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

Магістерська кваліфікаційна робота на тему:

"ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ СТАНУ ХВОРИХ НА ГЕПАТИТ"

Виконав: студент 2 курсу, групи 2ІСТ-22м
спеціальності 126 – «Інформаційні системи та технології»
_____ Ігор ГУЦУ
Керівник: к.т.н., доц. каф. САІТ
_____ Сергій ЖУКОВ
«08» ____12____ 2023 р.
Рецензент: к.т.н., доц. каф. КН
_____ Володимир ОЗЕРАНСЬКИЙ
«14» ____12____ 2023 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«05» ____12____ 2023 р.

Вінниця ВНТУ – 2023 рік

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

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«01» 09 2023 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Гуцу Ігорю Петровичу

1. Тема роботи: "Інформаційна технологія аналізу та передбачення стану хворих на гепатит",
керівник роботи: Сергій ЖУКОВ., к.т.н., доц. каф. САІТ,
затверджені наказом закладу вищої освіти від «18» 09 2023 року №247
2. Строк подання студентом роботи «30» 11 2023 року
3. Вихідні дані до роботи:
Датасет «Kaggle Stroke Prediction Dataset» з даними для передбачення ймовірності гепатиту у пацієнтів.
https://www.kaggle.com/datasets/fedesoriano/hepatitis-c-dataset
4. Зміст текстової частини:
   1) Актуальність проблеми;
   2) Аналіз бібліотек для вирішення задачі;
   3) Підготовка даних;
   4) Побудова моделей машинного навчання
   5) Економічна частина.
5. Перелік ілюстративного матеріалу:
   1) Кореляційна матриця показників;
   2) Матриця плутанини моделі Logistic Regression;
   3) Матриця плутанини моделі Random Forest Classifier;
   4) Матриця плутанини моделі Gradient Boosting;
   5) Матриця плутанини моделі Support Vector Machine;
   6) Матриця плутанини моделі Decision Tree Classifier;
   7) Графік результатів класифікації;
   8) Схема роботи алгоритму.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Наталія БУРЄННІКОВА, д.е.н., проф.каф. ЕПВМ	04.11.23	10.11.23

7. Дата видачі завдання «04» 09 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	09.09	20.09	
2	Аналіз методів машинного навчання для вирішення поставленої задачі	20.09	10.10	
3	Розробка моделі класифікації	10.10	30.10	
4	Економічна частина	01.11	10.11	
5	Оформлення матеріалів до захисту МКР	10.11	30.11	

Студент_____ Ігор ГУЦУ

Керівник роботи_____ Сергій ЖУКОВ

АНОТАЦІЯ

УДК 004.623

Гуцу І. П. Інформаційна технологія аналізу та передбачення стану хворих на гепатит. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2023. 100 с.

На укр. мові. Бібліогр.: 32 назв; рис.: 51 ; табл.: 12.

У магістерській кваліфікаційній роботі розроблено інформаційну технологію аналізу та передбачення стану хворих на гепатит. Під час розробки було обрано датасет, проведено розвідувальний аналіз та побудовано моделі передбачення.

Ілюстративна частина складається з 9 плакатів, що включають в себе результати тестування розроблених моделей.

У розділі економічної частини роботи детально розглядається питання доцільності розробки та впровадження інформаційної технології для прогнозування захворювань на гепатит.

Ключові слова: Python, гепатит, розвідувальний аналіз, захворювання.

ABSTRACT

UDC 004.623

Hutsu I.P. Information technology for analyzing and predicting the condition of hepatitis patients. Master's qualification thesis on specialty 126 - information systems and technologies, educational and professional program - information technologies of data and image analysis. Vinnytsia: VNTU, 2023. 100 p.

In Ukrainian speech Bibliography: 32 titles; Fig.: 51; tab.: 12.

The master's thesis developed information technology for analysis and prediction of the condition of hepatitis patients. During development, a dataset was selected, exploratory analysis was performed, and prediction models were built.

The illustrative part consists of 9 posters, which include the results of testing the developed models.

In the section of the economic part of the work, the feasibility of developing and implementing information technology for forecasting hepatitis diseases is considered in detail.

Key words: Python, hepatitis, intelligence analysis, disease.

