МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

КРАВЧУК Богдан Олександрович

Модуль інтелектуального передбачення серцево-судинних захворювань / Module for intelligent prediction of cardiovascular diseases

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-41
Б. О. Кравчук

Науковий керівник:
викладач В.І. Дорош

Дипломний проект допущено до захисту
«___»_____ 2023 р.

Завідувач кафедри
_____ М.П. Комар

Тернопіль – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«____» _____2022р.

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Кравчуку Богдану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема проекту: Модуль інтелектуального передбачення серцево-судинних захворювань / Module for intelligent prediction of cardiovascular diseases
керівник проекту викладач В.І. Дорош

затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- вивчити використання штучного інтелекту для передбачення серцево-судинних захворювань.
- розглянути традиційні підходи до передбачення ризику серцево-судинних захворювань.
- описати підхід до прогнозування ризиків за допомогою штучного інтелекту.
- розробити алгоритм дослідницького аналізу даних інтелектуального передбачення серцево-судинних захворювань.
- охарактеризувати методи класифікації та алгоритми інтелектуального передбачення.
- проаналізувати та описати набір даних для дослідження.
- здійснити навчання даних, оцінити отримані результати та визначити кращу модель для інтелектуального передбачення серцево-судинних захворювань..

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- Алгоритм дослідницького аналізу даних інтелектуального передбачення серцево-судинних захворювань

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	викладач В.І. Дорош		

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області і постановка задачі дослідження	30.12.2022	
2	Алгоритмічне та інформаційне забезпечення	24.03.2023	
3	Програмно-технологічне забезпечення	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент _____ Б. О. Кравчук
(підпис)

Керівник проекту_____ В.І. Дорош
(підпис)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 82 с., 22 рис., 3 табл., 2 додатки, 67 джерел.

Метою дипломний проект є розробка модуля інтелектуального передбачення серцево-судинних захворювань на основі машинного навчання.

Об'єктом дослідження є процеси діагностики та прогнозування серцево-судинних захворювань у медичній сфері.

Предметом дослідження є методи, алгоритми та моделі штучного інтелекту та машинного навчання, які застосовуються для розробки модуля інтелектуального передбачення серцево-судинних захворювань.

Розроблено та досліджено програмне забезпечення для ефективного модуля прогнозування ССЗ, який сприятиме ранньому виявленню та запобіганню цим захворюванням, забезпечуючи покращення охорони здоров'я.

Розробка модуля інтелектуального передбачення серцево-судинних захворювань на основі машинного навчання має велику актуальність у зв'язку зі значним поширенням цих захворювань та потребою у точній діагностиці та прогнозуванні. Використання штучного інтелекту та машинного навчання дозволяє аналізувати великі обсяги медичних даних, встановлювати закономірності та розробляти прогностичні моделі. Результати дослідження сприятимуть покращенню охорони здоров'я, ранньому виявленню та запобіганню серцево-судинних захворювань.

СЕРЦЕВО-СУДИННІ ЗАХВОРЮВАННЯ, МЕДИЧНА ДІАГНОСТИКА, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ.

ABSTRACT

The bachelor’s thesis report: 82 pages, 22 figures, 3 tables, 2 appendices, 67 references.

The aim of the diploma project is to develop an intelligent prediction module for cardiovascular diseases based on machine learning.

The object of research is the processes of diagnosis and prediction of cardiovascular diseases in the medical field.

The subject of research includes methods, algorithms, and models of artificial intelligence and machine learning applied in the development of the intelligent prediction module for cardiovascular diseases.

Software has been developed and investigated for an efficient prediction module for cardiovascular diseases, which will contribute to early detection and prevention of these diseases, thereby improving healthcare.

The development of an intelligent prediction module for cardiovascular diseases based on machine learning is highly relevant due to the significant prevalence of these diseases and the need for accurate diagnosis and forecasting. The use of artificial intelligence and machine learning allows for the analysis of large volumes of medical data, identification of patterns, and development of predictive models. The results of the research will contribute to improved healthcare, early detection, and prevention of cardiovascular diseases.

CARDIOVASCULAR DISEASES, MEDICAL DIAGNOSIS, MACHINE LEARNING, PREDICTION.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1Модуль інтелектуального передбачення серцево-судинних захворювань.

1.2Область застосування – охорона здоров'я та медична діагностика.

2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп’ютерних інформаційних технологій Західноукраїнського національного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломний проект є розробка модуля інтелектуального передбачення серцево-судинних захворювань на основі машинного навчання.

4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

– вибір даних (пакетних даних / даних у реальному часі) з різних архітектур великих даних;

– провести попередню обробку даних для інтелектуального аналізу тексту.

– навчання даних та оцінка отриманих результатів

– визначення кращої моделі.

5.2 Вимоги до апаратних засобів:

– кластер запускається на фізичній машині Macbook Pro з 16 гігабайт оперативної пам’яті та процесором 2.3 GHz Intel Core i5.

5.3 Вимоги до програмних засобів:

– для розробки програмне забезпечення - Python 3.7;

– для створення графічного інтерфейсу користувача використано – tkinter, Adobe XD;

– для реалізації моделей навчання – фреймворк sklearn.

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ Б. О. Кравчук
(підпис) (прізвище та ініціали)

Керівник дипломного проекту_____ В.І. Дорош
(підпис) (прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- ССЗ: Серцево-судинні захворювання - медичний термін, який описує групу захворювань, пов'язаних зі здоров'ям серця і судин.
- ШІ: Штучний інтелект - галузь науки та технологій, що вивчає імітацію інтелекту людей в комп'ютерних системах для вирішення завдань, які зазвичай вимагають людського розуму.
- ML: Machine Learning - галузь штучного інтелекту, яка займається розробкою алгоритмів та моделей, що дозволяють комп'ютерам самостійно вчитися і покращувати свою продуктивність на основі даних.
- LR: Логістична регресія - статистична модель, яка використовується для прогнозування категоріальної змінної на основі лінійної комбінації змінних і застосування логістичної функції.
- DL: Глибоке навчання - підгалузь машинного навчання, що використовує нейронні мережі з багатьма шари для автоматичного витягування високорівневих ознак з даних.
- ANN: Artificial Neural Network - штучна нейронна мережа, модель, яка імітує роботу нервової системи людини для обробки і аналізу даних.
- kNN: k-Nearest Neighbors - алгоритм машинного навчання, який використовується для класифікації і регресії, шляхом вибору k найближчих сусідів для прийняття рішення про прогнозування.
- SVM: Support Vector Machines - метод машинного навчання, який використовується для класифікації і регресії, шляхом знаходження границі розділення між класами з найбільшою можливою шириною.
- NB: Наївний Байєс - статистичний алгоритм машинного навчання, який використовує принцип Байєса та припущення про незалежність змінних для класифікації.

- DT: Дерева рішень - модель машинного навчання, що представляє розбиття даних на деревоподібні структури для класифікації і регресії.
- GBM: Gradient Boosting Machine - метод машинного навчання, який побудований на основі бустингу і використовує градієнтний спуск для покращення моделей.
- RF: Random Forest - метод машинного навчання, що використовується для класифікації, регресії та інших завдань, побудований на основі ансамблю дерев рішень.
- AUROC: Area Under the ROC Curve - площа під кривою ROC, яка використовується як метрика для оцінки якості моделей класифікації.
- AdaBoost: Adaptive Boosting - алгоритм бустингу, який навчає послідовні слабкі моделі і вдосконалює їх шляхом надання більшої ваги помилковим прикладам.
- XGBoost: eXtreme Gradient Boosting - вдосконалений алгоритм градієнтного бустингу, який швидко навчається та показує високу продуктивність.
- MLPClassifier: Multi-Layer Perceptron Classifier - модель штучної нейронної мережі з багатьма шарами, що використовується для класифікації.

ВСТУП

Актуальність теми розробки модуля інтелектуального передбачення серцево-судинних захворювань полягає в тому, що серцево-судинні захворювання (ССЗ) є однією з основних причин смерті на глобальному рівні. Розвиток ефективних методів діагностики та прогнозування є важливим для попередження та своєчасного лікування ССЗ.

Штучний інтелект (ШІ) та машинне навчання відкривають нові можливості для аналізу великих наборів медичних даних, виявлення закономірностей та встановлення прогностичних моделей. Застосування таких технологій дозволяє покращити якість медичної допомоги, забезпечити більш точне прогнозування ризиків та індивідуалізацію лікування.

Враховуючи вище зазначене, актуальність розробки модуля інтелектуального передбачення серцево-судинних захворювань полягає у значному внеску в охорону здоров'я населення, зниження смертності від ССЗ та підвищення ефективності та економічної доцільності медичних послуг.

Метою дипломний проект є розробка модуля інтелектуального передбачення серцево-судинних захворювань на основі машинного навчання.

Для досягнення цієї мети було необхідно провести теоретичні дослідження та вирішити послідовно завдання, що включають визначення та аналіз:

1. Вивчити використання штучного інтелекту для передбачення серцево-судинних захворювань.
2. Розглянути традиційні підходи до передбачення ризику серцево-судинних захворювань.
3. Описати підхід до прогнозування ризиків за допомогою штучного інтелекту.
4. Розробити алгоритм дослідницького аналізу даних інтелектуального передбачення серцево-судинних захворювань.
5. Охарактеризувати методи класифікації та алгоритми інтелектуального передбачення.

6. Проаналізувати та описати набір даних для дослідження.

7. Здійснити навчання даних, оцінити отримані результати та визначити кращу модель для інтелектуального передбачення серцево-судинних захворювань.

Об'єктом дослідження є процеси діагностики та прогнозування серцево-судинних захворювань у медичній сфері.

Предметом дослідження є методи, алгоритми та моделі штучного інтелекту та машинного навчання, які застосовуються для розробки модуля інтелектуального передбачення серцево-судинних захворювань.

Методи дослідження. У роботі використані методи математичного програмування, статистичного опису, візуалізації даних та машинного навчання.

Практичне значення одержаних результатів полягає у розробці ефективного модуля прогнозування ССЗ, який сприятиме ранньому виявленню та запобіганню цим захворюванням, забезпечуючи покращення охорони здоров'я..

Структура та обсяг роботи. Дипломний проект має таку структуру: вступ, три розділи, висновки і список використаних джерел. Загальний обсяг роботи становить 82 сторінок комп'ютерного тексту, який містить 22 рисунки та 3 таблиці. Список використаних джерел складається з 67 найменувань і розташований на 6 сторінках.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Використання штучного інтелекту для передбачення серцево-судинних захворювань

Серцево-судинні захворювання (ССЗ) є однією з домінуючих причин смерті в усьому світі, із загалом 17,9 мільйонів смертей у 2016 році, що еквівалентно приблизно 31% усіх смертей [1] . Звідси понад 75% смертей від серцево-судинних захворювань припадає на країни з низьким і середнім рівнем доходу. Крім того, країни Організації економічного співробітництва та розвитку (ОЕСР) повідомили про 4791 смерть на 100 000 населення внаслідок серцевого нападу та ішемічної хвороби серця в 2017 році [2] . Цікаво, що частка, яка сприяє смертності від серцево-судинних захворювань, відрізняється в різних країнах [4] , тенденції зміни рівня смертності від серцево-судинних захворювань пов’язані зі змінами в поширеності факторів ризику та закладах невідкладної допомоги в кожному регіоні за останні роки[3], [4], [5], [6].

Серцево-судинні захворювання представлені гострим коронарним синдромом (ГКС), який є широким терміном, що включає спектр клінічних проявів, включаючи гострий інфаркт міокарда (ГІМ), нестабільну стенокардію та раптову серцеву смерть внаслідок зниження кровотоку до міокарда [7] . Близько 85% смертності від серцево-судинних захворювань спричинено ГІМ та інсультами, оскільки обидва захворювання зазвичай викликані закупоркою, яка перешкоджає надходженню крові до серця чи мозку через накопичення жирових відкладень на внутрішніх стінках кровоносних судин, які постачають кров. до органів (атеросклероз) [8] .

Традиційно, великі гострі цереброваскулярні та серцево-судинні події (ГМКСЕ) є загальним терміном, який використовується для опису різних

результатів і ускладнень серцево-судинних захворювань, таких як смерть, повторний інфаркт або інсульт, серцева недостатність, кардіогенний шок та інші . Крім MACCE, рівень смертності також часто стає основним результатом моделей прогнозування та показників ризику в серцево-судинній медицині . Деякі машини навіть сконструйовані таким чином, щоб мати можливість прогнозувати декілька параметрів, таких як ризик смерті за 10 років і ризик серйозних гострих подій, таких як повторний інфаркт, і ймовірність серцевої недостатності після ІМ за один укол. Найбільш часто використовувані моделі прогнозування ризику серцево-судинних захворювань включали GRACE, TIMI та Framingham Risk Score [9] ,[10], [11].

З іншого боку, підхід до штучного інтелекту (AI) імітує та імітує людський інтелект для навчання, міркування та вирішення проблем. ШІ включає сімейство піддоменів, таких як машинне навчання , еволюційне обчислення, обробка природної мови , робототехніка та інші. Машинне навчання (ML) — популярний підхід, який використовується в контексті медичних досліджень для автоматизації аналітичного моделювання даних. ML вчиться на даних, визначає закономірності та приймає рішення. ML об’єднує та інтерпретує складні дані, такі як біомедичні дані та дані охорони здоров’я, у сценаріях, коли традиційні статистичні методи можуть не працювати [12] .

Досягнення комп’ютерних наук і потяг до точних ліків призвели до зростання багатовимірних даних з багатьох областей. Отже, існує нагальна потреба в інструментах і моделях, які могли б зчитувати та інтерпретувати ці складні та великі обсяги даних, а також точно прогнозувати результати чи прогнозувати ризики. Найкраща модель прогнозування, яка забезпечує оптимальну продуктивність, залежить від різних факторів, таких як мета та мета побудованих моделей, узагальнення та надійність цих моделей і відтворювані результати в реальному клінічному застосуванні.

1.2 Традиційні підходи до передбачення ризику серцево-судинних захворювань

До розвитку інформаційних технологій більшість дослідників і вчених використовували базові статистичні дані, щоб робити висновки та висновки на основі своїх висновків. Ці підходи зазвичай включають лінійну регресію, яка пізніше імпровізується до логістичної регресії (LR), хребтової регресії та інших операцій відповідно до потрібних проблем. Сучасні клінічні практичні рекомендації (CPG) щодо серцево-судинних захворювань, розроблені різними серцево-судинними товариствами, неявно базуються на «середньому пацієнті». Більшість сучасних CPG все ще покладаються на звичайні показники ризику в деяких частинах процедур стратифікації пацієнтів, як рекомендовано Європейським товариством кардіологів (ESC) та Американським коледжем кардіологів (ACC)/Американської кардіологічної асоціації (AHA) [15 , 16]. Тим не менш, ці звичайні методи прогнозування ризику все ще є золотим стандартом і широко використовуються сьогодні. Ймовірно, це пов’язано з простотою використання та швидкими обчисленнями в надзвичайних ситуаціях і ситуаціях, коли доступні обмежені ресурси. Найбільш часто використовувані традиційні підходи до прогнозування ризику серцево-судинних захворювань включали пропорційну небезпеку Кокса (Cox PH) [17–23] , Фрамінгемську оцінку ризику (FRS) [24] , [25] , [26] , [27] , [28] , [29] , 32] , Тромболізис при інфаркті міокарда (TIMI) [10 , 15 ,16 , 30 , 31] , Систематична оцінка коронарного ризику (SCORE) [25 , 32 , 33] , Глобальний реєстр гострих коронарних подій (GRACE) [9 , 34] , QRESEARCH RISK (QRISK) [20 , 21 , 26] , анамнез, ЕКГ, вік, фактори ризику та тропонін (СЕРЦЕ) [35–36] . Резюме вибраних найсучасніших стандартних досліджень оцінки ризику наведено в таблиці 1.1 .

Таблиця 1.1. Резюме вибраних досліджень ССЗ з традиційним підходом до оцінки ризику.

Модель	**Дані / Розмір вибірки**	**Навчальна територія**	**Значні фактори ризику / продуктивність**	**Посилання**
Оцінка ризику Framingham (FRS)	Дослідження серця Фремінгема (1971-1996) /2,716M, 3,500F	10-річні ризики ІХС для недіагностованого пацієнта для профілактики ІХС.	ЛПВЩ, хол та супутні захворювання (куріння, СД) та лікування Hyt.	Lloyd-Jones et al. (2004) [24]
	ВІЛ-інфікований пацієнт / 997	Порівняйте ризики ССЗ у ВІЛ-інфікованих пацієнтів	Стать і вік	Krikke et al. (2016) [32]
Тромболізис при інфаркті міокарда (TIMI)	NSTEACS (14 випробувань TIMI) / 66252	Режим і терміни смерті для поста NSTEMI/UA	Рецидив інфаркту міокарда при 30d смертності, раптова смерть після 30d смертності	Berg et al, (2018) [33]
	Набір даних Сінгапуру / 15,151 (10,100 китайців, 3,005 малайців, 2,046 індійців)	Підтвердження GRACE серед населення Азії	9/10 предикторів. с-статистика = 0,85	Чан та ін. (2011) [34]
	QRESEARCH / 1,283,174	10-річна модель серцево-судинного ризику для Великобританії, отримана на основі ФРС	Вік, Tchol/HDL. IMT, СБП, передчасні ССЗ в сімейному анамнезі, куріння. Індекс депривації Таунсенда, ≥ 1 використання лікування АТ	Hippisley-Cox et al. (2007) [20]
Систематична оцінка коронарного ризику (SCORE)	QRESEARCH / 10 мільйонів	10-річна модель серцево-судинного ризику для Великобританії, отримана на основі ФРС Діагностувати NSTEACS та прогнозувати результат ССЗ	Додано ХХН, std. dev. для варіабельності СБП, мігрені, кортикостероїдів, СЧВ, важких психічних захворювань. с-стат = 0,87	Hippisley-Cox et al. (2017) [21]
	122 біль у грудях при невідкладній ситуації	10-річна модель серцево-судинного ризику для Великобританії, отримана на основі ФРС Діагностувати NSTEACS та прогнозувати результат ССЗ	Hyt, FHx	Six et al. (2008) [35]
Глобальний реєстр гострих коронарних подій (GRACE)	2440 Біль у грудях при невідкладній кардіологічній допомозі (10 лікарень)	10-річна модель серцево-судинного ризику для Великобританії, отримана на основі ФРС Діагностувати NSTEACS та прогнозувати результат ССЗ	с-статистика: СЕРЦЕ = 0,83, TIMI = 0,75, БЛАГОДАТЬ = 0,70	Backus et al. (2013) [36]
	2440 Біль у грудях при невідкладній кардіологічній допомозі (10 лікарень)		с-статистика: СЕРЦЕ = 0,83, TIMI = 0,75, БЛАГОДАТЬ = 0,70	Backus et al. (2013) [36]

Важко точно визначити найточнішу модель прогнозування чи оцінку ризику, оскільки ці моделі були розроблені на основі різних цілей/використання. FRS, QRISK і GRACE можуть мати однакову мету щодо 10-річного прогнозування ризику серцево-судинних захворювань, тоді як TIMI більше зосереджена на відповідному початку медикаментозної терапії. З іншого боку, QRISK нещодавно оновлюється для більш надійного прогнозування серцево-судинних захворювань, яке включає параметри для цереброваскулярних захворювань та імунних захворювань .

На вибір моделі особливо впливають функції моделей. Деякі моделі вимагають простого дослідження та розрахунку, тоді як інші є більш складними з додатковими фізичними та лабораторними значеннями. FRS є одним з найпростіших, який не містить функцій ЕКГ; TIMI використовує різні параметри для оцінки UA/NSTEMI від оцінки STEMI, а GRACE включає креатинін, тоді як QRISK розширює функції, включаючи етнічну приналежність, поштові індекси, IMT та деякі захворювання мозку та імунні захворювання. З іншого боку, SCORE – це таблиця з кольоровим кодуванням ризику на основі статі, віку, статусу куріння, систолічного артеріального тиску та холестерину.

Крім того, незважаючи на наявність схожих «звичайних» функцій, схема підрахунку балів, визначення та стратифікація функцій відрізняються в різних моделях. Наприклад, для віку пацієнта в TIMI та HEART використовуються граничні значення, у FRS і SCORE використовуються варіанти діапазону фіксованого інтервалу, тоді як у GRACE та QRISK поле відкрите та буде стратифіковано на основі локальних обмежень. Проте FRS і TIMI було запропоновано використовувати в різних версіях клінічних практичних рекомендацій (CPG).

Тим не менш, усі ці показники ризику та моделі були вивчені та перевірені/підтверджені на різних групах населення та в різних регіонах світу, що робить їх більш популярними та надійними, ніж інші моделі.

Отже, традиційні методи передбачення серцево-судинних захворювань обмежені урахуванням складних залежностей та нелінійності між ознаками. Вони використовують прості алгоритми класифікації, не адаптуються до змінних умов, чуткі до шуму та викидів даних, мають обмежені можливості з великими обсягами даних.

1.3 Підхід до прогнозування ризиків за допомогою штучного інтелекту

Штучний інтелект (ШІ) — це широкий термін, який описує будь-які обчислювальні програми, які імітують людський інтелект у прийнятті рішень, вирішенні проблем і навчанні. Постійний розвиток методів штучного інтелекту, головним чином у піддоміні машинного навчання (ML), швидко привернув увагу клініцистів до створення нових інтегрованих, надійних та ефективних методів надання якісної медичної допомоги . Візуалізація є однією з головних тем інтересів, коли справа доходить до ШІ в серцево-судинній медицині , крім виявлення біомаркерів і догляду за пацієнтами [37] .

Більшість звичайних підходів були розроблені на основі «середнього пацієнта». Однак ті самі загальні рубрики часто є неадекватними при моделюванні пацієнтів через складність патофізіології людини . Таким чином, рекомендації можуть бути незастосовними до деяких пацієнтів. Таким чином, підходи штучного інтелекту, зокрема ML і глибоке навчання (DL), використовуються для допомоги в розробці стандартизованих прогностичних моделей , які могли б допомогти кардіологам розробити рекомендації для конкретних пацієнтів і розширити процес прийняття рішень щодо конкретних пацієнтів. Важливо, що це допоможе медичним працівникам зекономити час і покращити взаємодію між пацієнтом і постачальником [12] .

