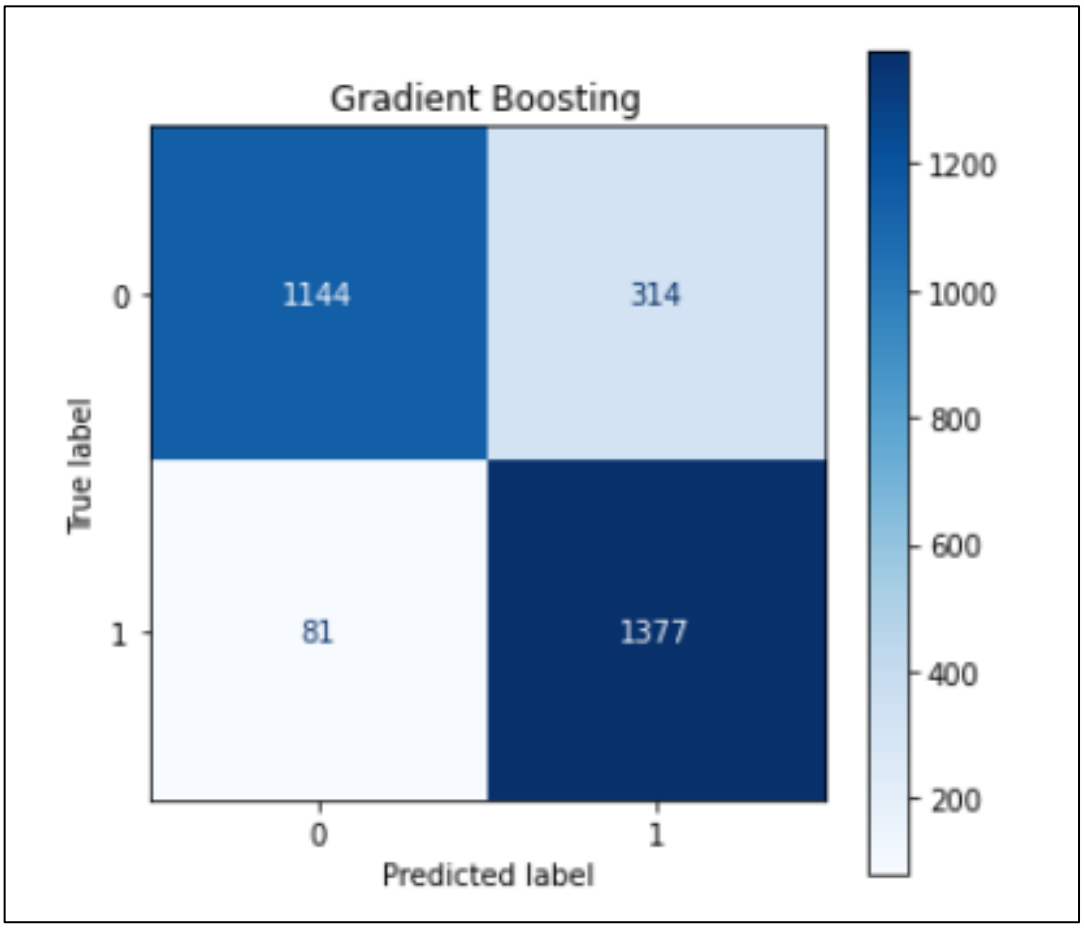


Рисунок Г.7 – Конфузійна матриця Gradient Boosting

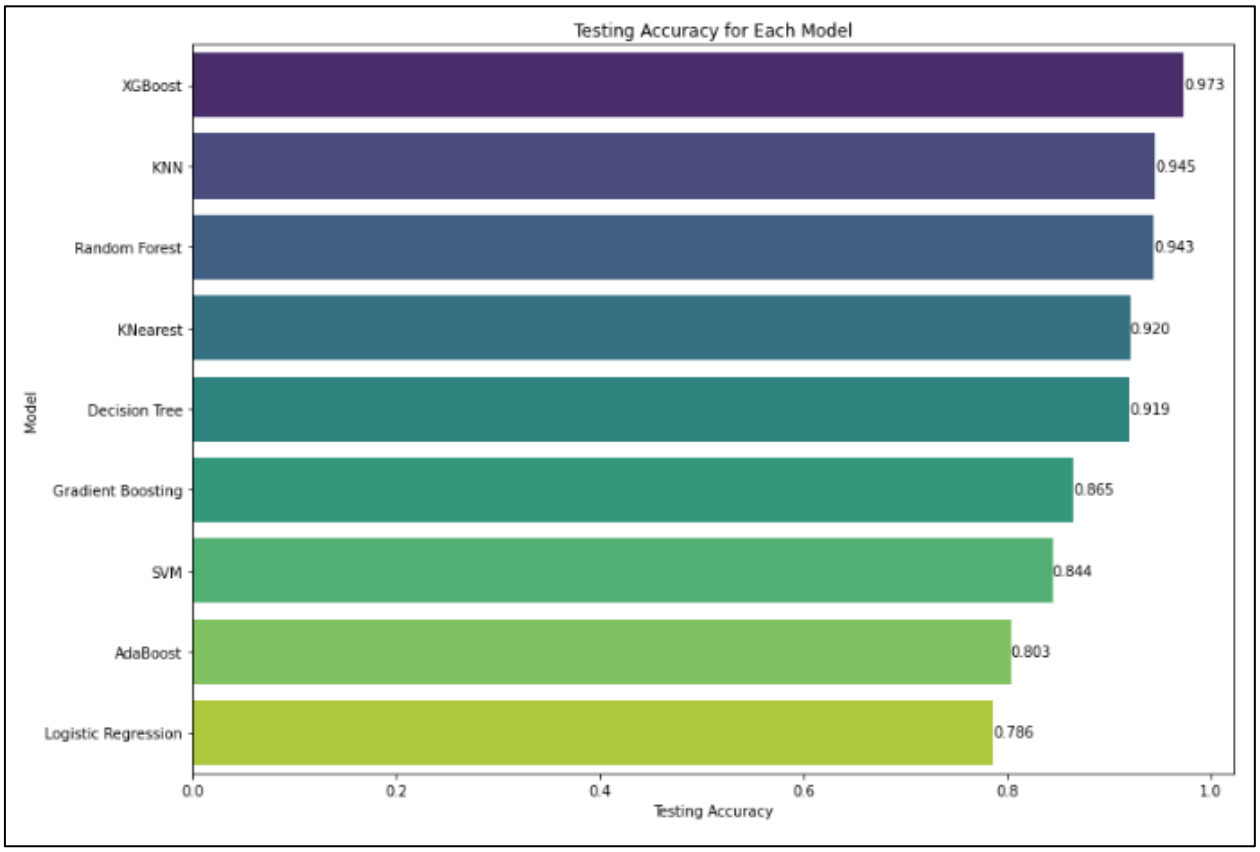


Рисунок Г.8 – Діаграма порівняння результатів кожної моделі.