ЗМІСТ

ВСТУП 4
1 ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ 6
1.1 Аналіз предметної області 6
1.2 Інформаційні технології для діагностики та виявлення гепатиту 9
1.3 Огляд готових рішень для передбачення стану хворих на гепатит 15
1.4 Висновки 23
2 АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ 25
2.1 Вибір оптимального середовища розробки 25
2.2 Модель машинного навчання «Decision Tree» 29
2.3 Модель машинного навчання «Support Vector Machine Classifier» 31
2.4 Модель машинного навчання «Gradient Boosting Classifier» 33
2.5 Модель машинного навчання «Random Forest Classifier» 34
2.6 Модель машинного навчання «Logistic Regression» 36
2.7 Висновки 38
3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ СТАНУ ХВОРИХ ТА ГЕПАТИТ 39
3.1 Аналіз обраних бібліотек та побудова схеми алгоритму 39
3.2 Підготовка даних до розвідувального аналізу 42
3.3 Розвідувальний аналіз 48
3.4 Використання моделей машинного навчання для аналізу даних 59
3.5 Висновки 71
4 ЕКОНОМІЧНА ЧАСТИНА 73
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки 73
4.2 Розрахунок узагальненого коефіцієнта якості розробки 78
4.3 Розрахунок витрат на проведення науково-дослідної роботи 80
4.3.1 Витрати на оплату праці 80
4.3.2 Відрахування на соціальні заходи 83

4.3.3 Сировина та матеріали 83
4.3.4 Розрахунок витрат на комплектуючі 84
4.3.5 Спецустаткування для наукових (експериментальних) робіт 85
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт 86
4.3.7 Амортизація обладнання, програмних засобів та приміщень 87
4.3.8 Паливо та енергія для науково-виробничих цілей 88
4.3.9 Службові відрядження 90
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації 90
4.3.11 Інші витрати 91
4.3.12 Накладні (загальновиробничі) витрати 91
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором 92
4.5 Висновки 97
ВИСНОВКИ 98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 100
Додаток А (обов’язковий) Технічне завдання 104
Додаток Б (обов’язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень 106
Додаток В (довідниковий) Лістинг програми 107
Додаток Г (обов’язковий) Ілюстративна частина 117

ВСТУП

Актуальність теми. Щодня інформаційні технології стрімко розвиваються, призводячи до експоненційного зростання обсягу даних у світі інформаційних мереж. Це безпосередньо сприяє виникненню можливостей використання даних для різних цілей, зокрема для аналізу, класифікації та прогнозування. Завдяки цим даним можна проводити системний аналіз, виявляючи тенденції та ідентифікуючи фактори, які можуть оптимізувати функціонування різних систем.

Однією з актуальних проблем на сучасному етапі є гепатит, і вирішення цієї проблеми передбачає вдосконалення методів та засобів її виявлення. Раннє виявлення гепатиту має велике значення, оскільки це надає людям більше часу для вживання ефективних заходів.

Таким чином, використання інформаційних технологій для аналізу та обробки даних стає ключовим елементом у вдосконаленні підходів до діагностики та управління захворюванням. Інформаційна технологія аналізу дозволяє виявити не лише поточний стан справ, але і прогнозувати майбутні тенденції, що створює можливість вчасного реагування та оптимізації лікувальних процесів.

Мета і завдання роботи. Метою даної магістерської кваліфікаційної роботи є підвищення точності передбачення стану хворих на гепатит. Для досягнення поставленої мети необхідно розв’язати наступні завдання:

- повести огляд існуючих систем;
- підготувати дані для подальшої роботи;
- провести розвідувальних аналіз даних;
- побудувати моделі та виконати прогнозування;
- оцінити результати роботи моделей.

Об’єктом дослідження магістерської кваліфікаційної роботи є процес розроблення інформаційної технології аналізу передбачення стану хворих на гепатит.

Предметом дослідження магістерської кваліфікаційної роботи є інформаційна технологія аналізу передбачення стану хворих на гепатит.

Новизна отриманих результатів полягає в подальшому розвитку інформаційної технології передбачення стану хворих на гепатит. Це можливо за рахунок сучасних алгоритмів машинного навчання, які дадуть змогу підвищити точність результатів передбачення.

Практичне значення. Отримані результати є цінними у сфері медицини для осіб, які мають захворювання печінки. Крім того, вони можуть служити основою для розробки ефективних стратегій виявлення хвороби на ранніх етапах.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи апробовані на LIII Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (2023-2024).

Публікації результатів магістерської кваліфікаційної роботи. За даною темою дослідження зроблено доповідь на LIII Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (2023-2024) з публікацією тез [1]

1 ХАРАКТЕРИСТИКА ОБ’ЄКТУ ДОСЛІДЖЕНЬ

1.1 Аналіз предметної області

Глобальна боротьба з гепатитом обумовлена численними факторами, що включають в себе розповсюдженість хвороби, доступ до медичної допомоги та ефективність вакцинаційних програм. Гепатит, як запальне захворювання печінки, може виникати внаслідок різних причин, включаючи інфекції та отруєння.

Найбільш розповсюдженими формами гепатиту є типи A, B та C, кожен з яких має свої особливості та ризики. Однією з головних труднощів у подоланні гепатиту є можливість переходу хвороби у хронічну форму, що може вести до серйозних ускладнень, таких як цироз та рак печінки.