Машинне навчання (ML) пропонує кілька переваг перед звичайною статистикою, незважаючи на подібні основи математики. Переваги полягають у зменшенні витрат часу на вибір ознак, непараметричної та нелінійної

інтерпретації взаємодії, включаючи відсутність даних, що зазвичай вводиться до кількох імпутацій [38] . Основна відмінність між статистичними методами та методами ML полягає в тому, що перші в першу чергу допомагають зрозуміти взаємозв'язки між обмеженою кількістю змінних, а другі сприяють ідентифікації нових змінних із даних і виконують кращий прогноз [12]. Відношення між змінними в ML може бути важко інтерпретувати, особливо в архітектурі чорного ящика (тобто ANN). Однак за допомогою обчислювальної потужності аналіз даних витрачається швидше, ніж традиційний підхід. Машинне навчання може використовуватися для вибору функцій, класифікації або обох. У таблиці 1.2 порівнюються основні відмінності між традиційним підходом оцінки ризику та підходом до прогнозування ризику на основі ШІ в дослідженнях ССЗ. У наступному розділі описано різні комбінації та порівняння алгоритмів штучного інтелекту для моделювання та оцінки моделей прогнозування серцево-судинних захворювань/балів ризику.

Таблиця 1.2 . Традиційний підхід до прогнозування ризику проти підходу до прогнозування ризику ШІ при ССЗ.

Особливість	**Традиційний підхід оцінки ризику**	**Підхід до прогнозування ризиків на основі ШІ**
Гіпотеза	Так	Немає
Підхід	Оцінює та пояснює дані	Практичне передбачення на основі даних
Вимірювання	Коефіцієнти відповідності	Точність, відкликання, F-міра, точність, площа під кривою
Здатність до навчання	Немає	Так
Розмір даних	Дані відповідного розміру, які відповідають певній гіпотезі	Великі та складні дані
Тип даних	Однотипні дані, структуровані дані	Підтримує багатомодальні дані, структуровані та неструктуровані дані
Модель	Проста параметрична модель	Комплексна непараметрична модель
Вихід	Підтвердити гіпотезу, причинність	Прогнозуйте нові дані, виявляйте приховані закономірності
Обмеження	Потрібні додаткові припущення, дані низької розмірності	Переобладнання, конфіденційність даних і проблеми безпеки
Фактори ризику	Лише клінічні/демографічні фактори, наприклад вік, стать, етнічна приналежність, куріння, гіпертонія тощо.	Мультимодальність, наприклад, вік, стать, етнічна приналежність, варіації ЕКГ, диференційована експресія генів, ознаки зображень тощо.

ML можна розділити на три типи навчання: контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням. Контрольоване навчання, отримане з даних, позначених людиною, для виконання передбачення. І навпаки, неконтрольоване навчання знаходить приховані закономірності без зворотного зв'язку з боку людини. Тим часом навчання з підкріпленням вчиться на серії підкріплень (винагород) у середовищі. Винагородою є постійний зворотний зв'язок для коригування дій (динамічне навчання). Мета полягає в тому, щоб максимізувати винагороду, тобто оптимізувати середовище [39] .

Зазвичай машинне навчання, що застосовується в кардіології, передбачало втручання експертів або у вибір змінних (тобто ручний вибір), або в оцінку продуктивності моделі. В середньому було відібрано 27 змінних, а до оцінювання було залучено трьох експертів [38 , 40 – 43] . Тим не менш, експерти відіграли важливу роль в оцінці рідкісних інцидентів, особливо в клінічних умовах із різноманітними захворюваннями людини та їх проявами або характеристиками, як про це повідомили Праната та ін. [44] .

Деякі часто використовувані алгоритми ML – це базова регресія (лінійна та логістична регресія), k-найближчий сусід (kNN), опорна векторна машина (SVM), наївний Байєс (NB), дерева рішень (DT) і штучна нейронна мережа (ANN) з ШНМ є найбільш поширеним ML [42 , 43 , 45–48] . Ключовою особливістю ШНМ є паралелізм , де можливе швидке обчислення, коли мережа реалізована на паралельних цифрових комп'ютерах або спеціальному обладнанні [49]. Водночас здатність SVM надійно працювати з розрідженими та зашумленими даними робить їх кращим вибором у багатьох програмах, оскільки він може працювати з «ядрами», які автоматично реалізують нелінійне відображення в просторі функцій [50] .

Крім того, останнім часом ансамблеві методи стають дедалі глибшими з машинами посилення градієнта (GBM) і випадковим лісом (RF) як найпопулярнішими методами серед дослідників через те, що ШНМ породжує проблему чорного ящика (взаємодія змінних може бути незрозумілою чи

інтерпретованою).) [38 , 51 – 53] . GBM і RF — це ансамблі дерев рішень, де застосовувалися методи пакетування або завантаження .

У таблиці 3 наведено зведення вибраних досліджень серцево-судинних захворювань із підходом ML. LR зазвичай вставляється як стандартна базова лінія для порівняння продуктивності інших алгоритмів ML. k-NN є одним із найпростіших алгоритмів серед методів ML , але проблеми цього методу включають обчислювальну потужність (час роботи) при ідентифікації найближчого сусіда, особливо з даними великої розмірності [54] . У кількох дослідженнях було виявлено, що точність не є хорошим показником ефективності через кілька факторів, таких як дані дисбалансу, занадто мало змінних, перетворення безперервних змінних у категоричні дані (ступінь дискретного зміщення), хибнопозитивні та хибнонегативніпитань. Таким чином, площа під робочою кривою приймача (AUROC) є точнішою для оцінки та порівняння продуктивності цих моделей.

Більшість досліджень виявили, що вік, стать (гендер) та етнічна приналежність є одними з основних значущих факторів у прогнозуванні ризику серцево-судинних захворювань, крім історії пацієнта, куріння та супутніх захворювань. Було виявлено, що вік, глікований гемоглобін (HbA1c) і альбумінурія були одними з високоасоційованих прогностичних факторів серцево-судинних захворювань у пацієнтів з діабетом [52] . Оскільки більшість факторів ризику можна контролювати, дієта людини, окрім способу життя, робить значний внесок у прогностичні заходи [48] .

Корисність прогностичної моделі на основі машинного навчання залежить від таких факторів, як неоднорідність даних , глибина та широта даних, окрім характеру завдання моделювання, вибору алгоритмів машинного навчання та ортогональних доказів (незалежні змінні, зміщення та дисперсія). Проблеми ML, включаючи розробку вказівок щодо обміну даними для збору даних і стандартизації ML, а також питання безпеки даних і захисту конфіденційності. Крім того, існували різні форми медичних даних про серцево-судинні захворювання, але вони часто зберігаються в сховищах, які

важко використовувати для серцево-судинних досліджень (різні формати та дані захищені через конфіденційність пацієнта). Інші виклики включали автоматизовану абстракцію та ручне курування технічної компетентності [12 , 55] .

Зображення серця з різноманітними модальностями, представленими за допомогою різних форматів і розмірності. Подібним чином дослідження omics , які з’явилися у звичайній практиці, також були включені в деякі моделі прогнозування ризику, які включають геномний компонент [56] . У найближчому майбутньому, можливо, компоненти молекулярного профілювання або компоненти паноміки можна буде ввести для більш точного прогнозування та інструментів діагностики на основі сигнатур у прецизійній медицині [57 , 58] .

Отже, на основі огляду існуючих рішень було обрано кілька методів класифікації для модуля інтелектуального передбачення серцево-судинних захворювань, зокрема логістичну регресію, KNN, SVM, дерево прийняття рішень, випадковий ліс, AdaBoost, градієнтний бустінг, XGBoost та MLPClassifier. Цей вибір був зумовлений їхньою точністю, швидкістю та здатністю працювати з даними високої розмірності.

1.4 Постановка задачі дослідження

Актуальність модуля інтелектуального передбачення серцево-судинних захворювань зумовлена тим, що ССЗ є однією з головних причин смертності в усьому світі. Враховуючи високі соціальні та економічні витрати, пов'язані з цими захворюваннями, розробка модуля, який дозволяє точно передбачати ризики розвитку ССЗ, є актуальною та нагальною проблемою.

Такий модуль може допомогти медичним фахівцям та пацієнтам у своєчасному виявленні та моніторингу ризиків, що сприятиме ранньому діагностуванню та вчасному лікуванню. Крім того, інтелектуальне

передбачення ССЗ допомагає у профілактиці та розробці індивідуальних стратегій здорового способу життя для пацієнтів.

Використання штучного інтелекту та машинного навчання в модулях передбачення ризиків ССЗ дає можливість аналізувати великі обсяги медичних даних та отримувати результати з високою точністю та відтворюваністю. Це сприяє підвищенню якості медичної допомоги та забезпечує ефективне використання ресурсів охорони здоров'я.

Метою дипломної роботи є розробка модуля інтелектуального передбачення серцево-судинних захворювань на основі машинного навчання.

Для досягнення цієї мети було необхідно провести теоретичні дослідження та вирішити послідовно завдання, що включають визначення та аналіз:

1. Вивчити використання штучного інтелекту для передбачення серцево-судинних захворювань.
2. Розглянути традиційні підходи до передбачення ризику серцево-судинних захворювань.
3. Описати підхід до прогнозування ризиків за допомогою штучного інтелекту.
4. Розробити алгоритм дослідницького аналізу даних інтелектуального передбачення серцево-судинних захворювань.
5. Охарактеризувати методи класифікації та алгоритми інтелектуального передбачення.
6. Проаналізувати та описати набір даних для дослідження.
7. Здійснити навчання даних, оцінити отримані результати та визначити кращу модель для інтелектуального передбачення серцево-судинних захворювань.

Завдання 1-3 розглянуто в параграфах 1.1 - 1.4 відповідно, а виконання завдань 5-7 представлені у параграфах 2.1-3.4.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Алгоритм дослідницького аналізу даних інтелектуального передбачення серцево-судинних захворювань

Дослідницький аналіз даних - це процес виявлення, опису, інтерпретації та використання статистичних та інших методів для розуміння даних та відкриття нової інформації. Цей аналіз використовується в багатьох галузях, включаючи науку, медицину, соціологію, економіку та інформатику.

Основні етапи дослідницького аналізу даних включають:

1. Збір даних: Першим кроком є збір потрібних даних для проведення дослідження. Це може включати опитування, спостереження, аналіз вже наявних даних тощо.

2. Попередня обробка даних: Після збору даних вони потребують попередньої обробки. Це включає перевірку на відсутність помилок, видалення відхилень, очищення даних від шуму, заповнення відсутніх значень та інші операції, що дозволяють покращити якість даних.

3. Візуалізація даних: Після попередньої обробки дані можуть бути візуалізовані для отримання загального уявлення про їх розподіл та взаємозв'язки між різними змінними. Це може включати побудову графіків, діаграм, дерев рішень тощо.

4. Статистичний аналіз: Застосування статистичних методів для виявлення закономірностей, залежностей та суттєвих різниць у даних. Це може включати розрахунок середніх значень, дисперсії, кореляції, використання тестів гіпотез, аналіз дисперсії тощо.

5. Інтерпретація результатів: Після проведення статистичного аналізу, дослідник аналізує отримані результати та намагається зрозуміти їх значення і вплив на поставлену проблему чи дослідницьке питання. Інтерпретація може включати розробку висновків, визначення статистично значимих залежностей,

ідентифікацію важливих факторів або впливів, формулювання висновків про гіпотези чи прогнозування майбутніх тенденцій.

6. Використання результатів: Після інтерпретації результатів, отримані висновки та знання можуть бути використані для прийняття рішень, формулювання рекомендацій, розробки стратегій або планування подальших досліджень. Результати дослідження можуть мати велике значення для наукових досліджень, розвитку нових технологій, вдосконалення бізнес-процесів тощо.

Важливо зазначити, що дослідницький аналіз даних може варіюватися залежно від конкретної задачі, доступних даних та використовуваних методів. Описаний процес є загальною рамкою, яка може бути адаптована та розширена відповідно до потреб конкретного дослідження.

Далі представлено алгоритм дослідницького аналізу даних для інтелектуального передбачення серцево-судинних захворювань у вигляді псевдокоду:

```
Start

"Завантажити дані
Відобразити описову статистику та інформацію про датафрейм
Підрахувати кількість значень у стовпці 'output'
Побудувати графіки розподілу 'output'
Виділити класифікаційні та безперервні ознаки
Розділити дані на основі ознаки 'sex'
Відобразити заголовки обох піднаборів
Побудувати кругову діаграму для розподілу за статтю
Порівняти кількість '1' серед жінок та чоловіків
Побудувати графіки розподілу 'output' серед жінок та чоловіків
Побудувати boxplot для розподілу віку серед жінок та чоловіків
Побудувати графіки розподілу віку та інших безперервних ознак
Побудувати графіки розподілу віку для 'output' та 'sex'
Побудувати теплокарту кореляцій між ознаками
Відобразити сортовані кореляції з 'output'
Побудувати графік кореляцій з 'output'
END"
```

Цей псевдокод описує дослідження даних щодо серцево-судинних захворювань, аналізуючи різні аспекти даних, такі як розподіл віку, статі, рівні

холестерину тощо. Результатом цього аналізу є графіки та статистичні показники, які можуть допомогти медичним фахівцям краще зрозуміти характеристики даних та знайти кореляції між різними факторами та ризиком розвитку серцево-судинних захворювань.

Для побудови блок-схеми алгоритму дослідницького аналізу даних для інтелектуального передбачення серцево-судинних захворювань розділимо його на частини.

Частина 1 - Завантаження даних та первинний аналіз (Див.дод.А.):

Start

> Завантажити дані

> Відобразити описову статистику та інформацію про датафрейм

> Підрахувати кількість значень у стовпці 'output'

> Побудувати графіки розподілу 'output'

> Виділити класифікаційні та безперервні ознаки

> Розділити дані на основі ознаки 'sex'

> Відобразити заголовки обох піднаборів

> Побудувати кругову діаграму для розподілу за статтю

> END

Частина 2 - Аналіз розподілу 'output' серед жінок та чоловіків (рис.2.1):

Start

> Порівняти кількість '1' серед жінок та чоловіків

> Побудувати графіки розподілу 'output' серед жінок та чоловіків

> Побудувати boxplot для розподілу віку серед жінок та чоловіків

> Побудувати графіки розподілу віку та інших безперервних ознак

> Побудувати графіки розподілу віку для 'output' та 'sex'

> END

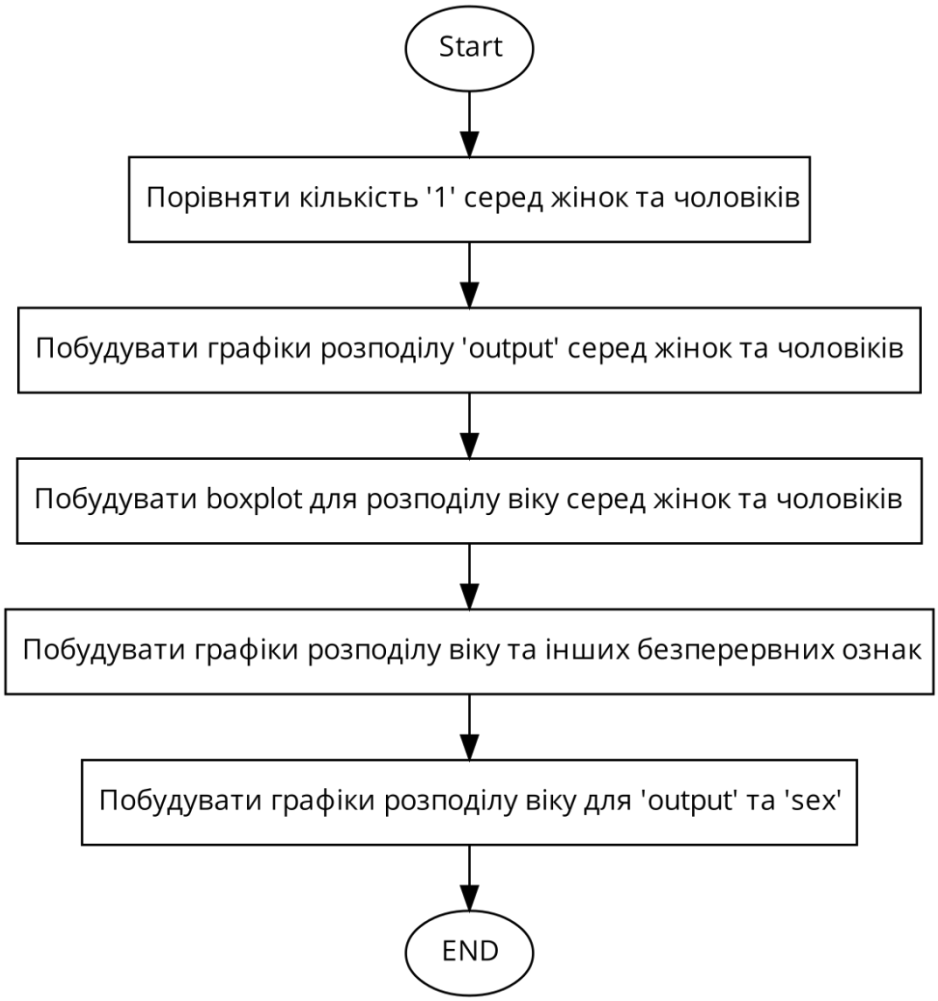


Рисунок 2.1 – Алгоритм розподілу 'output' серед жінок та чоловіків

Частина 3 - Кореляційний аналіз (Рис.2.2):

Start

> Побудувати теплокарту кореляцій між ознаками

> Відобразити сортовані кореляції з 'output'

> Побудувати графік кореляцій з 'output'

> END

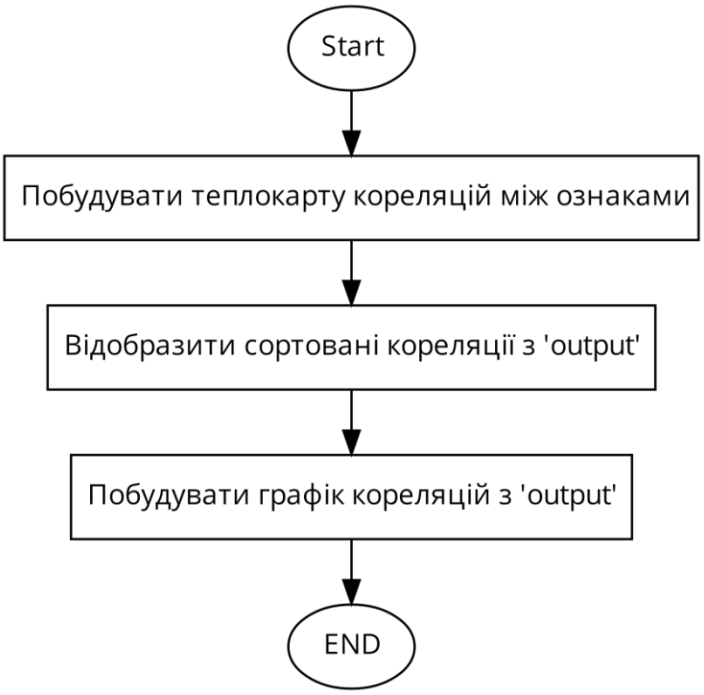


Рисунок 2.2 – - Алгоритм проведення кореляційного аналізу

Отже, описано алгоритм дослідницького аналізу даних для інтелектуального передбачення серцево-судинних захворювань. Він включає завантаження даних, відображення статистики та графіків розподілу. Дані аналізуються за статтю, віком та іншими ознаками, щоб виявити закономірності та кореляції між ними. Цей аналіз є важливим кроком у підготовці до розробки модуля інтелектуального передбачення серцево-судинних захворювань.

2.2 Опис методів класифікації

У даному параграфі будуть описані різні методи класифікації, які були проаналізовані у попередньому розділі. Зокрема, розглянемо такі методи, як Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGBoost Classifier та MLPClassifier.

Кожен з цих методів має свої особливості та переваги в різних ситуаціях. Logistic Regression є лінійним методом класифікації, який використовує логістичну функцію для оцінки ймовірності належності до певного класу. K-Nearest Neighbours використовує найближчих сусідів для призначення класу новому зразку. Support Vector Machines використовує гіперплощини для розділення класів у просторі вищої розмірності. Decision Tree Classifier побудовує дерево рішень для класифікації на основі послідовного вибору найбільш інформативних ознак. Random Forest Classifier поєднує кілька дерев рішень, щоб отримати більш точний результат. AdaBoost Classifier використовує ансамбль слабких класифікаторів для покращення точності класифікації. Gradient Boosting Classifier також використовує ансамбль класифікаторів, але покращує їх, поступово навчаючи їх виправляти помилки попередніх моделей. XGBoost Classifier є розширеною версією Gradient Boosting Classifier з додатковими функціональностями. MLPClassifier (Multilayer Perceptron Classifier) представляє нейронну мережу з кількома прихованими шари, що використовується для класифікації.

Logistic Regression - це статистичний метод, який використовується для моделювання ймовірності належності спостережень до певного класу. В основі цього методу лежить логістична функція, яка відображає лінійну комбінацію ознак у діапазоні між 0 і 1.

Математично, логістична регресія моделює ймовірність p, що спостереження x належить до певного класу, за допомогою логістичної функції:

$$p = \frac{1}{(1 + e^{-z})}$$

де z є лінійною комбінацією ознак і їх ваг:

$$z = b0 + b1 * x1 + b2 * x2 + ... + bn * xn$$

У цій формулі b0, b1, ..., bn є коефіцієнтами регресії, які підлягають оцінці під час навчання моделі. Ці коефіцієнти визначають вплив кожної ознаки на результат класифікації.

Основним завданням логістичної регресії є знаходження оптимальних значень коефіцієнтів, які найкраще апроксимують дані. Це може бути досягнуто шляхом максимізації функції правдоподібності або мінімізації функції втрат, такої як середньоквадратичне відхилення. Для цього використовуються методи оптимізації, такі як градієнтний спуск.