Ефективність боротьби з гепатитом суттєво залежить від географічного розташування та умов гігієни. Вакцинаційні програми, зокрема проти гепатитів типу A та B, визнані як важливий інструмент профілактики. Проте, необхідно посилити зусилля для забезпечення широкого доступу до вакцин та медичної допомоги, особливо в менше розвинених регіонах [2].

Однак лише вакцинація та лікування недостатні для вирішення проблеми гепатиту. Програми скринінгу важливі для вчасного виявлення та лікування хворих, особливо серед груп ризику. Соціально-економічний вплив гепатиту, такий як втрати працездатності та великі витрати на лікування, підкреслює необхідність комплексного підходу до вирішення проблеми.

Подолання гепатиту вимагає поєднання вакцинаційних стратегій, ефективних програм скринінгу, підвищення свідомості громадськості та забезпечення доступу до якісної медичної допомоги. Це завдання, що вимагає глобального співробітництва та прийняття комплексних заходів для зменшення поширення гепатиту та його негативного впливу на здоров'я світового населення.

Успішне подолання гепатиту також вимагає постійного вдосконалення наукових досліджень та розвитку нових технологій у галузі діагностики та лікування. Інноваційні методи, спрямовані на покращення ефективності терапії та раннє виявлення захворювання, можуть вирішити важливі аспекти глобальної стратегії.

Розширення зусиль у підтримці наукових досліджень та глобального обміну знанням може сприяти зниженню вартості ліків та поліпшенню їх доступності. Такий підхід сприятиме створенню більш доступної та ефективної системи охорони здоров'я.

Окрім того, важливо акцентувати увагу на ролі громадськості у боротьбі з гепатитом. Запуск інформаційних кампаній та освітніх програм може значно підвищити рівень усвідомлення про фактори ризику, методи профілактики та важливість скринінгу.

Напрямки для майбутніх заходів також повинні враховувати питання соціальної справедливості та врахування потреб найбільш вразливих груп населення. Забезпечення рівного доступу до медичних послуг та участь представників громадськості у прийнятті рішень стануть важливими елементами в цьому контексті [2].

Остаточно, лише через системний та взаємодіючий підхід можна досягти значущого прогресу в подоланні гепатиту. Глобальна співпраця, підтримка наукових досліджень, активна участь громадськості та розвиток доступних та ефективних методів лікування є ключовими елементами на шляху до створення здорового світу.

Сприяння громадськості у виявленні та профілактиці гепатиту може здійснюватися через широкомасштабні освітні кампанії, спрямовані на підвищення свідомості про фактори ризику та важливість раннього діагнозу. Залучення громадян до програм скринінгу та облік їхнього стану здоров'я може зменшити інцидентність та сприяти вчасному початку лікування.

Важливо також акцентувати увагу на значущості подолання соціального стигми, пов'язаного із гепатитом. Сприяння розумінню того, що гепатит — це захворювання, а не ознака особистої недбалості чи неправильного способу життя, є ключовим для створення сприятливого середовища та покращення якості життя хворих.

Міжнародне співробітництво та обмін досвідом мають вирішальне значення у розробці та впровадженні ефективних стратегій. Передові країни можуть надавати допомогу менше розвиненим регіонам у реалізації програм вакцинації, лікування та освіти.

Загалом, забезпечення збалансованого та всебічного підходу до проблеми гепатиту передбачає взаємодію між державними структурами, медичними організаціями, науковою спільнотою та громадськістю. Тільки в цьому випадку можна досягти значущих результатів у попередженні та лікуванні цієї серйозної медичної проблеми [2].

Особливу увагу слід звертати на аспекти взаємодії міжнародних організацій та урядів для розробки та реалізації глобальних стратегій управління гепатитом. Спільні ініціативи, спрямовані на створення стандартів лікування та надання підтримки країнам з обмеженими ресурсами, можуть значно полегшити ситуацію.

Застосування технологій телемедицини та інноваційних методів мобільного здоров'я може розширити доступ до медичних консультацій та діагностики в регіонах, де відсутня адекватна медична інфраструктура.

Підтримка пацієнтів та їхніх родин, зокрема шляхом формування груп взаємодопомоги та надання інформаційної підтримки, є важливим елементом. Посилення прав пацієнтів та визнання їхньої ролі у прийнятті рішень про своє лікування сприятиме формуванню більш інформованих та активних споживачів медичних послуг.

Загалом, вирішення проблеми гепатиту вимагає комплексного та системного підходу, враховуючи наукові досягнення, технологічні інновації, соціально-економічні аспекти та участь громадян у процесах управління своїм

здоров'ям. Тільки спільними зусиллями світова спільнота може досягти значущих результатів у попередженні та лікуванні гепатиту, забезпечивши здоров'я та добробут для всіх.