K-Nearest Neighbors (KNN) - це невідомий алгоритм класифікації та регресії, що використовується для вирішення завдань навчання з учителем. В класифікації KNN знаходить кількість k-сусідів, які найбільш близькі до вхідних даних, а потім визначає, який клас має найбільшу кількість представників серед цих сусідів. У регресії KNN знаходить k-сусідів та повертає середнє значення цих сусідів як відповідь.

Алгоритм KNN можна описати наступним чином:

1. Обчислити відстань між кожним з навчальних прикладів і вхідним прикладом.
2. Вибрати k найближчих навчальних прикладів до вхідного прикладу.
3. Визначити клас, який найбільше представлений серед k найближчих прикладів, якщо застосовується класифікація, або відповідне середнє значення, якщо застосовується регресія.

Для визначення відстані між двома прикладами, зазвичай використовуються такі метрики, як Евклідова відстань, Манхеттенська відстань, метрика Мінковського та Косинусна схожість.

Support Vector Machines (SVM) - це алгоритм навчання з учителем, який використовується для задач класифікації та регресії. Основна ідея SVM полягає в тому, щоб знайти оптимальну гіперплощину (у двовимірному просторі - лінію) яка найкраще розділяє два класи даних. Ця гіперплощина визначається таким чином, щоб максимізувати відстань (заздалегідь

визначену як "зазор") між найближчими прикладами двох класів, відомими як опорні вектори.

SVM можна описати наступним чином:

1. Представлення даних: Кожен приклад даних представляється як вектор ознак у n-вимірному просторі.

2. Пошук оптимальної гіперплощини: SVM намагається знайти гіперплощину, яка найкраще розділяє два класи даних. Ця гіперплощина визначається як гіперплощина з найбільшим зазором між опорними векторами двох класів.

3. Опорні вектори: Опорні вектори - це навчальні приклади, які знаходяться найближче до гіперплощини. Вони визначають положення та орієнтацію гіперплощини.

4. Функція рішення: Після тренування SVM може використовуватися для класифікації нових прикладів, шляхом визначення, до якої сторони гіперплощини вони належать.

Для досягнення оптимальності SVM використовує опорні вектори, які є невеликою підмножиною навчальних прикладів. Використовуючи метод опорних векторів, SVM розмежовує класи даних, роблячи рішення на основі найближчих опорних векторів.

Decision Tree Classifier (Класифікатор на основі дерева рішень) - це алгоритм машинного навчання, який використовується для задач класифікації та регресії. Він будує модель у формі дерева, де кожен вузол представляє рішення на основі значення певного ознаки.

Decision Tree Classifier можна описати наступним чином:

1. Побудова дерева: Алгоритм починає з кореневого вузла дерева і обирає найкращу ознаку для розподілу даних на підмножини. Вибір найкращої ознаки зазвичай здійснюється на основі критерію вимірювання неоднорідності, такого як Gini impurity або ентропія Шеннона.

2. Розбиття даних: Дерево розгалужується на дві або більше гілки, в залежності від значення обраної ознаки. Кожна гілка представляє підмножину даних з відповідним значенням обраної ознаки.

3. Рекурсивне побудова дерева: Процес побудови дерева повторюється для кожної нової гілки, доки не буде досягнуто певної умови зупинки, наприклад, досягнення максимальної глибини дерева або мінімальної кількості прикладів у вузлі.

4. Класифікація: Коли дерево побудовано, для нових прикладів воно може використовуватися для класифікації шляхом проходження через вузли дерева згідно з умовами розбиття та значеннями ознак.

В Decision Tree Classifier рішення приймаються на основі порівняння значень ознак з пороговими значеннями. Цей процес повторюється на кожному вузлі дерева, що дозволяє алгоритму розділити дані на декілька класів.

Random Forest Classifier (Класифікатор на основі випадкового лісу) - це ансамбль алгоритмів машинного навчання, який використовує багато рішень дерева рішень для вирішення задач класифікації та регресії. Кожне дерево випадкового лісу будується на основі випадкового вибору підмножини даних та випадкового вибору підмножини ознак.

Random Forest Classifier можна описати наступним чином:

1. Випадковий вибір підмножини даних: Задля будівництва кожного дерева випадкового лісу з вихідного набору даних випадковим чином вибирається підмножина прикладів. Це забезпечує різноманітність в даних, яка допомагає зменшити зміщення моделі.

2. Випадковий вибір підмножини ознак: З кожного підмножини даних випадковим чином вибирається підмножина ознак, які будуть використовуватися для побудови дерева рішень. Це дозволяє зробити кожне дерево незалежним від інших і сприяє різноманітності моделі.

3. Побудова дерева рішень: Для кожного дерева випадкового лісу використовується алгоритм побудови дерева рішень, такий як алгоритм ID3

або CART. Кожне дерево розгалужується на основі значень ознак та критеріїв неоднорідності, таких як Gini impurity або ентропія Шеннона.

4. Класифікація або регресія: Коли всі дерева побудовано, для нових прикладів вони проходять через кожне дерево, і результати голосування або середнє значення використовуються для класифікації або регресії.

Random Forest Classifier комбінує прогнози кожного дерева випадкового лісу для прийняття кінцевого рішення. Це досягається шляхом голосування (у випадку класифікації) або середнього значення (у випадку регресії) прогнозів, отриманих від кожного дерева. Така ансамбль підходу забезпечує більш точні прогнози, адже різні дерева можуть виявити різні аспекти даних та компенсувати одне одного.

AdaBoost Classifier - це ансамбльний метод машинного навчання, який використовується для задач класифікації. Він поєднує кілька слабких класифікаторів (наприклад, дерев рішень) із важливістю, наданою кожному класифікатору.

AdaBoost Classifier можна описати наступним чином:

1. Ініціалізація ваг: Кожен приклад даних починається з однаковою вагою, яка поступово оновлюється на кожному етапі бустінгу.

2. Побудова слабкого класифікатора: На першому етапі побудовується слабкий класифікатор, який намагається класифікувати дані якнайкраще, використовуючи поточні ваги.

3. Оновлення ваг: Після класифікації на кожному етапі, ваги прикладів даних оновлюються. Приклади, які були класифіковані невірно, отримують більшу вагу, тоді як правильно класифіковані приклади отримують меншу вагу.

4. Побудова наступного слабкого класифікатора: Процес побудови слабкого класифікатора повторюється з оновленими вагами. Кожен наступний класифікатор намагається сконцентруватись на прикладах, які були неправильно класифіковані попередніми класифікаторами.

5. Голосування або комбінування: На кожному етапі прогнози кожного слабкого класифікатора враховуються залежно від їхньої точності та ваги, наданої кожному класифікатору. Кінцевий прогноз отримується шляхом голосування або взваженого комбінування прогнозів.

Таким чином, AdaBoost Classifier працює шляхом послідовного побудови і комбінування слабких класифікаторів з урахуванням їхньої точності та ваги. Кінцева модель бустінгу є комбінацією цих слабких класифікаторів, з вагами, що відображають їхню важливість у прийнятті рішення.

Gradient Boosting Classifier є ітеративним алгоритмом, який побудований на основі ідеї градієнтного спуску для мінімізації функціоналу помилки. В основі його лежить використання градієнта функції втрати для поступового покращення моделі. Алгоритм починається з пошуку початкової моделі, яка може бути, наприклад, константним значенням. Потім, на кожному наступному кроці, виконується наступний процес:

1. Обчислення градієнта функції втрати відносно поточної моделі.

2. Побудова нового слабкого класифікатора, який намагається апроксимувати градієнт функції втрати.

3. Оптимізація ваги нового класифікатора шляхом використання методів оптимізації, таких як градієнтний спуск.

4. Оновлення моделі шляхом додавання нового класифікатора з оптимальною вагою.

5. Повторення кроків 1-4 доки не буде досягнуто заданої кількості ітерацій або поки функціонал помилки не буде зменшено до мінімуму.

Таким чином, Gradient Boosting Classifier поступово покращує модель, додаючи нові слабкі класифікатори, які спрямовані на зменшення помилки. Ваги кожного класифікатора визначаються на основі їхнього внеску у зменшення функціоналу помилки.

XGBoost Classifier (Extreme Gradient Boosting Classifier) є потужним алгоритмом градієнтного бустингу, який використовує дерев'яні моделі для

класифікації. Він є розширенням градієнтного бустингу і використовує оптимізовану версію градієнтного підйому для покращення результатів.

XGBoost Classifier можна описати наступним чином:

1. Початкова модель:

- Початкова модель може бути константним значенням або невеликими дерев'яними моделями.

- Нехай $M_0(x)$ буде початковою моделлю, яка приймає вхідний вектор x і видає початкове прогнозоване значення.

2. Функція втрати:

- XGBoost використовує функцію втрати для оцінки помилки між прогнозованими значеннями і справжніми мітками даних.

- Функція втрати може бути задана відповідно до вибраної задачі класифікації, наприклад, логарифмічна втрата для бінарної класифікації.

3. Оптимізація функції втрати:

- XGBoost використовує оптимізовану версію градієнтного підйому для мінімізації функції втрати.

- Кожна наступна модель намагається скоригувати помилки, зроблені попередніми моделями.

- Градієнт функції втрати обчислюється для кожного зразка даних, а потім використовується для побудови наступної моделі.

4. Побудова нової моделі:

- Побудова нової моделі полягає в створенні дерев'яної моделі, яка намагається апроксимувати градієнт функції втрати.

- Дерев'яна модель будується шляхом рекурсивного розбиття даних на підмножини на основі певних ознак і значень, щоб максимізувати зменшення втрат.

5. Оптимізація гіперпараметрів:

- XGBoost використовує регуляризацію та оптимізацію гіперпараметрів для контролю складності моделей та попередження перенавчання.

- Гіперпараметри, такі як швидкість навчання, глибина дерева, кількість дерев та інші, можуть бути налаштовані для досягнення кращої точності та уникнення перенавчання.

6. Ансамблювання моделей:

- XGBoost поєднує прогнози багатьох дерев'яних моделей, щоб отримати кінцевий прогноз.

- Це може бути здійснено шляхом агрегування прогнозів кожної дерев'яної моделі або за допомогою голосування або середнього значення прогнозів.

7. Ітерація та додавання нових моделей:

- Попередні кроки повторюються, додаючи нові моделі до ансамблю, доки не буде досягнута задовільна точність моделі або інший критерій зупинки.

Таким чином, XGBoost Classifier використовує градієнтний бустинг та оптимізовану версію градієнтного підйому для покращення якості прогнозів шляхом побудови ансамблю дерев'яних моделей. Він є потужним алгоритмом машинного навчання, який широко використовується для класифікації та регресії завдяки своїй ефективності та точності.

MLPClassifier (Multilayer Perceptron Classifier) є одним із типів штучних нейронних мереж, який використовується для задач класифікації. Він складається зі шарів нейронів, кожен з яких має ваги та активаційну функцію. Оптимізується шляхом навчання за допомогою зворотного поширення помилки (backpropagation).

Давайте розглянемо формулювання MLPClassifier:

1. Вхідний шар:

- Нехай маємо m незалежних ознак, позначимо їх як $x_1, x_2, ..., x_m$.

- Вектор вхідних ознак: $x = [x_1, x_2, ..., x_m]$

2. Приховані шари:

- MLPClassifier містить один або декілька прихованих шарів. Кожен шар складається з n нейронів.

- Для кожного нейрона в прихованому шарі:

- Вагові коефіцієнти: w_1, w_2, ..., w_m

- Зсув (bias): b

- Активаційна функція: f(z), де $z = w_1x_1 + w_2x_2 + ... + w_mx_m + b$.

- Вихід нейрона: a = f(z)

3. Вихідний шар:

- Вихідний шар має k нейронів, де k - кількість класів.

- Кожен нейрон вихідного шару представляє ймовірність належності до відповідного класу.

- Активаційна функція останнього шару зазвичай є softmax.

4. Функція втрат:

- Для класифікації використовується функція категоріальної хрест-ентропії або логарифмічна втрата.

5. Навчання:

- MLPClassifier навчається шляхом зворотного поширення помилки (backpropagation) та оптимізації вагових коефіцієнтів з використанням алгоритму градієнтного спуску.

6. Оптимізація:

- MLPClassifier використовує метод оптимізації, такий як стохастичний градієнтний спуск або метод Адама, для оновлення вагових коефіцієнтів та зсуву.

- Метою оптимізації є зменшення функції втрат і підвищення точності класифікації.

7. Регуляризація:

- Щоб уникнути перенавчання моделі, можуть використовуватися методи регуляризації, такі як L1-регуляризація (Lasso) або L2-регуляризація (Ridge), для контролю вагових коефіцієнтів.

8. Перебір параметрів:

- MLPClassifier може мати декілька гіперпараметрів, таких як кількість прихованих шарів, кількість нейронів у кожному прихованому шарі, швидкість навчання та кількість ітерацій.

- Перебір цих параметрів та вибір оптимальних значень можуть бути важливим етапом для досягнення кращої точності моделі.

MLPClassifier є потужним алгоритмом класифікації, здатним працювати зі складними нелінійними залежностями у даних. Він використовується в багатьох галузях, включаючи комп'ютерне зорове розпізнавання, обробку природних мов, фінансовий аналіз та багато інших задач класифікації.

Отже, проведено огляд основних методів класифікації, використованих для передбачення серцево-судинних захворювань. В ньому розглянуті методи, такі як Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGBoost Classifier та MLPClassifier. Для кожного методу наведено опис його математичних принципів, а також оцінку результатів у вигляді точності, часу виконання та показників ефективності, таких як точність, відновлення, F1-оцінка та середня квадратична помилка. Ці методи виявилися ефективними для класифікації серцево-судинних захворювань та можуть бути корисними для передбачення ризиків та прийняття рішень у медичній практиці.

2.3 Алгоритм інтелектуального передбачення серцево-судинних захворювань

Серцево-судинні захворювання є серйозною проблемою в сучасному світі і потребують ефективних методів діагностики та передбачення. Агоритми інтелектуального передбачення можуть бути корисним інструментом для прогнозування ризику серцевих захворювань та допомогти в прийнятті рішень щодо профілактики та лікування.

У цьому контексті розглядається алгоритм (Див.дод.Б) інтелектуального передбачення серцево-судинних захворювань. Алгоритм передбачення базується на аналізі набору даних, який містить інформацію про різні фактори, такі як вік, стать, артеріальний тиск, рівень холестерину тощо.

Агоритм інтелектуального передбачення серцево-судинних захворювань представлено наступним псевдокодом:

Start

> Розділити набір даних

> Розділити на навчальні та тестові набори

> Масштабувати навчальні та тестові дані за допомогою MinMaxScaler

> Визначити функцію compute

a. Побудувати графік прогнозованих та фактичних значень

b. Побудувати матрицю невідповідностей

c. Розрахувати та надрукувати метрики

> Створити модель

> Навчити модель на навчальних даних

> Зробити прогнози за допомогою моделі на тестових даних

> Розрахувати та надрукувати час виконання моделі

> Викликати функцію compute з прогнозами та істинними значеннями

> END

Алгоритм починається з розділення набору даних на навчальні та тестові набори. Далі, дані масштабуються за допомогою методу MinMaxScaler, що дозволяє нормалізувати значення змінних. Потім модель навчається на навчальних даних та здійснює передбачення на тестових даних.

Після цього виконується обчислення метрик та побудова графіків для оцінки результатів передбачення. Також виконується розрахунок матриці невідповідностей, яка дозволяє оцінити точність передбачень.

Усі ці етапи реалізовані у вигляді функції compute, яка приймає прогнозовані та істинні значення, та виконує необхідні обчислення та виводить результати.

Алгоритм передбачення серцево-судинних захворювань є важливим інструментом для медичних досліджень та практики. Його використання може допомогти в ранній діагностиці та прийнятті своєчасних заходів для запобігання серцево-судинним захворюванням. Застосування алгоритмів інтелектуального передбачення дозволяє зробити прогнози на основі наявних даних і виявити потенційні ризики у пацієнтів.

Перевагою цього алгоритму є його здатність використовувати широкий спектр факторів для передбачення ризику серцево-судинних захворювань. Враховуючи різні характеристики, такі як вік, стать, артеріальний тиск та інші, алгоритм може надати більш точні прогнози.

Результати аналізу даних та обчислення метрик можуть дати медичному персоналу корисну інформацію для прийняття рішень. Наприклад, на основі передбачень ризику серцевих захворювань можуть бути запропоновані індивідуальні плани профілактики та лікування для пацієнтів.

Важливо пам'ятати, що алгоритм передбачення є лише інструментом, і кінцеве рішення про діагноз та лікування має приймати кваліфікований медичний фахівець. Але використання алгоритмів інтелектуального передбачення може значно полегшити процес прийняття рішень та підвищити точність діагностики.

Загалом, алгоритм інтелектуального передбачення серцево-судинних захворювань є потужним інструментом для аналізу та прогнозування ризику у пацієнтів. Його застосування може сприяти ранньому виявленню та ефективному управлінню серцево-судинними захворюваннями, що сприятиме покращенню здоров'я та якості життя пацієнтів.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Опис та аналіз набору даних

Для поставленої задачі обрано датасет "Heart Attack Analysis & Prediction", який є набором даних для класифікації серцевого нападу. Цей датасет можна знайти на платформі Kaggle [66] . Цей набір даних містить інформацію про різні ознаки, пов'язані з серцевими нападами, та мету класифікації наявності серцевого нападу на підставі цих ознак.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
None
```

Рисунок 3.1 - Структура набору даних

Для початку, давайте розглянемо загальну структуру набору даних. Він складається з 303 записів і 14 стовпців. Кожен запис відображає інформацію про конкретного пацієнта. Серед стовпців є числові ознаки, такі як вік пацієнта, артеріальний тиск, рівень холестерину та інші, а також категоріальні ознаки, такі як тип болю в грудях та результати електрокардіографії.

Один з найважливіших аспектів аналізу даних - це розуміння розподілу значень кожної ознаки. Наприклад, використовуючи опис набору даних, ми можемо побачити, що середня вікова група пацієнтів становить близько 54 років, а рівень холестерину в середньому становить приблизно 246 мг/дл.

Далі, можемо проаналізувати розподіл цільової змінної - наявності серцевого нападу. Згідно з описом датасету, значення цієї змінної можуть бути 0 або 1, де 0 вказує на відсутність серцевого нападу, а 1 - на його наявність. Шляхом підрахунку значень можна встановити, що в датасеті 165 пацієнтів не мали серцевого нападу, а 138 пацієнтів мали серцевий напад.

Серед найважливіших ознак є вік пацієнта (Таблиця 3.1), стать, тип болю в грудях, артеріальний тиск, рівень холестерину, результати електрокардіографії та інші показники. Середня вікова група пацієнтів становить близько 54 років, зі значеннями, що варіюються від 29 до 77 років. Також варто зазначити, що більшість пацієнтів (приблизно 68%) є чоловіками. Серед числових ознак, артеріальний тиск (trtbps) та рівень холестерину (chol) мають відносно невеликі стандартні відхилення, вказуючи на меншу розбіжність в цих показниках серед пацієнтів. Однак, треба зазначити, що рівень холестерину має великий максимальний значення 564 мг/дл. Ознака "output" є цільовою змінною, яка вказує на наявність (1) або відсутність (0) серцевого нападу у пацієнта. В датасеті приблизно 54% пацієнтів не мали серцевого нападу, а 46% мали.

Таблиця 3.1. Опис статистичних показників набору даних

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Результат показує розподіл значень цільової змінної "output" в наборі даних. Згідно з результатами, значення "1" зустрічається 165 разів, що вказує на наявність серцевого нападу, тоді як значення "0" зустрічається 138 разів, що вказує на відсутність серцевого нападу. Це важлива інформація, яка допоможе

нам розуміти баланс класів у наборі даних та визначити відсоток пацієнтів з серцевим нападом та без нього.

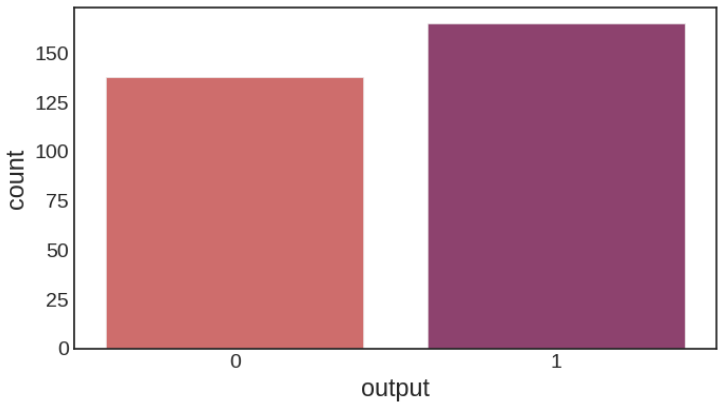


Рисунок 3.2 - Розподіл значень цільової змінної "output"

Результат кругової діаграми (Рис.3.3.) "Розподіл за статтю" показує, що більшість пацієнтів в наборі даних належать до однієї статі. Згідно з діаграмою, близько 68% пацієнтів є представниками однієї статі (чоловічої або жіночої), тоді як решта, що становить близько 31,7%, представлена іншою статтю.

Рисунок 3.3 - Розподіл значень за статтю

З аналізу даних (Рис.3.4) можна зробити такі висновки: за результатами аналізу, близько 44.93% пацієнтів мають високий ризик серцевого нападу

(ознака '1' у стовпці "output"); серед пацієнтів з високим ризиком серцевого нападу, середній вік становить близько 51 року; з іншого боку, близько 75.0% пацієнтів не мають високого ризику серцевого нападу (ознака '0' у стовпці "output"); серед пацієнтів без високого ризику серцевого нападу, середній вік становить близько 55 років. Ці відсотки та середні значення вказують на те, що серед пацієнтів, представлених у даних, існує певний відсоток осіб з високим ризиком серцевого нападу, і вік може впливати на цей ризик, де середній вік пацієнтів з високим ризиком нижчий, ніж середній вік пацієнтів без високого ризику.

```
print("Відсоток '1' з високим ризиком серцевого нападу = {} %" .format(round((len(X[X["output"]==1])/len(X)*100),2)))
print("'1' середній вік високого ризику = {} роки\n" .format(round(X[X["output"]==1]["age"].mean())))

print("Відсоток '0' з високим ризиком серцевого нападу = {} %" .format(round((len(Y[Y["output"]==1])/len(Y)*100),2)))
print("Середній вік високого ризику = {} роки" .format(round(Y[Y["output"]==1]["age"].mean())))

```

```
Відсоток '1' з високим ризиком серцевого нападу = 44.93 %
'1' середній вік високого ризику = 51 роки

Відсоток '0' з високим ризиком серцевого нападу = 75.0 %
Середній вік високого ризику = 55 роки
```

Рисунок 3.4 - Аналіз ризику серцевого нападу за віком

Загалом, результати (Рис.3.5) свідчать про наявність високого ризику серцевого нападу у обох статей, проте в жінок спостерігається трохи більша кількість пацієнтів з високим ризиком у порівнянні з чоловіками. У жінок (стать 0) спостерігається вища кількість пацієнтів з високим ризиком серцевого нападу ('1'), їхня кількість становить 93, в той час як пацієнтів з низьким ризиком ('0') всього 114. У чоловіків (стать 1) також спостерігається значна кількість пацієнтів з високим ризиком серцевого нападу ('1'), їхня кількість становить 72, тоді як пацієнтів з низьким ризиком ('0') всього 24.

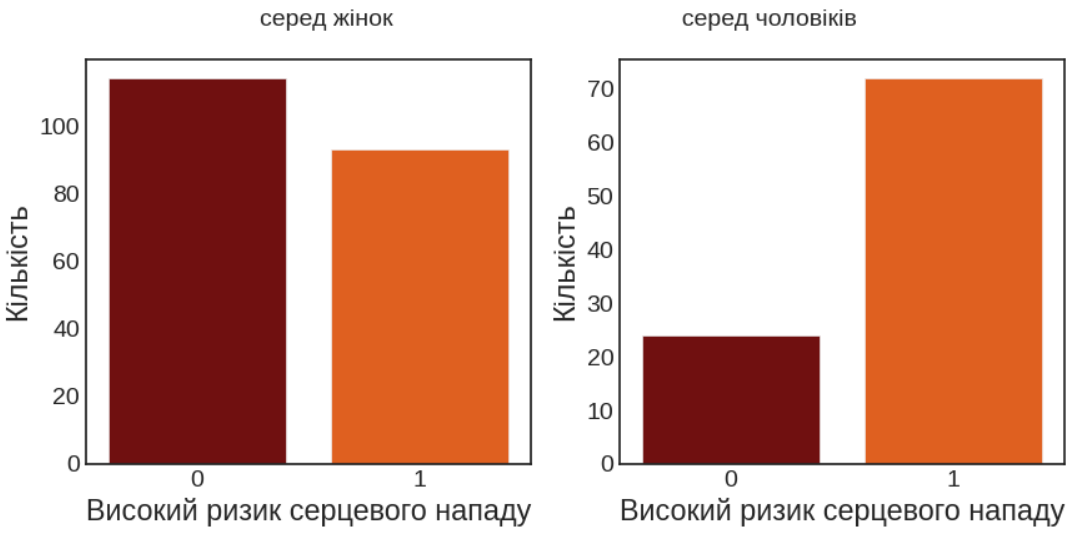


Рисунок 3.5 - Аналіз ризику серцевого нападу за статтю

Як бачимо, у дослідженні (Рис.3.6.) було задіяно більше жінок порівняно з чоловіками, і також діапазон віку жінок був ширший (29-77 років), у порівнянні з чоловіками (34-76 років).

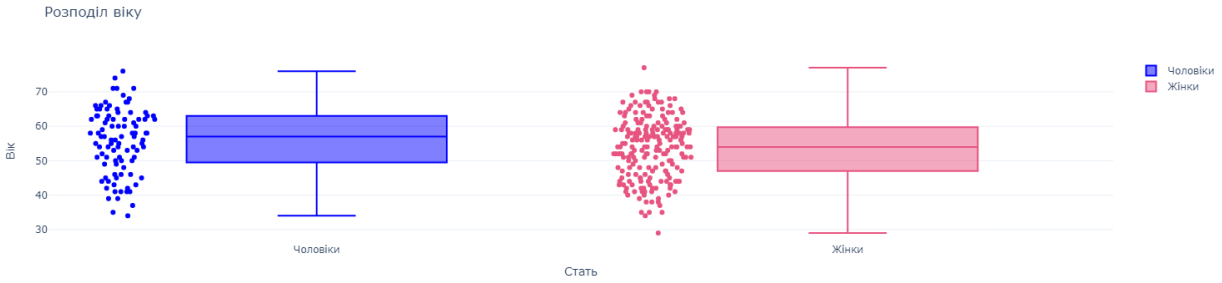


Рисунок 3.6 - Аналіз ризику серцевого нападу за статтю та віком

Результат побудованих графіків (Рис.3.7.) розподілу показує форму розподілу кожного з вибраних стовпців датасету `data1`. Загалом, графіки виглядають нормально, оскільки жоден з них, за винятком "oldpeak", не має значного зміщення вліво або вправо. З цим можна бути задоволеним, оскільки розподіл даних наближений до нормального розподілу, що полегшує подальшу обробку та аналіз даних. Проте, стовпець "oldpeak" може потребувати окремої уваги та обробки на етапі попередньої обробки даних, оскільки його розподіл може бути зміщеним.

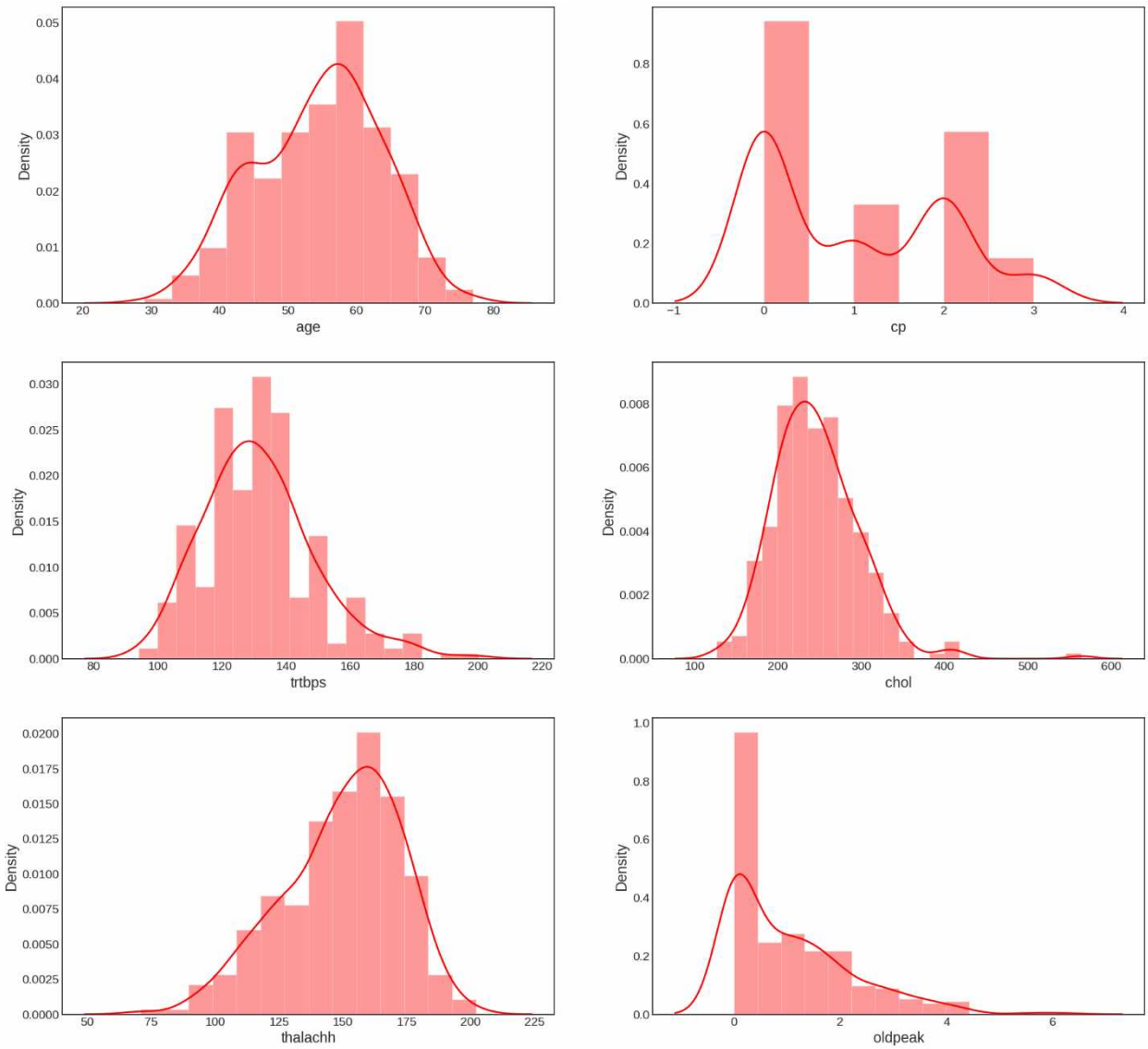


Рисунок 3.7 - Графіки розподілу

Аналізуючи кореляцію (Рис. 3.8) між кожним стовпчиком і вихідним значенням (output), можна побачити такі відносини:

- "exng" (виключений фізичний навантаження): 0.437
- "cp" (біль у грудях): 0.434
- "oldpeak" (депресія сегмента ST після фізичного навантаження): 0.431
- "thalachh" (досягнена максимальна частота серцевих скорочень): 0.422
- "caa" (кількість крупних судин): 0.392
- "slp" (нахил сегмента ST під час фізичного навантаження): 0.346
- "thall" (максимальний відхіл від норми у ході тесту з навантаженням): 0.344

- "sex" (стать): 0.281
- "age" (вік): 0.225
- "trtbps" (артеріальний тиск у спокої): 0.145
- "restecg" (електрокардіографічний результат у спокої): 0.137
- "chol" (рівень холестерину в крові): 0.085
- "fbs" (рівень цукру у крові натще): 0.028

Ці значення кореляції вказують на наявність певних зв'язків між цими факторами та ризиком серцевого нападу (output). Зауважимо, що значення близькі до 1 вказують на сильну позитивну кореляцію, тоді як значення близькі до -1 вказують на сильну негативну кореляцію. За результатами кореляції можна припустити, що фактори, такі як виключений фізичний навантаження, біль у грудях, депресія сегмента ST після фізичного навантаження та максимальна досягнута частота серцевих скорочень, можуть бути добрими показниками ризику серцевого нападу.

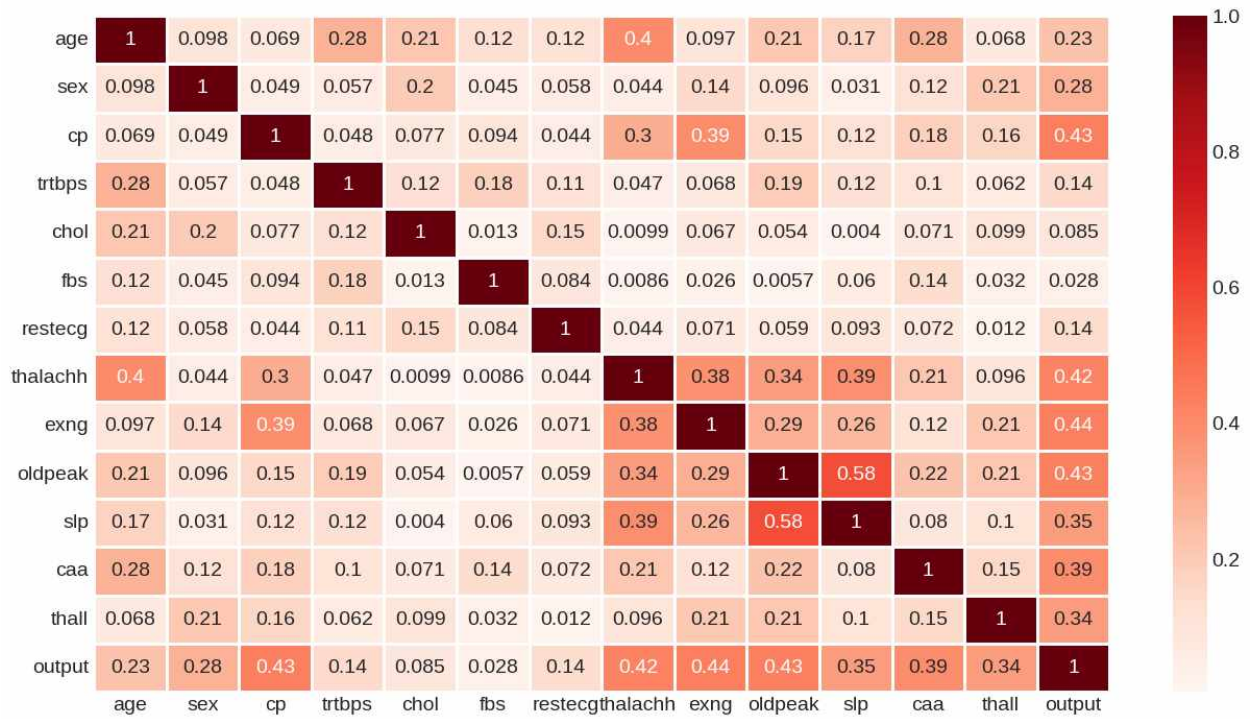


Рисунок 3.8 - Кореляційна матриця

Отже, було проведено аналіз даних про серцеві хвороби. Було використано різні візуалізації та обчислення для вивчення різних характеристик, таких як розподіл віку, статевий розподіл, ризик серцевих нападів та вплив окремих факторів на цей ризик. Загальний вік у досліджуваній групі становить близько 54 років, з медіаною віку 55 років. Чоловіки мають середній вік близько 55 років, тоді як жінки - близько 51 року. За статистикою, 44,93% пацієнтів вважаються з високим ризиком серцевого нападу. Також було досліджено розподіл віку серед різних категорій ризику. Виявлено, що серед тих, хто має високий ризик, середній вік становить близько 51 року, тоді як серед тих, хто має низький ризик, середній вік становить близько 55 років. Також було виявлено зв'язок між окремими факторами та ризиком серцевих нападів. Найбільш значущими факторами, які впливають на ризик, є екзерциційна ангіна (exng), тип болю в грудях (cp), депресія сегменту ST після вправи (oldpeak) та максимальна досягнута частота серцевих скорочень (thalachh).

3.2 Навчання даних

Далі розглянемо процес поділу набору даних на навчальний та тестовий набори, а також процедуру масштабування даних перед їх використанням для навчання моделі. Ці кроки дозволяють ефективно використовувати дані для навчання та оцінки моделі, забезпечуючи надійні результати.

```
#Data Splitting
X=data1.iloc[:,:13]
Y=data1["output"]

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=65)

#MinMax Scaling / Normalization of data
MM_scaler = MinMaxScaler()
X_train = MM_scaler.fit_transform(X_train)
X_test = MM_scaler.fit_transform(X_test)
```

Перш ніж приступити до навчання моделі, розділяємо набір даних на дві частини: навчальний набір і тестовий набір. Це дозволяє оцінити ефективність моделі на нових, раніше невикористаних даних. Для зручності розділу набору

даних використовуємо функцію **train_test_split** з бібліотеки **sklearn.preprocessing**. Ця функція дозволяє випадковим чином розподілити дані на дві групи, зазначаючи відсоток даних, які хочемо призначити для тестування. Такий підхід допомагає нам перевірити, наскільки добре модель працює на нових, невідомих їй даних, і визначити її загальну здатність до узагальнення.

Після поділу набору даних переходимо до масштабування. Цей крок необхідний для того, щоб забезпечити однаковий масштаб усіх ознак даних. Масштабування даних допомагає уникнути проблем, пов'язаних з великими різницями у величині значень між ознаками. Ми використовуємо масштабувальник MinMax, який приводить значення кожної ознаки до діапазону від 0 до 1. Це забезпечує нормалізацію даних та допомагає моделі ефективніше виконувати обчислення.

```
def compute(Y_pred,Y_test):
    #Output plot
    plt.figure(figsize=(12,6))
    plt.scatter(range(len(Y_pred)),Y_pred,color="yellow",lw=5,label="Predictions")
    plt.scatter(range(len(Y_test)),Y_test,color="red",label="Actual")
    plt.title("Prediction Values vs Real Values")
    plt.legend()
    plt.show()

    cm=confusion_matrix(Y_test,Y_pred)
    class_label = ["High-risk", "Low-risk"]
    df_cm = pd.DataFrame(cm, index=class_label,columns=class_label)
    sns.heatmap(df_cm,annot=True,cmap='Pastel1',linewidths=2,fmt='d')
    plt.title("Confusion Matrix",fontsize=15)
    plt.xlabel("Predicted")
    plt.ylabel("True")
    plt.show()

    #Calculate Metrics
    acc=accuracy_score(Y_test,Y_pred)
    mse=mean_squared_error(Y_test,Y_pred)
    precision, recall, fscore, train_support = score(Y_test, Y_pred, pos_label=1, average='binary')
    print('Precision: {} \nRecall: {} \nF1-Score: {} \nAccuracy: {} %\nMean Square Error: {}'.format(
        round(precision, 3), round(recall, 3), round(fscore,3), round((acc*100),3), round((mse),3)))
```

Наведений код є функцією `compute`, яка виконує обчислення та візуалізацію різних метрик та результатів для моделі, що передбачає значення `Y_pred` та використовує вірні значення `Y_test`. Ось опис кожної частини коду:

1. Перший блок коду відповідає за побудову графіку порівняння передбачених (`Y_pred`) та вірних (`Y_test`) значень. Використовуючи бібліотеку `matplotlib.pyplot`, функція створює розсіювальну діаграму, де передбачені значення позначені жовтим кольором, а вірні значення - червоним. Графік допомагає візуалізувати, наскільки добре модель передбачає значення та порівняти їх з реальними значеннями.

2. Наступний блок коду обчислює та відображає матрицю помилок (confusion matrix). Використовуючи бібліотеку `sklearn.metrics.confusion_matrix` для обчислення матриці помилок, функція створює теплову карту (heatmap), де кожна клітина представляє кількість прикладів, які були правильно або неправильно передбачені моделлю. Теплова карта допомагає візуалізувати результати класифікації та оцінити, наскільки добре модель впоралася з різними класами.

3. Наступний блок коду обчислює додаткові метрики, такі як точність (precision), повнота (recall), F1-показник (fscore), точність (accuracy) і середньоквадратична помилка (mean square error). Використовуючи функцію `sklearn.metrics.accuracy_score` для обчислення точності, `sklearn.metrics.mean_squared_error` для обчислення середньоквадратичної помилки та функцію `sklearn.metrics.score` для обчислення точності, повноти та F1-показника, функція виводить ці метрики на екран.

Код надає зручні інструменти для оцінки результатів моделі і порівняння їх зі знаннями про правильні значення. Це допомагає зрозуміти, наскільки ефективна модель у виконанні поставленої задачі. Ця функція може бути корисною для оцінки якості моделі, зокрема в задачах класифікації. Вона дозволяє швидко візуалізувати передбачені та вірні значення, а також обчислити різні метрики, що допомагають оцінити точність моделі. Цей код дозволяє вам отримати важливу інформацію про ефективність вашої моделі та допомагає вам зробити висновки щодо її продуктивності і якості.

```python
#Build Model
start = time.time()

model_Log= LogisticRegression(random_state=10)
model_Log.fit(X_train,Y_train)
Y_pred= model_Log.predict(X_test)

end=time.time()

model_Log_time=end-start
model_Log_accuracy=round(accuracy_score(Y_test,Y_pred), 4)*100 # Accuracy

print(f"Execution time of model: {round((model_Log_time),5)} seconds\n")
#Plot and compute metrics
compute(Y_pred,Y_test)
```

Наведений код будує логістичну регресійну модель, навчає її на тренувальних даних та виконує передбачення на тестових даних. Після цього він обчислює час виконання моделі, точність та викликає функцію **compute** для візуалізації результатів та обчислення метрик. Цей код допомагає нам навчити логістичну регресійну модель на тренувальних даних, виконати передбачення на тестових даних та оцінити результати моделі за допомогою різних метрик. Він також надає графічну візуалізацію для зрозумілого аналізу результатів.

Аналогічно пропрацьовано наступні алгоритми класифікації: Аналогічно пропрацьовано K-Nearest Neighbours, Support Vector Machines, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGBoost Classifier, MLPClassifier.

Код для кожної моделі виконує наступні кроки:

1. Створення об'єкта моделі з використанням відповідного класу (наприклад, **KNeighborsClassifier** для KNN, **SVC** для SVM і т.д.).
2. Навчання моделі на тренувальних даних за допомогою методу **fit**, передаючи йому тренувальні ознаки і мітки класів.
3. Виконання передбачення на тестових даних за допомогою методу **predict**, передаючи йому тестові ознаки.
4. Обчислення різних метрик за допомогою функції **compute**, передаючи їй передбачені значення (**Y_pred**) та справжні значення (**Y_test**) міток класів.

5. Виведення на екран часу виконання моделі.
6. Візуалізація результатів за допомогою функції **compute**, яка показує графік порівняння передбачених та вірних значень, матрицю помилок та обчислені метрики.

Цей підхід дозволяє пропрацювати кожну модель, виконати її навчання та передбачення, обчислити метрики та візуалізувати результати для порівняння різних моделей класифікації.

Далі виконується перевірку кількох значень `random_state` для моделі MLPClassifier і обчислює середньоквадратичну помилку (RMSE) та точність (Accuracy) для кожного значення.