1.2 Інформаційні технології для діагностики та виявлення гепатиту

Сучасні інформаційні технології виявлення гепатиту орієнтовані на поліпшення точності та швидкості діагностики. До засобів входять полімеразна ланцюгова реакція (PCR) для молекулярного виявлення вірусного матеріалу та імунодіагностичні методи, такі як ELISA (рис. 1.1). Ці технології сприяють ранньому виявленню гепатиту та покращують ефективність лікування.

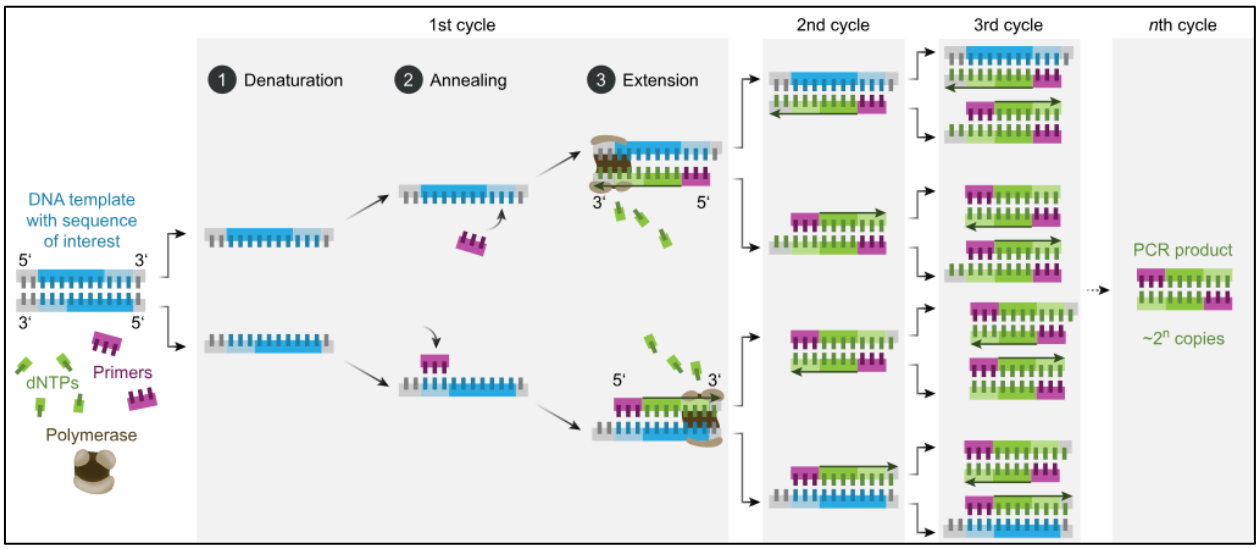


Рисунок 1.1 – Принцип роботи полімеразної ланцюгової реакції.

Додатково, електронні медичні записи та системи обробки даних можуть значно полегшити обмін інформацією між лікарнями та лікарями, сприяючи координації у наданні допомоги пацієнтам з гепатитом. Також, з розвитком технологій телемедицини можливості дистанційного моніторингу стану хворих збільшуються, що може бути корисним для пацієнтів, які проживають віддалено від медичних закладів [3].

Машинне навчання та аналіз великих обсягів даних також використовуються для виявлення закономірностей та ризикових факторів гепатиту, що може допомогти в удосконаленні стратегій профілактики та ранньої діагностики.

Крім того, розвиток біомаркерів та нових технологій для аналізу біологічних речовин може дозволити більш докладно оцінювати стан печінки та виявляти патологічні зміни на ранніх стадіях. Наприклад, використання мас-спектрометрії та інших аналітичних методів дозволяє визначати концентрації певних метаболітів у крові, що може служити індикатором функціонального стану печінки та допомагати в ранньому виявленні патологій, включаючи гепатит.

Технології блокчейн також можуть знайти застосування в забезпеченні безпеки та конфіденційності медичних даних, що є важливим аспектом в області діагностики та лікування гепатиту.

На рівні конкретних програм та систем у сфері лікування гепатиту важливу роль відіграють різні медичні платформи та інформаційні системи.

HCV-TARGET — це подовжнє обсерваційне дослідження, яке створить ретельно підтримуваний дослідницький реєстр пацієнтів з ВГС, які отримували противірусну терапію, призначений для швидкого інформування про стратегії кращого лікування популяцій, недостатньо представлених у клінічних випробуваннях, виявлення та усунення прогалин в освіті щодо рекомендацій щодо лікування та побічних ефектів. управління подіями, щоб оптимізувати рівень стійкої вірусологічної відповіді (SVR), і служити основним ресурсом для важливих спільних трансляційних досліджень з використанням біозразків і клінічних даних різних груп пацієнтів [3].