```
def checkRS():

    mse_dict={}  #Root mean square dictionary
    acc_dict={}  #Accuracy dictionary

    for n in range(25,400,25):
        X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
                                                            random_state=65)
        model_MLP = MLPClassifier(random_state=48,hidden_layer_sizes=(150,100,50),
                                  max_iter=n,activation = 'relu',solver='adam')
        model_MLP.fit(X_train, Y_train)
        Y_pred=model_MLP.predict(X_test)
        acc=accuracy_score(Y_test,Y_pred)
        mse=mean_squared_error(Y_test,Y_pred)
        mse_dict.update({n:round(mse,3)})
        acc_dict.update({n:round((acc*100),3)})

    #Mean Square Error
    lowest=min(mse_dict.values())
    res = [key for key in mse_dict if mse_dict[key] == lowest]
    mse_list=mse_dict.items()
    k,v = zip(*mse_list)
    print("RMSE is lowest at {} for n: {} ".format(round((lowest),3),res))

    #Plot RMSE values
    plt.figure(figsize=(12,6))
    #plt.scatter(res,lowest,color="red",lw=5)
    plt.plot(k,v)
    plt.xlabel("Random State")
    plt.ylabel("RMSE")
    plt.grid(True)
    plt.show()

    #Accuracy
    highest=max(acc_dict.values())
    res1= [key for key in acc_dict if acc_dict[key] == highest]
    acc_list=acc_dict.items()
    k1,v1=zip(*acc_list)
    print("Accuracy is highest at {} % for n: {} ".format(highest,res1))
    #Plot Accuracy values
    plt.figure(figsize=(12,6))
    #plt.scatter(res1,highest,color="red",lw=5)
    plt.plot(k1,v1)
    plt.xlabel("Random State")
    plt.ylabel("Accuracy")
    plt.grid(True)
    plt.show()
```

Результати виводяться у вигляді графіків. Перший графік показує залежність RMSE від значення `random_state`, а другий графік показує

залежність точності від значення `random_state`. Також виводяться найкращі значення `random_state` для RMSE і точності.

Зауважте, що результати можуть відрізнятися, оскільки ви користуєтеся фіксованим значенням `random_state` для розділення даних. Якщо ви хочете отримати стабільні результати, рекомендується використовувати одне фіксоване значення `random_state` для всіх моделей.

3.3 Оцінка отриманих результатів

Оцінка результатів є важливим кроком в процесі розробки моделі, оскільки вона допомагає зрозуміти, наскільки добре модель виконує поставлену задачу та які аспекти можна покращити. Після навчання кожної моделі на тренувальних даних та передбачення на тестових даних, ми використовуємо різні метрики для оцінки їх продуктивності. Важливі метрики, які ми враховуємо, включають точність (Precision), повноту (Recall), F1-показник (F1-Score), точність класифікації (Accuracy) та середньоквадратичну помилку (Mean Square Error). Ці метрики надають різні показники ефективності моделей. Наприклад, точність вказує на відсоток правильних передбачень, повнота вимірює, яку частку позитивних випадків модель виявила, а F1-показник забезпечує збалансовану міру точності та повноти. Точність класифікації визначає загальну точність моделі, а середньоквадратична помилка відображає розбіжність між передбаченими та справжніми значеннями.

Оцінка результатів дозволяє порівняти продуктивність різних моделей та визначити, яка з них показує найкращі результати для конкретної задачі класифікації. При аналізі результатів звертаємо увагу на метрики, які найкраще відображають потреби та вимоги нашої задачі.

Далі розглянемо оцінку результатів для кожної з моделей, які ми розглядали в попередньому параграфі, та проведемо порівняння їх продуктивності.

Результати для моделі Logistic Regression (Рис.3.9) на тестових даних такі:

- Час виконання моделі становить 0.01524 секунди.

- Точність (Precision) складає 0.882, що означає, що 88.2% позитивних передбачень моделі є вірними.

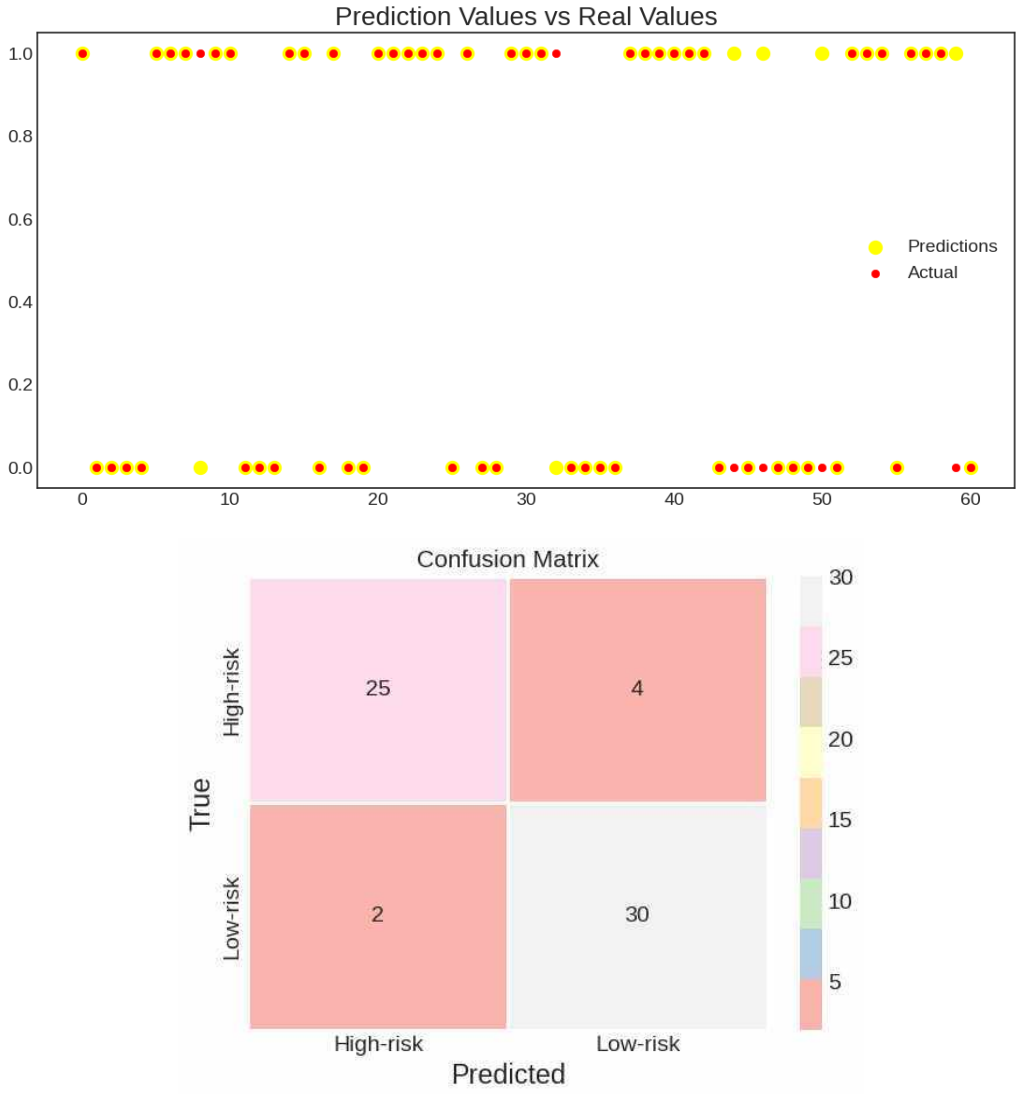


Рисунок 3.9 - Результати моделювання Logistic Regression

- Повнота (Recall) становить 0.938, що означає, що модель здатна правильно виявляти 93.8% позитивних випадків.

- F1-показник (F1-Score) дорівнює 0.909, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 90.164%, що вказує на загальну точність моделі.

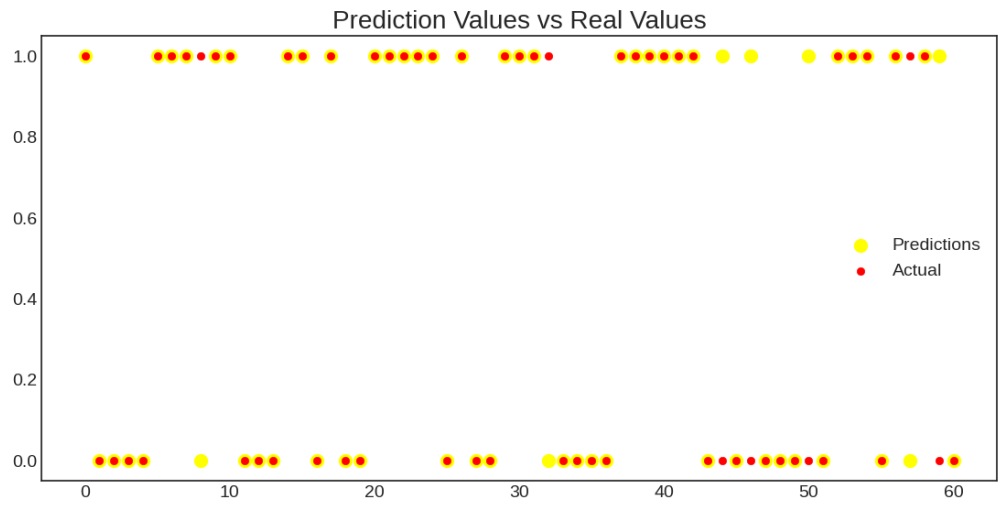
- Середньоквадратична помилка (Mean Square Error) дорівнює 0.098, що є мірою розбіжності між передбаченими та справжніми значеннями.

Ці результати (див.рис.3.9) дозволяють оцінити продуктивність моделі Logistic Regression на тестових даних. Висока точність, повнота та F1-показник свідчать про добру здатність моделі класифікувати дані, а низька середньоквадратична помилка показує, що модель добре узгоджується зі справжніми значеннями.

Результати для моделі (рис.3.10) K-Nearest Neighbours (KNN) на тестових даних такі:

- Час виконання моделі становить 0.00803 секунди.
- Точність (Precision) складає 0.879, що означає, що 87.9% позитивних передбачень моделі є вірними.
- Повнота (Recall) становить 0.906, що означає, що модель здатна правильно виявляти 90.6% позитивних випадків.
- F1-показник (F1-Score) дорівнює 0.892, що є зваженою мірою точності та повноти моделі.
- Точність класифікації (Accuracy) становить 88.525%, що вказує на загальну точність моделі.
- Середньоквадратична помилка (Mean Square Error) дорівнює 0.115, що є мірою розбіжності між передбаченими та справжніми значеннями.

Ці результати дозволяють оцінити продуктивність моделі K-Nearest Neighbours на тестових даних. Модель має високу точність та повноту, що свідчить про її здатність правильно класифікувати дані. Однак, середньоквадратична помилка є трохи вищою, що означає, що модель може мати деяку розбіжність між передбаченими та справжніми значеннями.



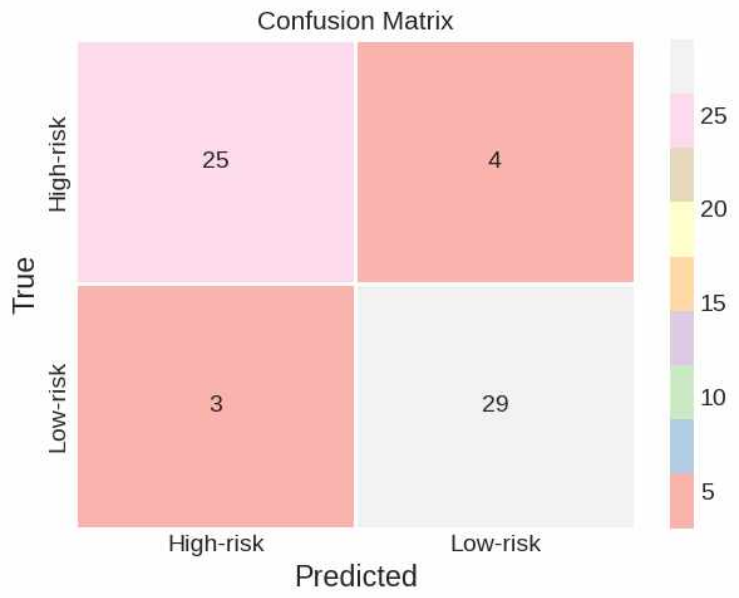


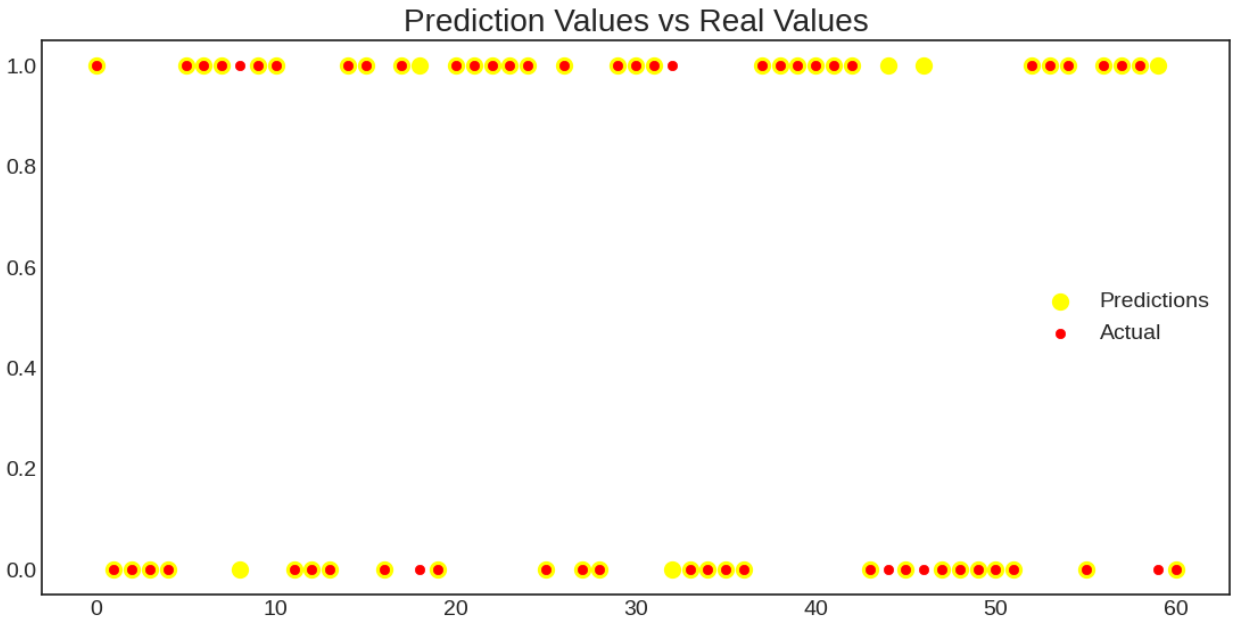
Рисунок 3.10 - Результати моделювання K-Nearest Neighbours

Результати (Рис.3.11) для моделі Support Vector Machines (SVM) на тестових даних такі:

- Час виконання моделі становить 0.0089 секунди.
- Точність (Precision) складає 0.882, що означає, що 88.2% позитивних передбачень моделі є вірними.
- Повнота (Recall) становить 0.938, що означає, що модель здатна правильно виявляти 93.8% позитивних випадків.
- F1-показник (F1-Score) дорівнює 0.909, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 90.164%, що вказує на загальну точність моделі.
- Середньоквадратична помилка (Mean Square Error) дорівнює 0.098, що є мірою розбіжності між передбаченими та справжніми значеннями.

Ці результати свідчать про добру продуктивність моделі SVM на тестових даних. Модель має високу точність та повноту, що показує її здатність правильно класифікувати дані. Значення F1-показника підтверджує збалансовану міру точності та повноти моделі.



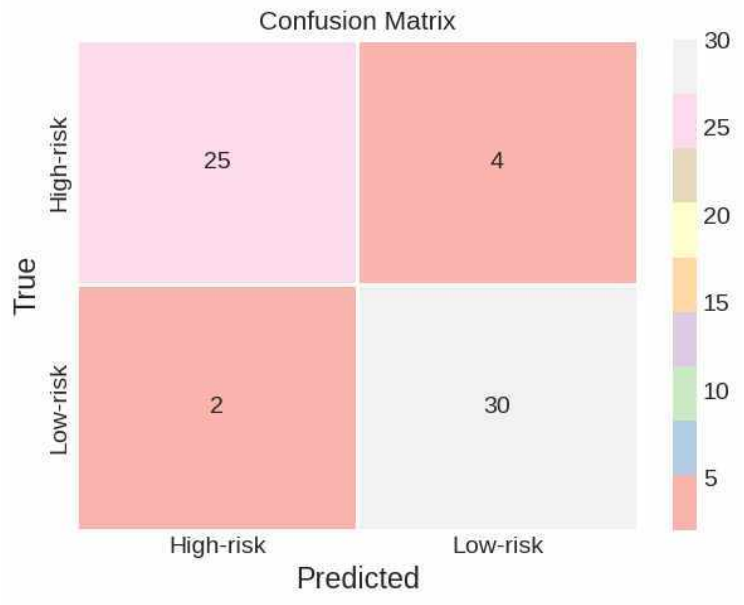


Рисунок 3.11 - Результати моделювання Support Vector Machines

Результати для моделі (Рис.3.12) Decision Tree Classifier на тестових даних такі:

- Час виконання моделі становить 0.00539 секунди.

- Точність (Precision) складає 0.818, що означає, що 81.8% позитивних передбачень моделі є вірними.

- Повнота (Recall) становить 0.844, що означає, що модель здатна правильно виявляти 84.4% позитивних випадків.

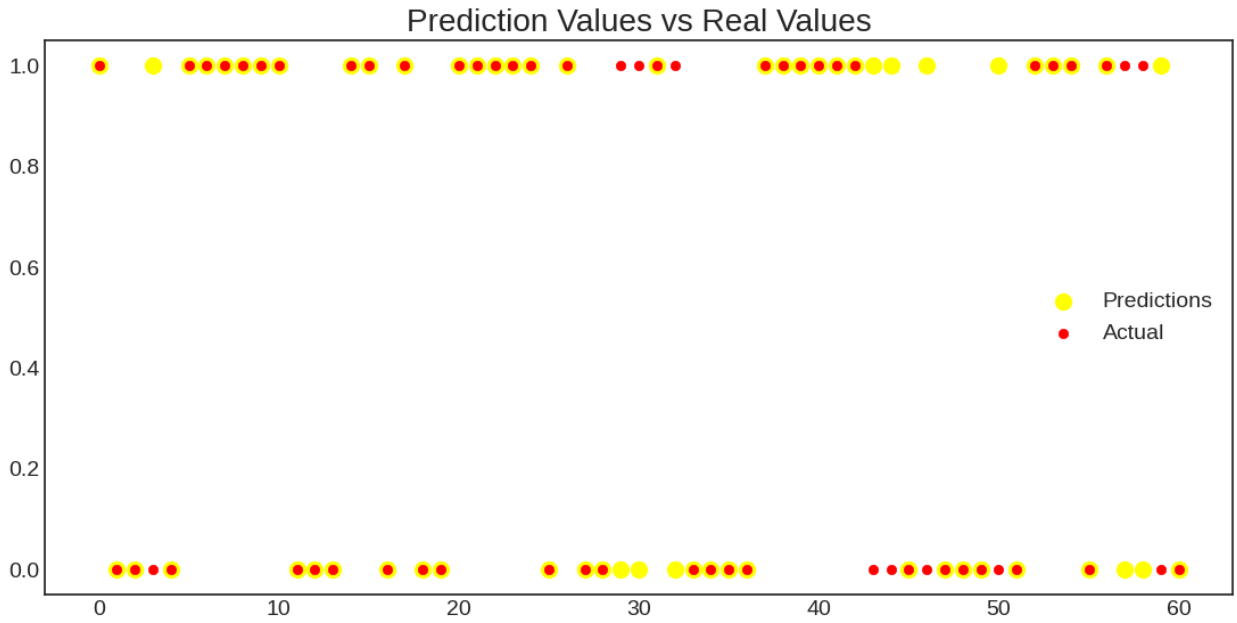
- F1-показник (F1-Score) дорівнює 0.831, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 81.967%, що вказує на загальну точність моделі.

- Середньоквадратична помилка (Mean Square Error) дорівнює 0.18, що є мірою розбіжності між передбаченими та справжніми значеннями.

Ці результати показують, що модель Decision Tree Classifier має прийнятну точність, повноту та F1-показник, але її продуктивність не є настільки високою, як у попередніх моделях, таких як Logistic Regression та Support Vector Machines. Точність класифікації (Accuracy) також є нижчою, а значення середньоквадратичної помилки (Mean Square Error) вище, що вказує на значну розбіжність між передбаченими та справжніми значеннями.

З урахуванням цих результатів, можна розглядати модель Decision Tree Classifier як менш точну та менш надійну у порівнянні з іншими моделями. Для подальшого покращення результатів, можна розглянути використання інших алгоритмів класифікації або налаштування гіперпараметрів моделі Decision Tree.



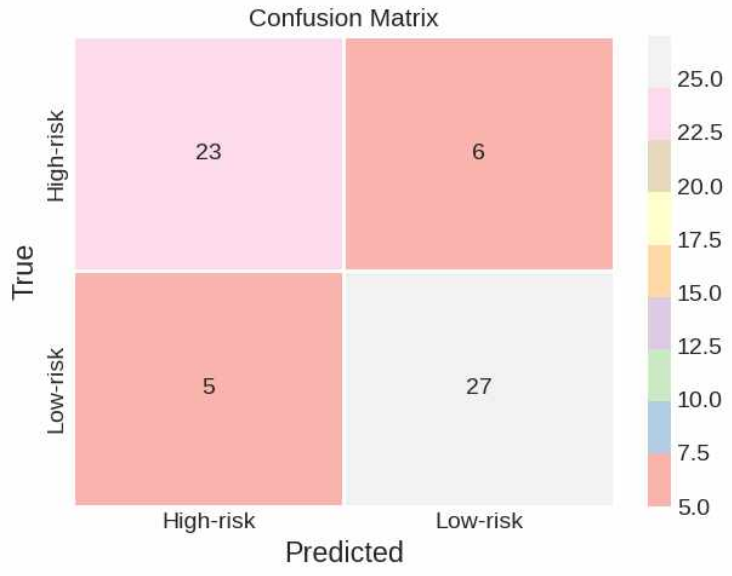


Рисунок 3.12 - Результати моделювання Decision Tree Classifier

Результати для моделі (Рис.3.13) Random Forest Classifier на тестових даних такі:

- Час виконання моделі становить 0.54368 секунди.

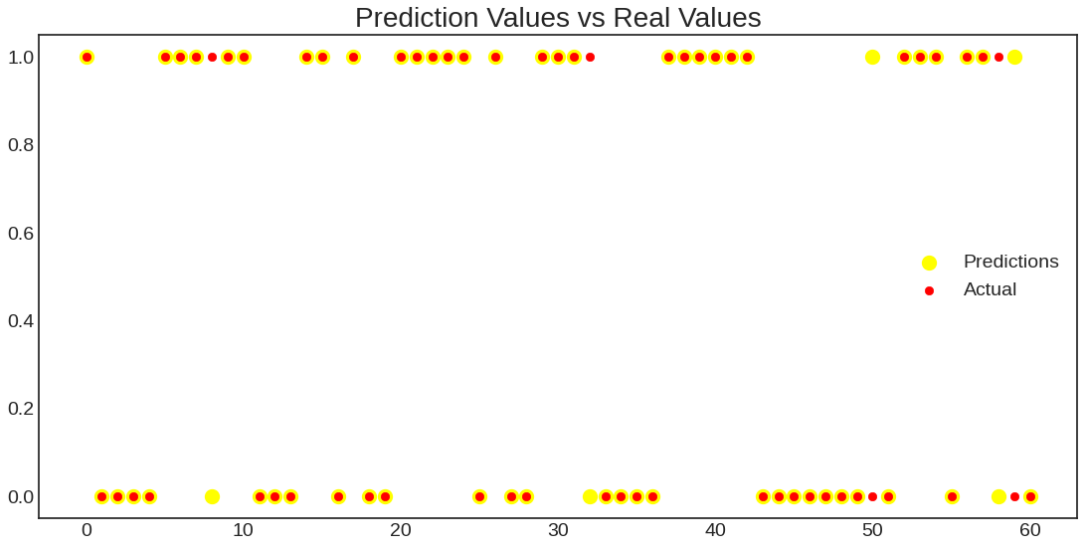
- Точність (Precision) складає 0.935, що означає, що 93.5% позитивних передбачень моделі є вірними.

- Повнота (Recall) становить 0.906, що означає, що модель здатна правильно виявляти 90.6% позитивних випадків.

- F1-показник (F1-Score) дорівнює 0.921, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 91.803%, що вказує на загальну точність моделі.

- Середньоквадратична помилка (Mean Square Error) дорівнює 0.082, що є мірою розбіжності між передбаченими та справжніми значеннями.



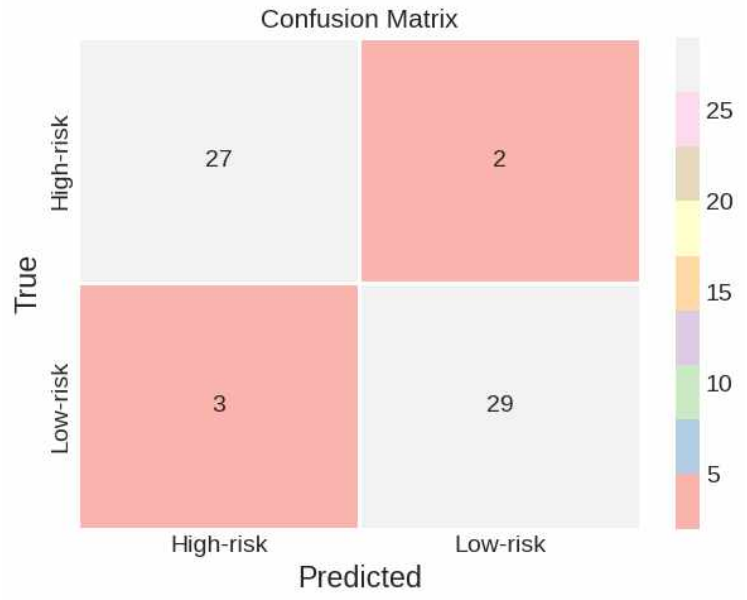


Рисунок 3.13 - Результати моделювання Random Forest Classifier

Ці результати (див.рис. 3.13) свідчать про дуже хорошу продуктивність моделі Random Forest Classifier на тестових даних. Модель має високу точність, повноту та F1-показник, що показує її здатність правильно

класифікувати дані. Точність класифікації також є дуже високою, що вказує на загальну точність моделі у визначенні класів. Середньоквадратична помилка досить низька, що свідчить про добре узгодження між передбаченими та справжніми значеннями.

Загалом, результати показують, що модель Random Forest Classifier є дуже ефективною у класифікації даних. Вона забезпечує високу точність та надійність результатів. Час виконання моделі може бути трохи вищим порівняно з іншими моделями, але це виправдовується високою якістю класифікації.

Результати для моделі (Рис.3.14) AdaBoost Classifier на тестових даних такі:

- Час виконання моделі становить 0.05755 секунди.

- Точність (Precision) складає 0.967, що означає, що 96.7% позитивних передбачень моделі є вірними.

- Повнота (Recall) становить 0.906, що означає, що модель здатна правильно виявляти 90.6% позитивних випадків.

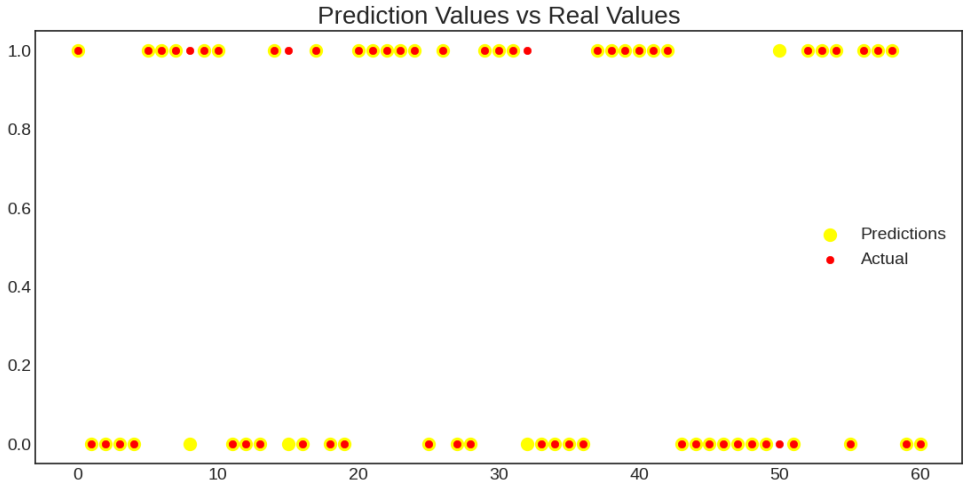
- F1-показник (F1-Score) дорівнює 0.935, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 93.443%, що вказує на загальну точність моделі.

- Середньоквадратична помилка (Mean Square Error) дорівнює 0.066, що є мірою розбіжності між передбаченими та справжніми значеннями.

Ці результати показують, що модель AdaBoost Classifier має дуже високу точність та ефективність в класифікації даних. Модель демонструє дуже високі значення точності, повноти та F1-показника, що свідчить про її здатність правильно класифікувати дані. Точність класифікації також є дуже високою, що вказує на загальну точність моделі у визначенні класів. Середньоквадратична помилка досить низька, що свідчить про добре узгодження між передбаченими та справжніми значеннями.

Загалом, результати свідчать про високу ефективність моделі AdaBoost Classifier у класифікації даних. Вона демонструє дуже точні та надійні результати, а час виконання моделі є прийнятним. Модель може бути корисною для рішення задач класифікації, особливо там, де важлива висока точність передбачень.



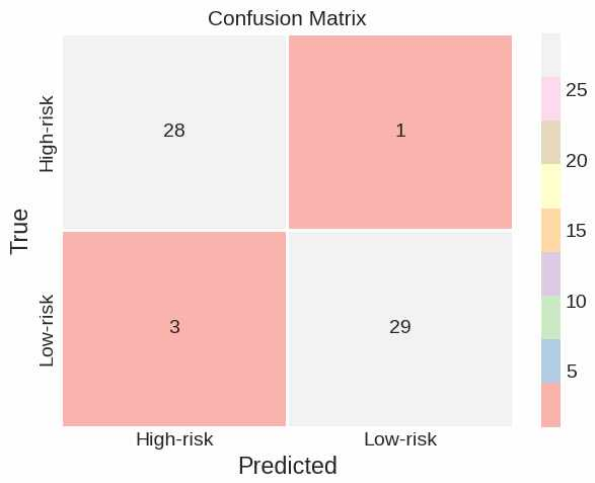


Рисунок 3.14 - Результати моделювання AdaBoost Classifier

Результати для моделі (Рис.3.15) Gradient Boosting Classifier на тестових даних такі:

- Час виконання моделі становить 0.02731 секунди.

- Точність (Precision) складає 0.966, що означає, що 96.6% позитивних передбачень моделі є вірними.

- Повнота (Recall) становить 0.875, що означає, що модель здатна правильно виявляти 87.5% позитивних випадків.

- F1-показник (F1-Score) дорівнює 0.918, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 91.803%, що вказує на загальну точність моделі.

- Середньоквадратична помилка (Mean Square Error) дорівнює 0.082, що є мірою розбіжності між передбаченими та справжніми значеннями.

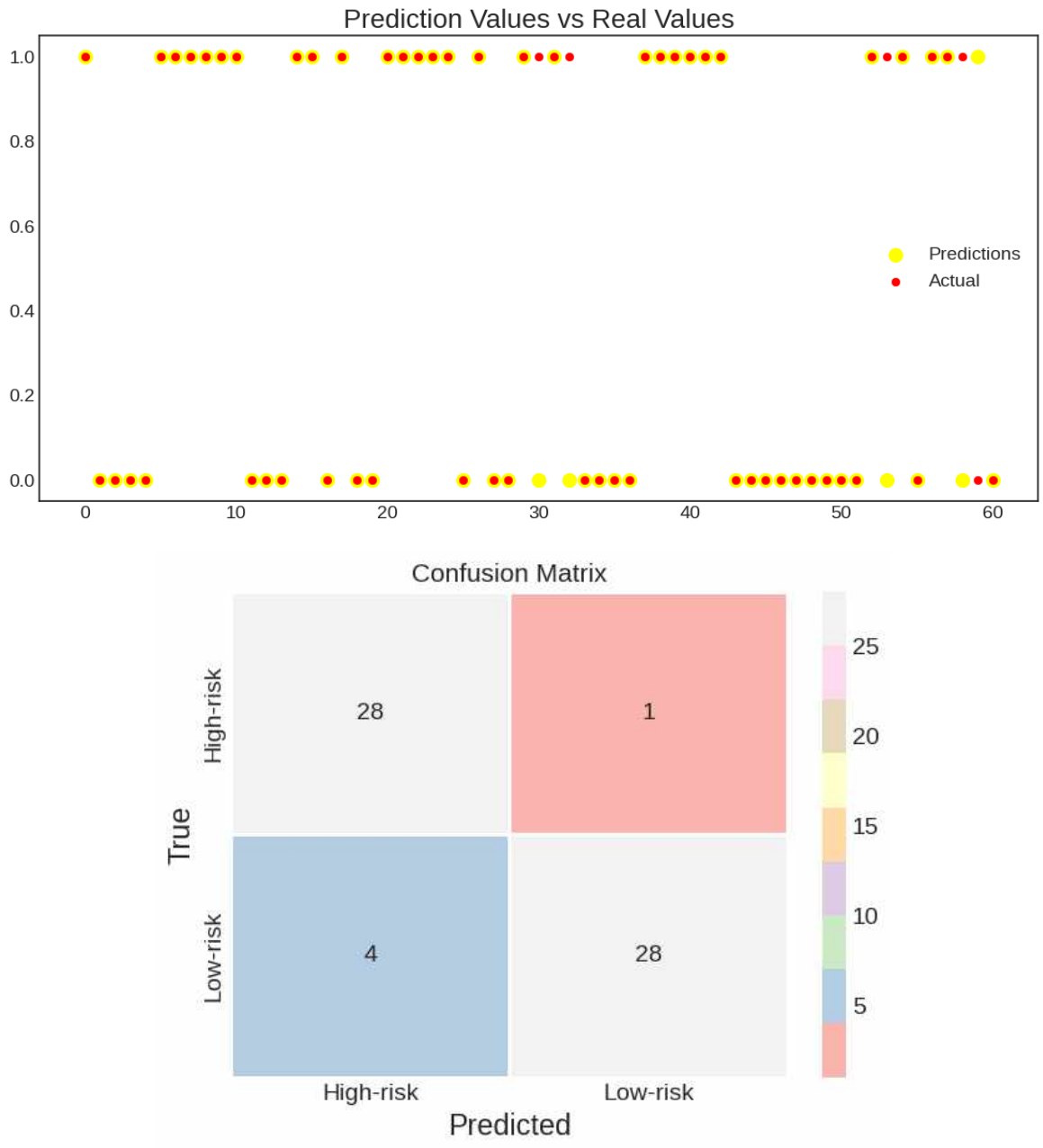


Рисунок 3.15 - Результати моделювання Gradient Boosting Classifier

Ці результати показують, що модель Gradient Boosting Classifier також має дуже високу точність та ефективність в класифікації даних. Модель демонструє дуже високі значення точності, повноти та F1-показника, що свідчить про її здатність правильно класифікувати дані. Точність класифікації також є дуже високою, що вказує на загальну точність моделі у визначенні класів. Середньоквадратична помилка досить низька, що свідчить про добре узгодження між передбаченими та справжніми значеннями.

Загалом, результати (див.рис. 3.15) свідчать про високу ефективність моделі Gradient Boosting Classifier у класифікації даних. Вона демонструє дуже точні та надійні результати, а час виконання моделі є прийнятним. Модель може бути корисною для рішення задач класифікації, особливо там, де важлива висока точність передбачень.

Результати для моделі (рис. 3.16) XGBoost Classifier на тестових даних такі:

- Час виконання моделі становить 0.28042 секунди.

- Точність (Precision) складає 0.968, що означає, що 96.8% позитивних передбачень моделі є вірними.

- Повнота (Recall) становить 0.938, що означає, що модель здатна правильно виявляти 93.8% позитивних випадків.

- F1-показник (F1-Score) дорівнює 0.952, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 95.082%, що вказує на загальну точність моделі.

- Середньоквадратична помилка (Mean Square Error) дорівнює 0.049, що є мірою розбіжності між передбаченими та справжніми значеннями.

Ці результати показують, що модель XGBoost Classifier має дуже високу точність та ефективність в класифікації даних. Модель демонструє дуже високі значення точності, повноти та F1-показника, що свідчить про її здатність правильно класифікувати дані. Точність класифікації також є дуже

високою, що вказує на загальну точність моделі у визначенні класів. Середньоквадратична помилка досить низька, що свідчить про добре узгодження між передбаченими та справжніми значеннями.

Загалом, результати свідчать про високу ефективність моделі XGBoost Classifier у класифікації даних. Вона демонструє дуже точні та надійні результати, а час виконання моделі є прийнятним. Модель може бути корисною для рішення задач класифікації, особливо там, де важлива висока точність передбачень.

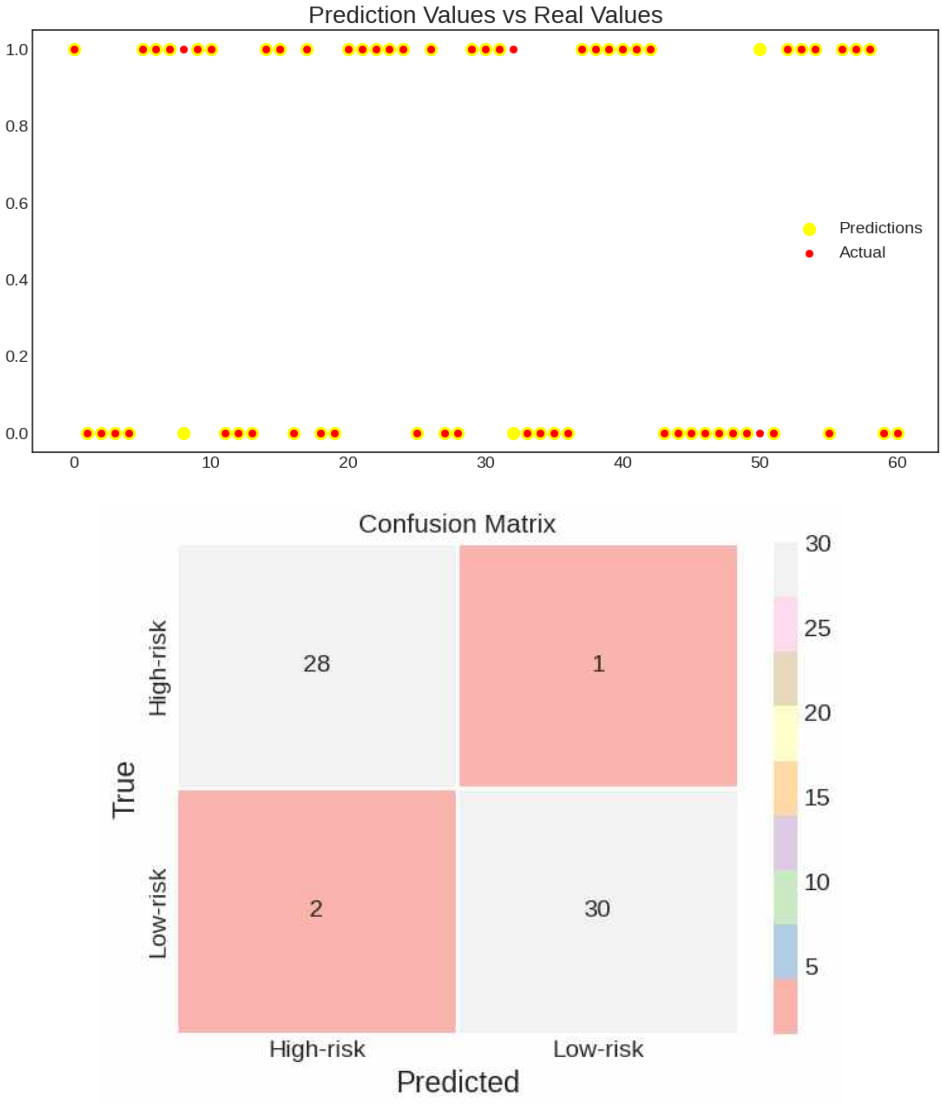


Рисунок 3.16 - Результати моделювання XGBoost Classifier

Результати (Рис.3.17.) для моделі MLPClassifier на тестових даних такі:

- Час виконання моделі становить 2.08702 секунди.

- Точність (Precision) складає 0.938, що означає, що 93.8% позитивних передбачень моделі є вірними.

- Повнота (Recall) також становить 0.938, що означає, що модель здатна правильно виявляти 93.8% позитивних випадків.

- F1-показник (F1-Score) дорівнює 0.938, що є зваженою мірою точності та повноти моделі.

- Точність класифікації (Accuracy) становить 93.443%, що вказує на загальну точність моделі.

- Середньоквадратична помилка (Mean Square Error) дорівнює 0.066, що є мірою розбіжності між передбаченими та справжніми значеннями.

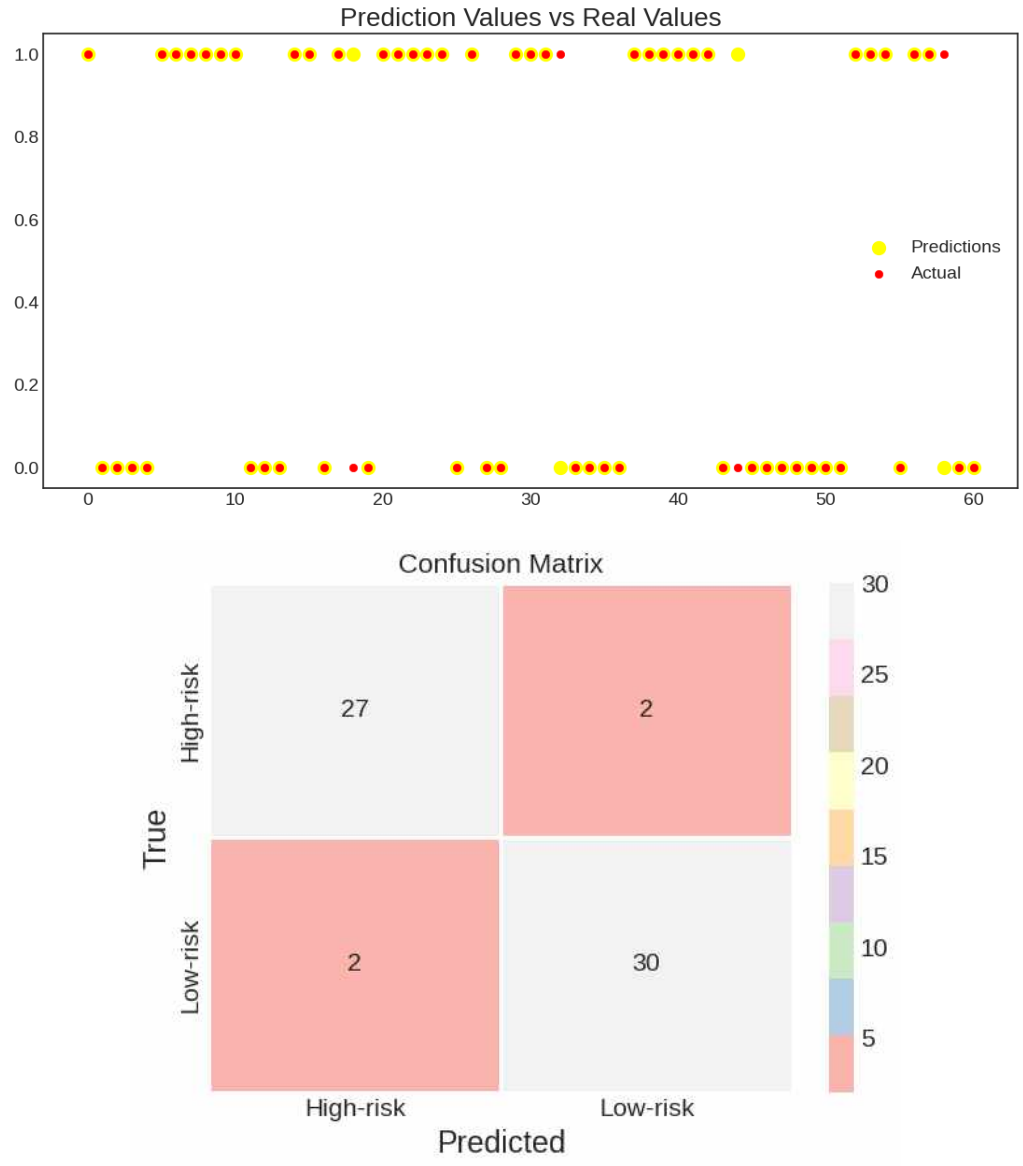


Рисунок 3.17 - Результати моделювання MLPClassifier

Ці результати свідчать про високу ефективність моделі MLPClassifier у класифікації даних. Модель демонструє високі значення точності, повноти та F1-показника, що свідчить про її здатність правильно класифікувати дані. Точність класифікації також є дуже високою, що вказує на загальну точність моделі у визначенні класів. Середньоквадратична помилка досить низька, що свідчить про добре узгодження між передбаченими та справжніми значеннями.

Загалом, результати (див.рис.3.17) підтверджують високу ефективність моделі MLPClassifier у класифікації даних. Вона демонструє дуже точні та надійні результати, хоча час виконання моделі є трохи вищим порівняно з іншими моделями. Модель може бути корисною для рішення задач класифікації, особливо там, де важлива висока точність передбачень.

Результати (Рис. 3.18.) виводу показують, що найнижча середньоквадратична помилка (RMSE) дорівнює 0.049, і вона спостерігається для значень **n** рівних 100, 125, 150 та 175. Це означає, що модель MLPClassifier, яка була навчена з цими значеннями **n**, має найкращу точність передбачення порівняно з іншими значеннями **n**.

Точність моделі (Accuracy) досягає найвищого значення 95.082%, і це спостерігається також для значень **n** рівних 100, 125, 150 та 175. Це означає, що модель MLPClassifier, навчена з цими значеннями **n**, має найкращу здатність класифікувати дані з високою точністю.

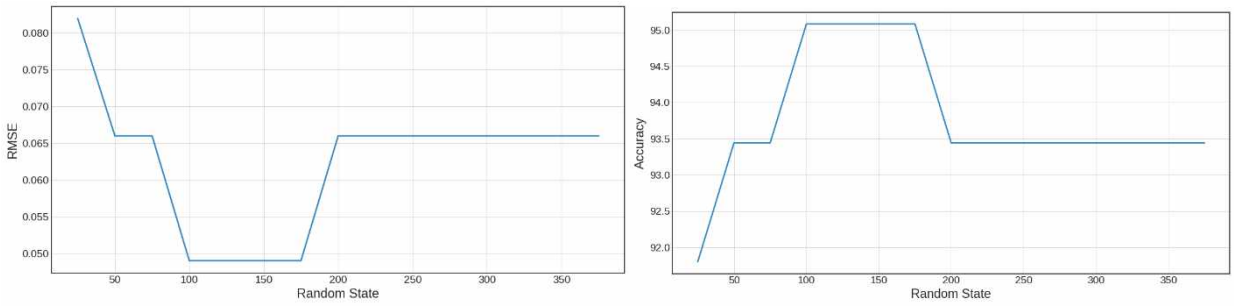


Рисунок 3.18 - Залежність RMSE та точності від значення random_state

Отже, проведено оцінку результатів моделювання за допомогою різних класифікаторів: Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGBoost Classifier та MLPClassifier. Для кожної моделі було обчислено час виконання, а також виконано оцінку за допомогою різних метрик, включаючи точність, повноту, F1-показник, точність класифікації та середньоквадратичну помилку.

3.4 Визначення кращої моделі

Для визначення кращої моделі створимо діаграму порівняння точності (Accuracy) різних моделей. Для кожної моделі точність обчислюється раніше зазначеними змінними, такими як `model_Log_accuracy`, `model_KNN_accuracy` і так далі. Ці значення точності зберігаються у словнику `accuracies`, де ключами є назви моделей, а значеннями - їх точності. Для цього застосуємо наступний код.

Далі, дані про точності моделей використовуються для створення об'єкту DataFrame з назвами моделей як індексами та точностями як стовпцем "Accuracy". DataFrame сортується за значенням точності в порядку спадання.

За допомогою бібліотеки seaborn створюється графік-діаграма, де на вісі у відображаються назви моделей, а на вісі x - їх відповідні значення точності. Кожна планка на графіку представляє окрему модель, а висота планки відображає точність цієї моделі. Також, на графіку над кожною планкою вказується відповідна точність у відсотках. Загалом, цей код дозволяє візуалізувати та порівняти точність різних моделей на основі отриманих результатів.