HCV-TARGET — це спільний академічний консорціум головних дослідників із академічних установ, що фінансуються Clinical and Translational Award (CTSA), а також сайтів громад, пов’язаних із академічними сайтами в географічній близькості. Клінічний координаційний центр (CCC) знаходиться

в Університеті Флориди, а Центр координації даних (DCC) знаходиться в Університеті Північної Кароліни в Чапел-Хілл.

Реєстр HCV-TARGET характеризуватиме популяцію пацієнтів із хронічним гепатитом С (ВГС), які отримують противірусну терапію в академічних та громадських установах. Будуть досліджені такі характеристики пацієнта, як вік, раса, етнічна приналежність, супутня патологія, захворювання та статус лікування.

mHealth Apps — це мобільні додатки, призначені для підтримки пацієнтів і медичних фахівців у керуванні та моніторингу гепатиту. Вони можуть включати функції моніторингу симптомів, планування лікування, ведення електронного щоденника, доступ до інформаційних ресурсів та сприяти взаємодії з медичними фахівцями. Такі додатки спрощують управління захворюванням та надають пацієнтам додаткові засоби для самостійного контролю над своїм здоров'ям [5].

Liver Health - це мобільний додаток, призначений для збереження та підтримки здоров'я печінки. Зазвичай він надає інформацію про те, як дотримуватися здорового способу життя, щоб забезпечити оптимальну функцію печінки. Додаток може містити поради щодо харчування, обмеження споживання алкоголю, рекомендації з фізичної активності, а також інші корисні поради для збереження здоров'я печінки. Такі додатки можуть бути корисними для тих, хто бажає підтримувати та покращити своє здоров'я печінки (рис. 1.2) [6].

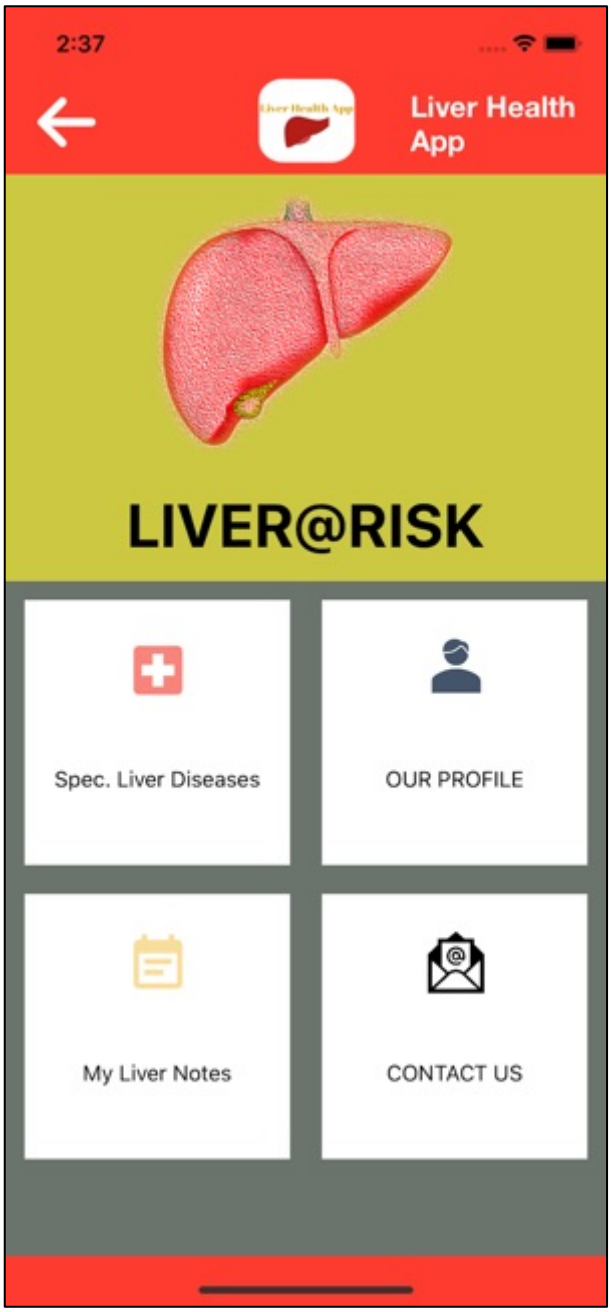


Рисунок 1.2 – Мобільний додаток «Liver Health».

В такому додатку можуть також міститися інструменти для ведення щоденника з метою моніторингу стану здоров'я печінки, а також нагадування про важливі аспекти, наприклад, прийом ліків чи регулярні медичні обстеження.