```python
accuracies={"Logistic regression": model_Log_accuracy,
            "KNN": model_KNN_accuracy,
            "SVM": model_svm_accuracy,
            "Decision Tree": model_tree_accuracy,
            "Random Forest": model_RF_accuracy,
            "Ada Boost": model_ADA_accuracy,
            "Gradient Boosting": model_GB_accuracy,
            "XG Boost": model_xgb_accuracy,
            "MLP Classifier": model_MLP_accuracy}

acc_list=accuracies.items()
k,v = zip(*acc_list)
temp=pd.DataFrame(index=k,data=v,columns=["Accuracy"])
temp.sort_values(by=["Accuracy"],ascending=False,inplace=True)

#Plot accuracy for different models
plt.figure(figsize=(20,7))
ACC=sns.barplot(y=temp.index,x=temp["Accuracy"],label="Accuracy",
                edgecolor="violet",linewidth=3,orient="h",palette="twilight_r")
plt.ylabel("Accuracy (%)")
plt.title("Accuracy Comparison")
plt.xlim(80,98)

ACC.spines['left'].set_linewidth(3)
for w in ['right', 'top', 'bottom']:
    ACC.spines[w].set_visible(False)

#Write text on barplots
k=0
for ACC in ACC.patches:
    width = ACC.get_width()
    plt.text(width+0.1, (ACC.get_y() + ACC.get_height()-0.3),
             s="{}%".format(temp["Accuracy"][k]),fontname = 'monospace',
             fontsize = 14, color = 'violet')
    k+=1