CDC (Центри контролю та профілактики захворювань) надають докладну інформацію про гепатит С (HCV). Ця інформація включає різні аспекти, такі як передача, симптоми, тестування, профілактика та лікування.

CDC є надійним джерелом актуальної та довідкової інформації з питань громадського здоров'я, включаючи гепатит С.

На рисунку 1.3 показано приклад мобільного додатку що використовує дані з CDC.

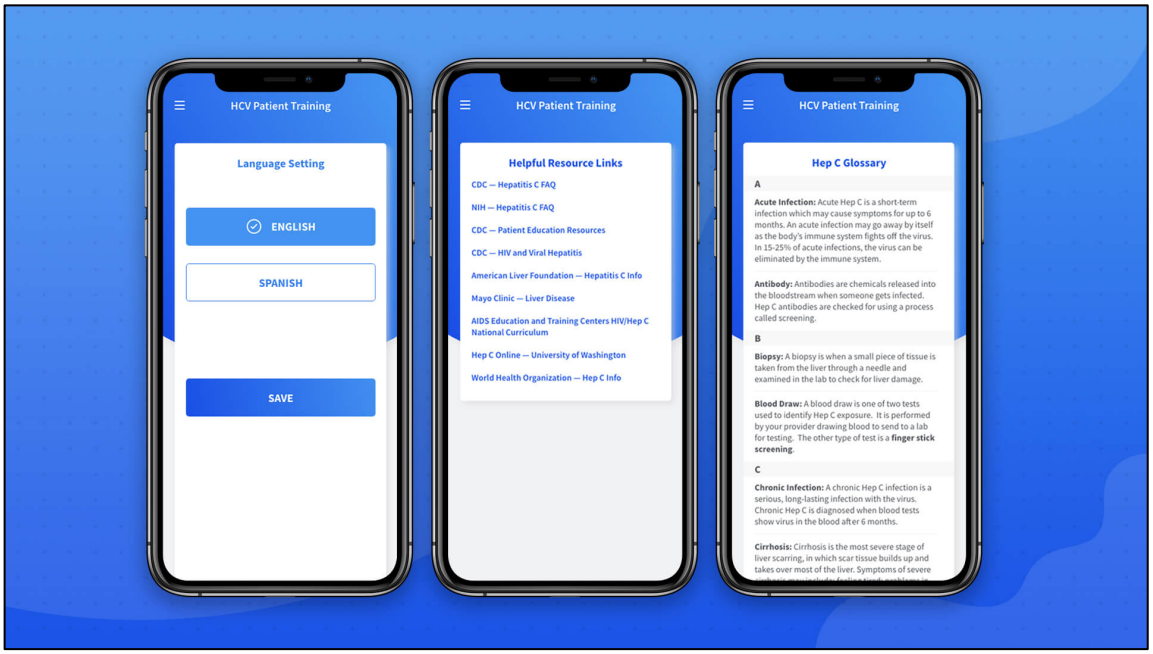


Рисунок 1.3 – Мобільний додаток «ConnQuER Hep C».

Інформація CDC щодо гепатиту С містить важливі відомості для громадськості та фахівців з охорони здоров'я. Вона охоплює важливі аспекти такі як ризики передачі вірусу, методи діагностики, ознаки зараження, рекомендації щодо тестування та вакцинації, а також доступні методи профілактики та лікування.

Звернення до офіційного веб-сайту CDC або їхнього розділу, приділеного вірусному гепатиту, може забезпечити широкий доступ до актуальної та авторитетної інформації про гепатит С [7].

My GI Health - це мобільний додаток, який зосереджений на здоров'ї та функціонуванні системи травлення (гастроінтестинальної системи). Додаток призначений для відстеження та моніторингу різних аспектів здоров'я органів травлення, зокрема шлунку та кишечника. Може надавати користувачам можливість реєструвати їжу, симптоми, а також вести щоденник здоров'я.

Використання такого додатка може бути корисним для тих, хто стикається з різними проблемами травлення та бажає вести більш свідомий спосіб життя щодо харчування та здоров'я шлунку та кишечника (рис. 1.4).

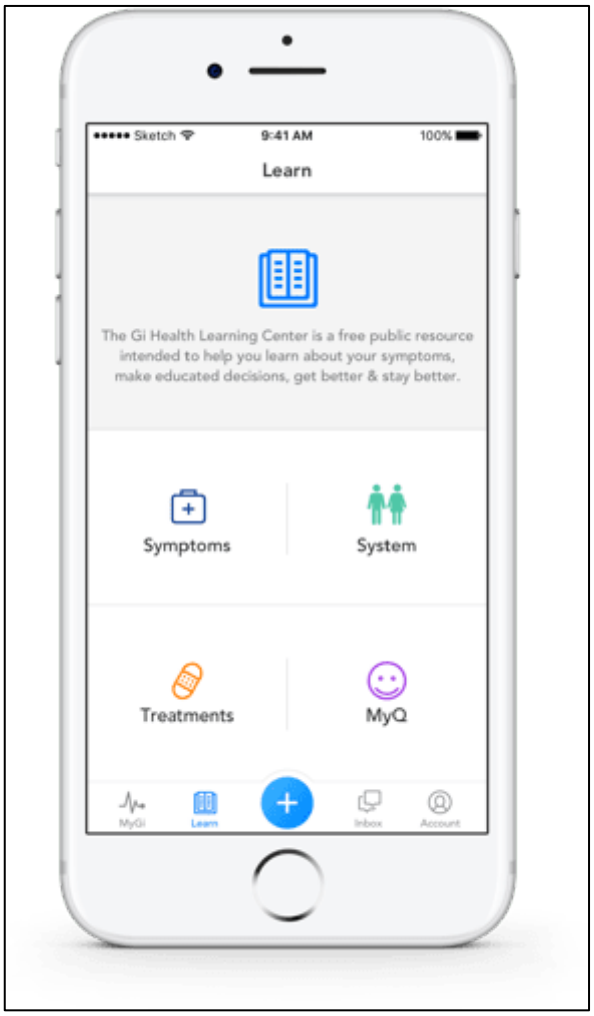


Рисунок 1.4 – Мобільний додаток «My GI Health»

Окрім цього, "My GI Health" може містити корисні рекомендації та інші інструменти, що допомагають в особистому контролі та управлінні здоров'ям системи травлення [8].

HepVu - це інтерактивний інструмент, створений з метою візуалізації та розуміння поширення вірусного гепатиту в Сполучених Штатах. Даний інструмент використовує географічні карті та графіки для надання користувачам доступу до інформації про розподіл гепатиту за різними регіонами та демографічними групами (рис. 1.5).

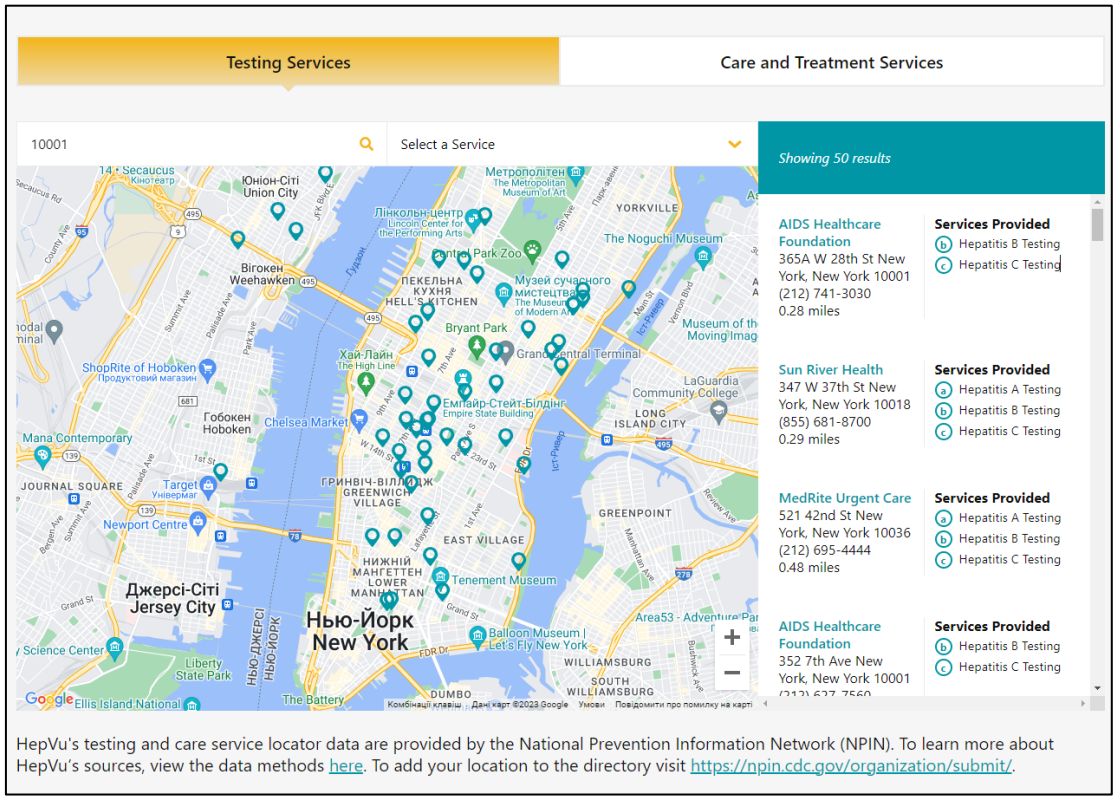


Рисунок 1.5 – Сервіс «HepVu».