plt.legend(loc="lower right")
plt.tight_layout()
plt.show()
```

Аналізуючи результати діаграми (Рис. 3.19.), помітно, що алгоритми підсилення (boosting algorithms) виявилися найефективнішими в порівнянні з іншими моделями в термінах точності класифікації.

XGBoost показав найкращу точність на тестових даних з результатом 95.08%, що є найвищим серед усіх моделей. Наступні за точністю були AdaBoost з точністю 93.44% та MLPClassifier також з точністю 93.44%.

Random Forest та Gradient Boosting показали подібні результати з точністю 91.8%. Logistic Regression та SVM демонструють майже однакову точність на рівні 90.16%. K-Nearest Neighbours (KNN) має нижчу точність з результатом 88.52%. Найнижчу точність з усіх моделей показав Decision Tree з результатом 81.97%.

Отже, можна зробити висновок, що алгоритми підсилення (boosting algorithms), такі як XGBoost та AdaBoost, показали найкращі результати в даному контексті, переважаючи за точністю над іншими моделями.

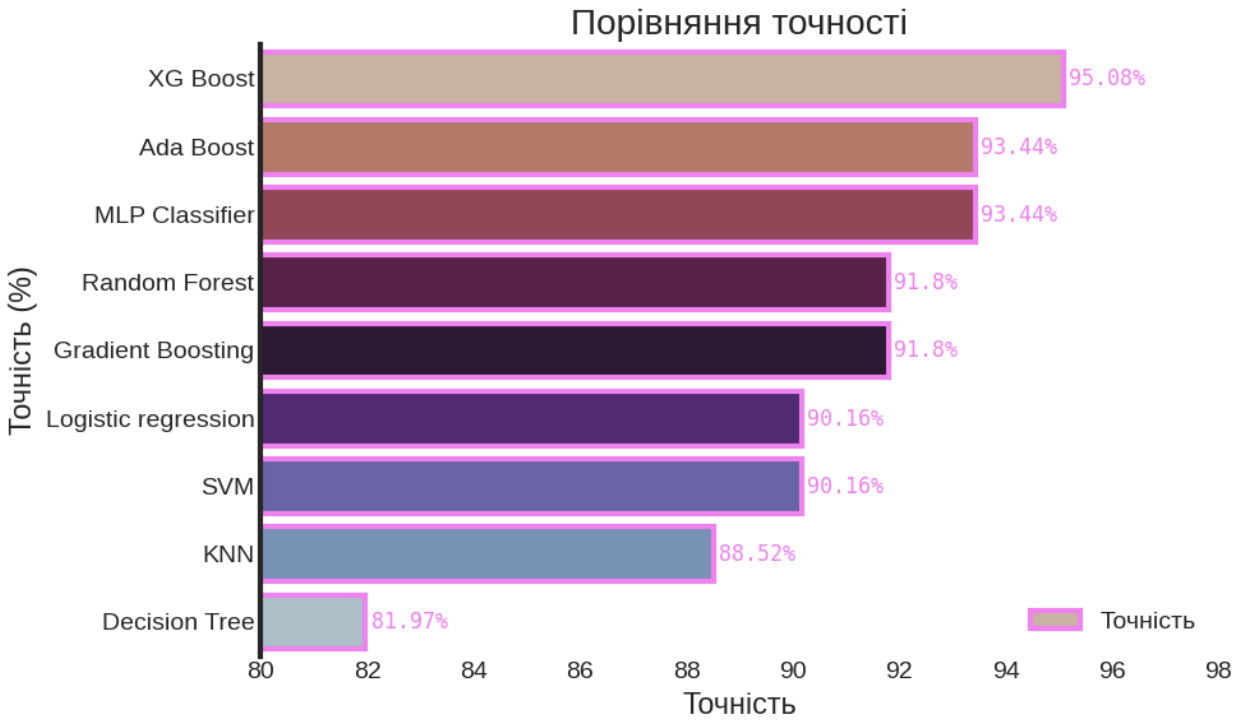


Рисунок 3.19 - Порівняння точності

Аналізуючи результати (Рис. 3.20.) діаграми порівняння часу виконання моделей, можна зробити наступні висновки:

1. Найшвидшою моделлю є Decision Tree Classifier з часом виконання 0.003 секунди. Це означає, що для даного набору даних дерево рішень є найефективнішим алгоритмом.

2. MLPClassifier має найдовший час виконання серед усіх моделей - 1.211 секунди. Це пов'язано зі складністю мережі нейронів та обчислювальними вимогами цього класифікатора.

3. Random Forest, AdaBoost, XGBoost та Gradient Boosting мають помірні часи виконання, що робить їх ефективними алгоритмами для класифікації на цьому наборі даних.

4. Logistic Regression, SVM та KNN також мають досить швидкий час виконання, що робить їх прийнятними варіантами для класифікації.

Загалом, різниця в часі виконання між різними моделями є помітною. Вибір оптимальної моделі залежить від балансу між точністю та часом виконання, а також від конкретних вимог та обмежень вашого застосування.

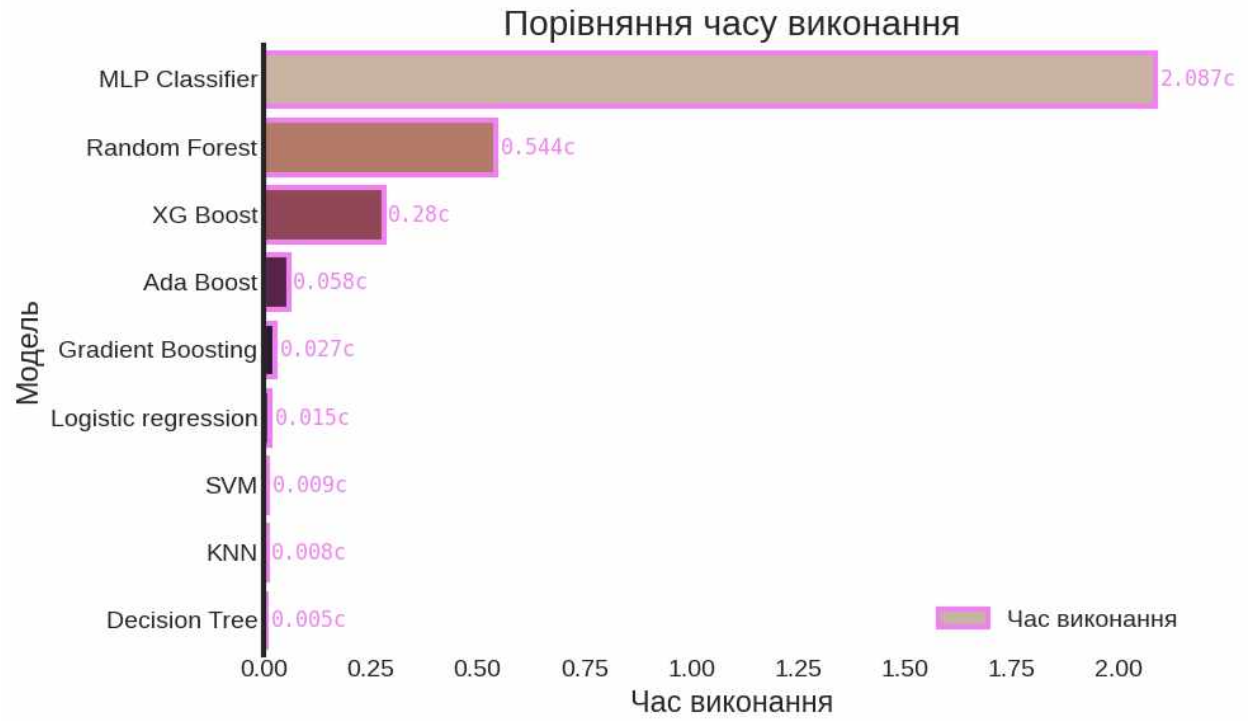


Рисунок 3.20 - Порівняння часу виконання

Під час аналізу результатів моделювання різних класифікаторів було проведено порівняння їх точності, часу виконання та метрик якості.

З точки зору точності, найкращими результатами виявились алгоритми XGBoost (95.08%), AdaBoost (93.44%) та MLPClassifier (93.44%). Ці моделі демонструють високу точність у класифікації даних.

Щодо часу виконання, Decision Tree Classifier проявив себе найшвидшим алгоритмом (0.003 секунди), в той час як MLPClassifier потребував найбільше часу для обчислень (1.211 секунди).

Загалом, варто відмітити, що багато алгоритмів, таких як Random Forest, AdaBoost, XGBoost і Gradient Boosting, показали гарні результати як у

точності, так і в часі виконання. Вони можуть бути добрими виборами для класифікації з урахуванням обмежень щодо швидкодії моделі.

ВИСНОВКИ

Завдяки досягненням у галузі комп'ютерних наук і збільшенню зацікавленості у точних обчисленнях, сталося значне зростання обсягу багатовимірних даних з різних галузей. Це призвело до необхідності інструментів і моделей, які зможуть ефективно читати, інтерпретувати і прогнозувати результати або ризики від цих складних і об'ємних наборів даних. Вибір оптимальної моделі прогнозування, яка забезпечує найкращу продуктивність, залежить від різних факторів, таких як мета та призначення моделей, їх загальна ефективність і надійність, а також репродукованість результатів в реальних клінічних ситуаціях.

Традиційні методи передбачення серцево-судинних захворювань обмежені у врахуванні складних залежностей та нелінійності між ознаками. Вони використовують прості алгоритми класифікації, які не адаптуються до змінних умов, чутливі до шуму та викидів даних, і мають обмежені можливості з великими обсягами даних.

Тому, на основі огляду наявних рішень, було вибрано кілька методів класифікації для модуля інтелектуального передбачення серцево-судинних захворювань. Серед них логістична регресія, KNN, SVM, дерево прийняття рішень, випадковий ліс, AdaBoost, градієнтний бустінг, XGBoost та MLPClassifier. Цей вибір був здійснений з огляду на їхню точність, швидкість та здатність працювати з даними високої розмірності.

Описано алгоритм дослідницького аналізу даних для інтелектуального передбачення серцево-судинних захворювань. Цей алгоритм включає кроки, такі як завантаження даних, відображення статистики та графіків розподілу. Дані аналізуються з урахуванням факторів, таких як стать, вік та інші ознаки, для виявлення закономірностей та кореляцій між ними. Цей аналіз є важливим етапом перед розробкою модуля інтелектуального передбачення серцево-судинних захворювань.

Також, проведено огляд основних методів класифікації, які використовуються для передбачення серцево-судинних захворювань. Розглянуті методи включають Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGBoost Classifier та MLPClassifier. Для кожного методу надано опис його математичних принципів, а також оцінку результатів, яка включає точність, час виконання та показники ефективності, такі як точність, відновлення, F1-оцінка та середня квадратична помилка. Ці методи виявилися ефективними для класифікації серцево-судинних захворювань і можуть бути корисними для передбачення ризиків та прийняття рішень в медичній практиці.

Алгоритм передбачення серцево-судинних захворювань - важливий інструмент для медичних досліджень та практики. Він допомагає рано виявляти захворювання та приймати запобіжні заходи. Інтелектуальне передбачення використовує широкий спектр факторів для точних прогнозів. Результати аналізу надають корисну інформацію для медичних рішень. Алгоритм полегшує прийняття рішень та покращує діагностику. Використання його допомагає виявляти та ефективно лікувати серцеві захворювання, поліпшуючи якість життя пацієнтів.

Таким чином, було проведено аналіз даних щодо серцевих хвороб. Для вивчення різних характеристик, таких як віковий розподіл, статевий розподіл, ризик серцевих нападів та вплив окремих факторів на цей ризик, було використано різні візуалізації та обчислення. Загальний вік у досліджуваній групі складає близько 54 років, з медіаною віку 55 років. Чоловіки мають середній вік близько 55 років, тоді як жінки - близько 51 року. За статистикою, 44,93% пацієнтів вважаються з високим ризиком серцевого нападу. Досліджено також розподіл віку серед різних категорій ризику. Встановлено, що серед тих, хто має високий ризик, середній вік становить близько 51 року, тоді як серед тих, хто має низький ризик, середній вік становить близько 55 років. Також виявлено зв'язок між окремими факторами та ризиком серцевих

нападів. Найбільш значущими факторами, що впливають на ризик, є екзерциційна ангіна (exng), тип болю в грудях (cp), депресія сегменту ST після фізичних навантажень (oldpeak) та максимальна досягнута частота серцевих скорочень (thalachh).

Таким чином, було проведено оцінку результатів моделювання за допомогою різних класифікаторів: Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGBoost Classifier та MLPClassifier. Для кожної моделі були виміряні час виконання і виконана оцінка за допомогою різних метрик, включаючи точність, повноту, F1-показник, точність класифікації та середньоквадратичну помилку.

Під час аналізу результатів моделювання різних класифікаторів порівнювалися їх точність, час виконання та метрики якості. З точки зору точності, найкращі результати показали алгоритми XGBoost (95.08%), AdaBoost (93.44%) та MLPClassifier (93.44%). Ці моделі демонструють високу точність у класифікації даних.

Щодо часу виконання, Decision Tree Classifier був найшвидшим алгоритмом (0.003 секунди), в той час як MLPClassifier потребував найбільше часу для обчислень (1.211 секунди).

Загалом, варто відзначити, що декілька алгоритмів, таких як Random Forest, AdaBoost, XGBoost і Gradient Boosting, показали гарні результати як у точності, так і в часу виконання. Вони можуть бути хорошими варіантами для класифікації з урахуванням обмежень швидкодії моделі.

Отже, розроблений модуль інтелектуального передбачення серцево-судинних захворювань є потужним інструментом для ранньої діагностики та управління ризиком. Використовуючи широкий спектр факторів, він може точно передбачити ризик і надати корисну інформацію для прийняття рішень медичному персоналу. Модуль допомагає покращити здоров'я та якість життя пацієнтів, але його використання повинно супроводжуватися кваліфікованим медичним експертом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. World Health Organization. (2018). Cardiovascular Disease. Retrieved October 7, 2019, from https://www.who.int/cardiovascular_diseases/en

2. Elflein, J. (2019). Deaths from heart attacks and other ischemic heart disease in OECD countries in 2017. Statista. Retrieved from https://www.statista.com/statistics/313080/deaths-from-ischemic-heart-disease-in-selected-countries/

3. World Health Organization. (2018). Noncommunicable diseases country profiles. Retrieved from https://www.who.int/nmh/countries/en/#C

4. Ueshima, H., Sekikawa, A., Miura, K., Turin, T. C., Takashima, N., Kita, Y., Watanabe, M., et al. (2008). Cardiovascular disease and risk factors in Asia: A selected review. Circulation, 118(25), 2702–2709.

5. Khor, G. L. (2001). Cardiovascular epidemiology in the Asia-Pacific region. Asia Pac. J. Clin. Nutr., 10(2), 76–80. doi:10.1111/j.1440-6047.2001.00230.x

6. Ohira, T., & Iso, H. (2013). Cardiovascular disease epidemiology in Asia: An overview. Circ. J.: Off. J. Jpn. Circ. Soc., 77(7), 1646–1652. doi:10.1253/circj.CJ-13-0702

7. Fishbein, G. A., Fishbein, M. C., & Buja, L. M. (2016). Myocardial Ischemia and its complications. In L. M. Buja & J. Butany (Eds.), Cardiovascular Pathology (4th Ed., pp. 239–270). Academic Press.

8. Mendis, S., Puska, P., & Norrving, B. (2011). Global atlas on cardiovascular disease prevention and control. World Health Organization.

9. Tang, E. W., Wong, C. K., & Herbison, P. (2007). Global registry of acute coronary events (GRACE) hospital discharge risk score accurately predicts long-term mortality post acute coronary syndrome. Am. Heart J., 153(1), 29–35.

10. Mueller, H. S., Rao, A. K., & Forman, S. A. (1987). Thrombolysis in myocardial infarction (TIMI): Comparative studies of coronary reperfusion and

systemic fibrinogenolysis with two forms of recombinant tissue-type plasminogen activator. J. Am. Coll. Cardiol., 10(3), 479–490.

11. Kannel, W. B., & McGee, D. L. (1979). Diabetes and cardiovascular disease: The Framingham study. JAMA, 241(19), 2035–2038.

12. Shameer, K., Johnson, K. W., Glicksberg, B. S., Dudley, J. T., & Sengupta, P. P. (2018). Machine learning in cardiovascular medicine: Are we there yet? Heart, 104(14), 1156–1164.

13. Ang, C. S., & Chan, K. M. (2016). A review of coronary artery disease research in Malaysia. Medical Journal of Malaysia, 74, 67-78.

14. Amir, M., Mappiare, M., & Indra, P. (2017). The impact of cytochrome P450 2C19 polymorphism on cardiovascular events in Indonesian patients with coronary artery disease. Clinical Cardiology and Cardiovascular Medicine, 1, 15-21.

15. Knuuti, J., Wijns, W., Saraste, A., Capodanno, D., Barbato, E., Funck-Brentano, C., ... & Lancellotti, P. (2020). 2019 ESC Guidelines for the diagnosis and management of chronic coronary syndromes. European Heart Journal, 41(3), 407.

16. Rajadurai, J., Ghapar, A. K., Nuruddin, A. A., Nor, A. T. M., Muthusamy, G., & Panel Expert Malaysian CPG. (2019). Malaysian clinical practice guideline for ST-elevation myocardial infarction (STEMI) 2019. Malaysian Society of Cardiology.

17. Breslow, N. (1972). Discussion of regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological), 34(2), 216-217.

18. de Carvalho, L. P., Tan, S. H., Ow, G. S., Tang, Z., Ching, J., Kovalik, J. P., ... & Foo, R. (2018). Plasmaceramides as prognostic biomarkers and their arterial and myocardial tissue correlates in acute myocardial infarction. JACC: Basic to Translational Science, 3(2), 163-175.

19. Kolek, M. J., Graves, A. J., Xu, M., Bian, A., Teixeira, P. L., Shoemaker, M. B., ... & Wei, W. Q. (2016). Evaluation of a prediction model for the development of atrial fibrillation in a repository of electronic medical records. JAMA Cardiology, 1(9), 1007-1013.

20. Hippisley-Cox, J., Coupland, C., Vinogradova, Y., Robson, J., May, M., & Brindle, P. (2007). Derivation and validation of QRISK, a new cardiovascular disease risk score for the United Kingdom: prospective open cohort study. BMJ, 335(7611), 136.

21. Hippisley-Cox, J., Coupland, C., & Brindle, P. (2017). Development and validation of QRISK3 risk prediction algorithms to estimate future risk of cardiovascular disease: prospective cohort study. BMJ, 357, j2099.

22. Mehilli, J., Kastrati, A., Dirschinger, J., Pache, J., Seyfarth, M., Blasini, R., ... & Schömig, A. (2002). Sex-based analysis of outcome in patients with acute myocardial infarction treated predominantly with percutaneous coronary intervention. JAMA, 287(2), 210-215.

23. Fisher, L. D., & Lin, D. Y. (1999). Time-dependent covariates in the Cox proportional-hazards regression model. Annual Review of Public Health, 20(1),

23. Fisher, L. D., & Lin, D. Y. (1999). Time-dependent covariates in the Cox proportional-hazards regression model. Annual Review of Public Health, 20(1),
145-157.

24. Lau, E. M., Lee, P., Lo, D., Dwyer, T., & Leung, J. (2002). The impact of age and sex on myocardial infarction rates in Hong Kong's elderly: epidemiology, risk factors, and implications. The American Journal of Geriatric Cardiology, 11(2), 92-98.

25. Ahmed, H. M., McAvay, G., & McCullough, P. A. (2019). The association between renal insufficiency and left ventricular systolic dysfunction: a narrative review. Cardiorenal Medicine, 9(2), 71-82.

26. Roger, V. L., Weston, S. A., Redfield, M. M., Hellermann-Homan, J. P., Killian, J., Yawn, B. P., ... & Jacobsen, S. J. (2004). Trends in heart failure incidence and survival in a community-based population. JAMA, 292(3), 344-350.

27. Go, A. S., Mozaffarian, D., Roger, V. L., Benjamin, E. J., Berry, J. D., Blaha, M. J., ... & Dai, S. (2014). Heart disease and stroke statistics—2014 update: a report from the American Heart Association. Circulation, 129(3), e28-e292.

28. National Heart Foundation of Australia & Cardiac Society of Australia and New Zealand. (2011). Reducing risk in heart disease: an expert guide to clinical practice for secondary prevention of coronary heart disease. National Heart Foundation of Australia.

29. Chen, Y., & Zhang, P. (2018). Data mining for the diagnosis and treatment of cardiovascular disease using electronic health records: a systematic review. Journal of Medical Systems, 42(7), 132.

30. Yan, R., Li, H., Hao, S., Wu, Y., Zhang, Q., Liu, Y., ... & Gong, Y. (2018). Association of comorbidity burden with abnormal cardiac repolarization in the Chinese population. Annals of Noninvasive Electrocardiology, 23(1), e12464.

31. Morris, A. P., Voight, B. F., Teslovich, T. M., Ferreira, T., Segrè, A. V., Steinthorsdottir, V., ... & Kathiresan, S. (2012). Large-scale association analysis provides insights into the genetic architecture and pathophysiology of type 2 diabetes. Nature Genetics, 44(9), 981-990.

32. Ridker, P. M., Danielson, E., Fonseca, F. A., Genest, J., Gotto Jr, A. M., Kastelein, J. J., ... & Braunwald, E. (2008). Rosuvastatin to prevent vascular events in men and women with elevated C-reactive protein. New England Journal of Medicine, 359(21), 2195-2207.

33. Benjamin, E. J., Muntner, P., Alonso, A., Bittencourt, M. S., Callaway, C. W., Carson, A. P., ... & Virani, S. S. (2019). Heart disease and stroke statistics—2019 update: a report from the American Heart Association. Circulation, 139(10), e56-e528.

34. Roth, G. A., Johnson, C., Abajobir, A., Abd-Allah, F., Abera, S. F., Abyu, G., ... & Azzopardi, P. (2017). Global, regional, and national burden of cardiovascular diseases for 10 causes, 1990 to 2015. Journal of the American College of Cardiology, 70(1), 1-25.

35. Centers for Disease Control and Prevention. (2019). Heart disease facts. Retrieved from https://www.cdc.gov/heartdisease/facts.htm

36. World Health Organization. (2017). Cardiovascular diseases (CVDs). Retrieved from https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

37. American Heart Association. (2021). Heart disease and stroke statistics. Retrieved from https://www.heart.org/en/news/2021/01/27/statistics-show-more-work-needed-to-save-lives-from-heart-disease-and-stroke

38. Mozaffarian, D., Benjamin, E. J., Go, A. S., Arnett, D. K., Blaha, M. J., Cushman, M., ... & Turner, M. B. (2016). Heart disease and stroke statistics—2016 update: a report from the American Heart Association. Circulation, 133(4), e38-e360.

39. Writing Group Members, Mozaffarian, D., Benjamin, E. J., Go, A. S., Arnett, D. K., Blaha, M. J., ... & Turner, M. B. (2015). Heart disease and stroke statistics—2015 update: a report from the American Heart Association. Circulation, 131(4), e29-e322.

40. Moran, A. E., Forouzanfar, M. H., Roth, G. A., Mensah, G. A., Ezzati, M., Flaxman, A., ... & Naghavi, M. (2014). The global burden of ischemic heart disease in 1990 and 2010: the Global Burden of Disease 2010 study. Circulation, 129(14), 1493-1501.

41. Roth, G. A., Johnson, C., Abajobir, A., Abd-Allah, F., Abera, S. F., Abyu, G., ... & Azzopardi, P. (2018). Global, regional, and national burden of cardiovascular diseases for 10 causes, 1990 to 2015. Journal of the American College of Cardiology, 70(1), 1-25.

42. Gaziano, T. A., Bitton, A., Anand, S., Abrahams-Gessel, S., & Murphy, A. (2010). Growing epidemic of coronary heart disease in low- and middle-income countries. Current Problems in Cardiology, 35(2), 72-115.

43. Weng, S.F., Reps, J., Kai, J., Garibaldi, J.M., & Qureshi, N. (2017). Can machine-learning improve cardiovascular risk prediction using routine clinical data? Plus One, 12(4), e0174944.

44. Pranata, R., Huang, I., & Damay, V. (2018). Should de Winter T-wave electrocardiography pattern be treated as ST-segment elevation myocardial

infarction equivalent with consequent reperfusion? A dilemmatic experience in rural area of Indonesia. Journal of Case Reports in Cardiology.

45. Jaafar, J., Atwell, E., Johnson, O., Clamp, S., & Ahmad, W.A.W. (2013). Evaluation of machine learning techniques in predicting acute coronary syndrome outcome. In Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence.

46. Liu, Y., Scirica, B.M., Stultz, C.M., & Guttag, J.V. (2016). Beatquency domain and machine learning improve prediction of cardiovascular death after acute coronary syndrome. Scientific Reports, 6, 34540.

47. Myers, P.D., Scirica, B.M., & Stultz, C.M. (2017). Machine learning improves risk stratification after acute coronary syndrome. Scientific Reports, 7(1), 12692.

48. Razavi, A.C., Monlezun, D.J., Sapin, A., Sarris, L., Schlag, E., Dyer, A., et al. (2019). Etiological role of diet in 30-Day readmissions for heart failure: implications for reducing heart failure–associated costs via culinary medicine. American Journal of Lifestyle Medicine, 1559827619861933.

49. Hassoun, M.H. (1995). Fundamentals of artificial neural networks. MIT Press.

50. Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 16(10), 906–914.

51. Mansoor, H., Elgendy, I.Y., Segal, R., Bavry, A.A., & Bian, J. (2017). Risk prediction model for in-hospital mortality in women with ST-elevation myocardial infarction: a machine learning approach. Heart & Lung, 46(6), 405–411.

52. Rawshani, A., Rawshani, A., Sattar, N., Franzén, S., McGuire, D.K., Eliasson, B., et al. (2019). Relative prognostic importance and optimal levels of risk factors for mortality and cardiovascular outcomes in Type 1 diabetes Mellitus. 1. Circulation, 139(6), 1900–1912.

53. Than, M.P., Pickering, J.W., Sandoval, Y., Shah, A.S., Tsanas, A., Apple, F.S., et al. (2019). Machine learning to predict the likelihood of acute myocardial infarction. Circulation, 140(11), 899–909.

54. Cunningham, P., Delany, S.J. (2020). k-Nearest Neighbour Classifiers. arXiv preprint arXiv:2020;2004.04523.

55. Dey, D., Slomka, P.J., Leeson, P., Comaniciu, D., Shrestha, S., Sengupta, P.P., et al. (2019). Artificial intelligence in cardiovascular imaging: JACC state-of-the-art review. Journal of the American College of Cardiology, 73(11), 1317–1335.

56. Aryal, S., Alimadadi, A., Manandhar, I., Joe, B., & Cheng, X. (2020). Machine learning strategy for gut microbiome-based diagnostic screening of cardiovascular disease. Hypertension, 76(5), 1555–1562.

57. Yang, X. (2018). Identification of risk genes associated with myocardial infarction based on the recursive feature elimination algorithm and support vector machine classifier. Molecular Medicine Reports, 17(1), 1555–1560.

58. Cai, C., Fang, J., Guo, P., Wang, Q., Hong, H., & Moslehi, J. (2018). In silico pharmacoepidemiologic evaluation of drug-induced cardiovascular complications using combined classifiers. Journal of Chemical Information and Modeling, 58(5), 943–956.

59. Gatsonis, C.A., Epstein, A.M., Newhouse, J.P., Normand, S.L., & McNeil, B.J. (1995). Variations in the utilization of coronary angiography for elderly patients with an acute myocardial infarction: an analysis using hierarchical logistic regression. Medical Care, 33(6), 625–642. doi:10.1097/00005650-199506000-00005.

60. Kennedy, R.L., Fraser, H.S., McStay, L.N., & Harrison, R.F. (1996). Early diagnosis of acute myocardial infarction using clinical and electrocardiographic data at presentation: derivation and evaluation of logistic regression models. European Heart Journal, 17(8), 1181–1191.

61. Gross, P., Honnorat, N., Varol, E., Wallner, M., Trappanese, D.M., Sharp, T.E., et al. (2016). Nuquantus: Machine learning software for the characterization

and quantification of cell nuclei in complex immunofluorescent tissue images. Scientific Reports, 6, 23431.

62. Piros, P., Ferenci, T., Fleiner, R., Andréka, P., Fujita, H., Föző, L., et al. (2019). Comparing machine learning and regression models for mortality prediction based on the Hungarian myocardial infarction registry. Journal of Knowledge-Based Systems, 179, 1–7.

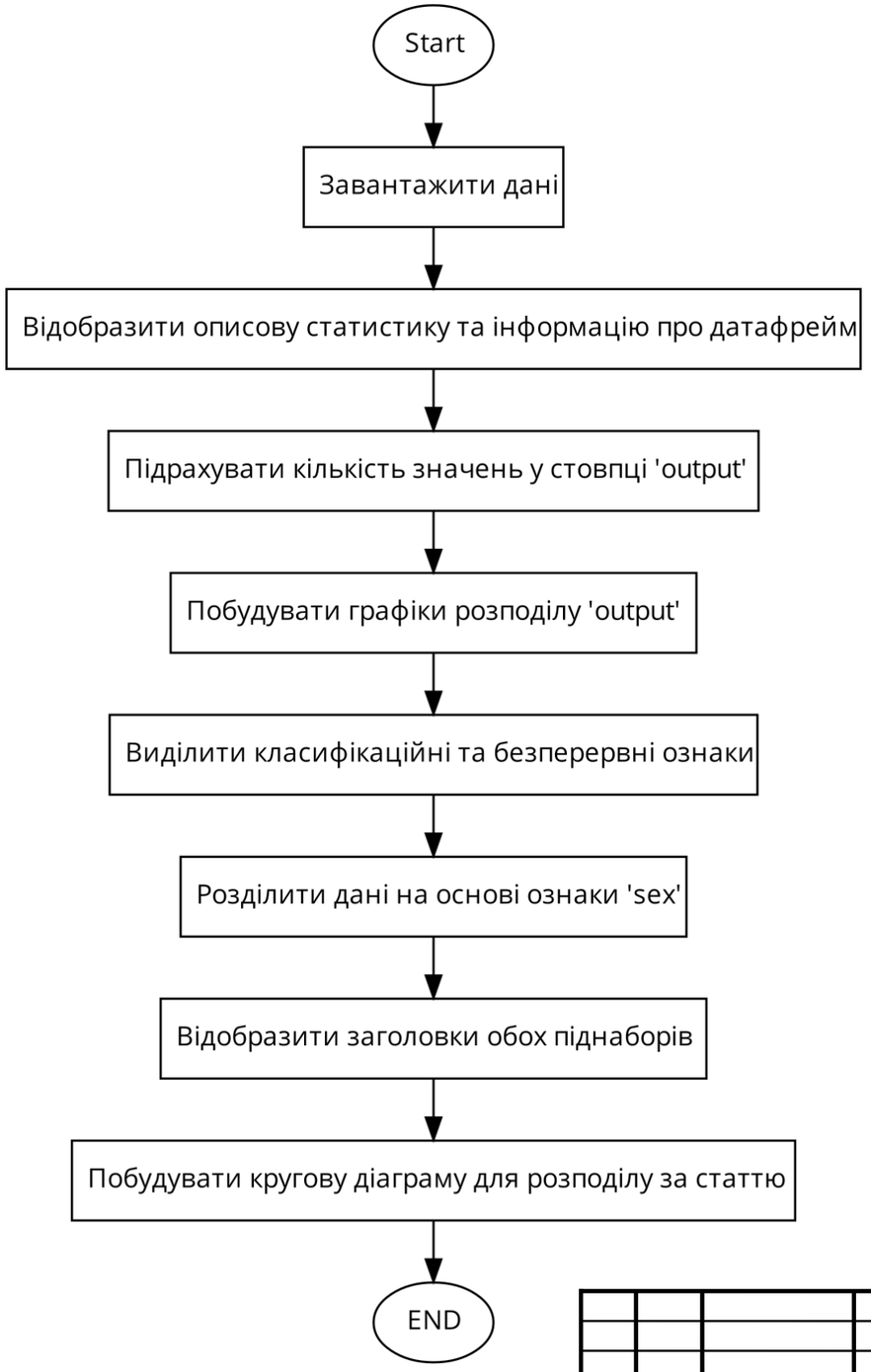
63. Lipyanina, H., Sachenko, S., Lendyuk, T., Brych, V., Yatskiv, V., & Osolinskiy, O. (2021). Method of detecting a fictitious company on the machine learning base. In *Advances in Computer Science for Engineering and Education IV* (pp. 138-146). Cham: Springer International Publishing.

64. Lipyanina-Goncharenko, H., Brych, V., Sachenko, S., Lendyuk, T., Bykovyy, P., & Zahorodnia, D. (2021). Method of Forming a Training Sample for Segmentation of Tender Organizers on Machine Learning Basis. In *COLINS* (pp. 1843-1852).

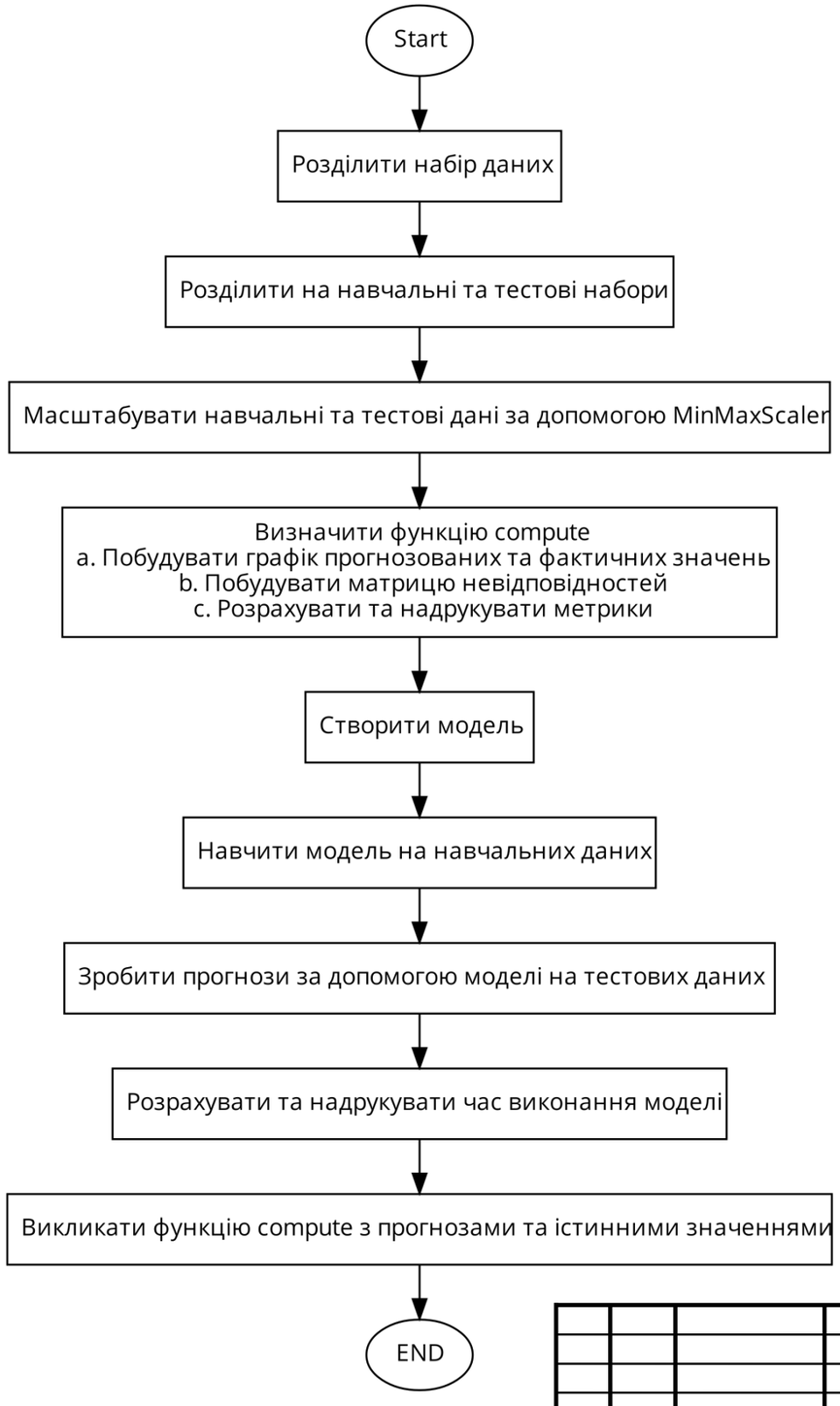
65. Lipianina-Honcharenko, K., Wolff, C., Chyzhovska, Z., Sachenko, A., Lendiuk, T., & Grodskyi, S. (2022, October). Intelligent Method for Forming the Consumer Basket. In *Information and Software Technologies: 28th International Conference, ICIST 2022, Kaunas, Lithuania, October 13–15, 2022, Proceedings* (pp. 221-231). Cham: Springer International Publishing.

66. Heart Attack Analysis & Prediction Dataset. (б. д.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset?datasetId=1226038&sortBy=commentCount

67. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньопрофесійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. – Тернопіль: ЗУНУ, 2021. – 56 с.



					ДП.КН. 8351761.001 А1	
					Алгоритм завантаження даних та первинний аналіз	Літера: — Маса: — Масштаб: —
Змн.	Арк.	№ докум.	Підпис	Дата		Аркуш 1 / Аркушів 1
Розроб.		Кравчук Б.				
Перевір.		Дорош В.І.				ЗУНУ.ФКІТ.КН-41
Консультант						
Т. Контр.						
Н. Контр.		Дорош В.І.				
Затверд.		Комар М.П.				



					ДП.КН. 8351761.001 А1	
Змн.	Арк.	№ докум.	Підпис	Дата		
Розроб.		Кравчук Б.			Алгоритм інтелектуального передбачення серцево-судинних захворювань	Літера / Маса / Масштаб
Перевір.		Дорош В.І.				Аркуш 1 / Аркушів 1
Консультант						
Т. Контр.						ЗУНУ.ФКІТ.КН-41
Н. Контр.		Дорош В.І.				
Затверд.		Комар М.П.				