HepVu розроблений для того, щоб створити зручну та легку у використанні інтерактивну платформу, де користувачі можуть отримувати доступ до актуальної інформації про гепатит у вигляді картографічних зображень. Це може сприяти розумінню розподілу захворювання, виявленню пандемійних областей та сприяти розробці стратегій з контролю та профілактики гепатиту [9].

HepVu дозволяє користувачам легко взаємодіяти з даними та розширює можливості спільнот та медичних фахівців в боротьбі з вірусним гепатитом.

1.3 Огляд готових рішень для передбачення стану хворих на гепатит

Оскільки програмне рішення буде розроблятись на основі готового датасету, доцільно розглянути вже наявні рішення, які були розроблені з його використанням.

Після аналізу сторінки датасету на платформі Kaggle, було виявлено деяку кількість готових проектів, розглянемо рішення під назвою «Ensemble-of-3-Models-Accuracy=95%» (рис. 1.6) [10].

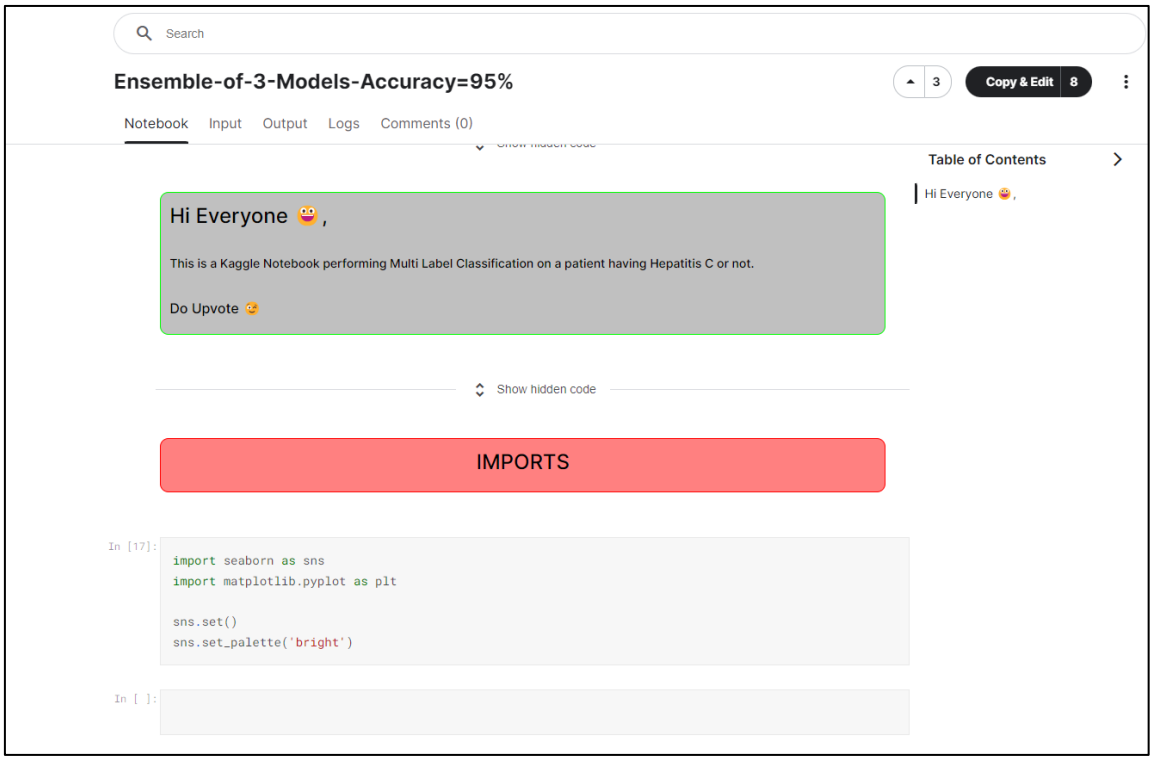


Рисунок 1.6 – Сторінка готового рішення «Ensemble-of-3-Models-Accuracy=95%».

Для впровадження інформаційної технології було успішно використано низку моделей машинного навчання, які демонстрували вражаючі результати на кожному етапі валідації. Результати вимірювалися за допомогою метрики ROC (Receiver Operating Characteristic), що вказує на ефективність класифікаційних моделей (рис. 1.7).

CatBoost: ROC: 0.9491525423728814

XGBoost: ROC: 0.940677966101695

LGBM (Light Gradient Boosting Machine): ROC: 0.9491525423728814

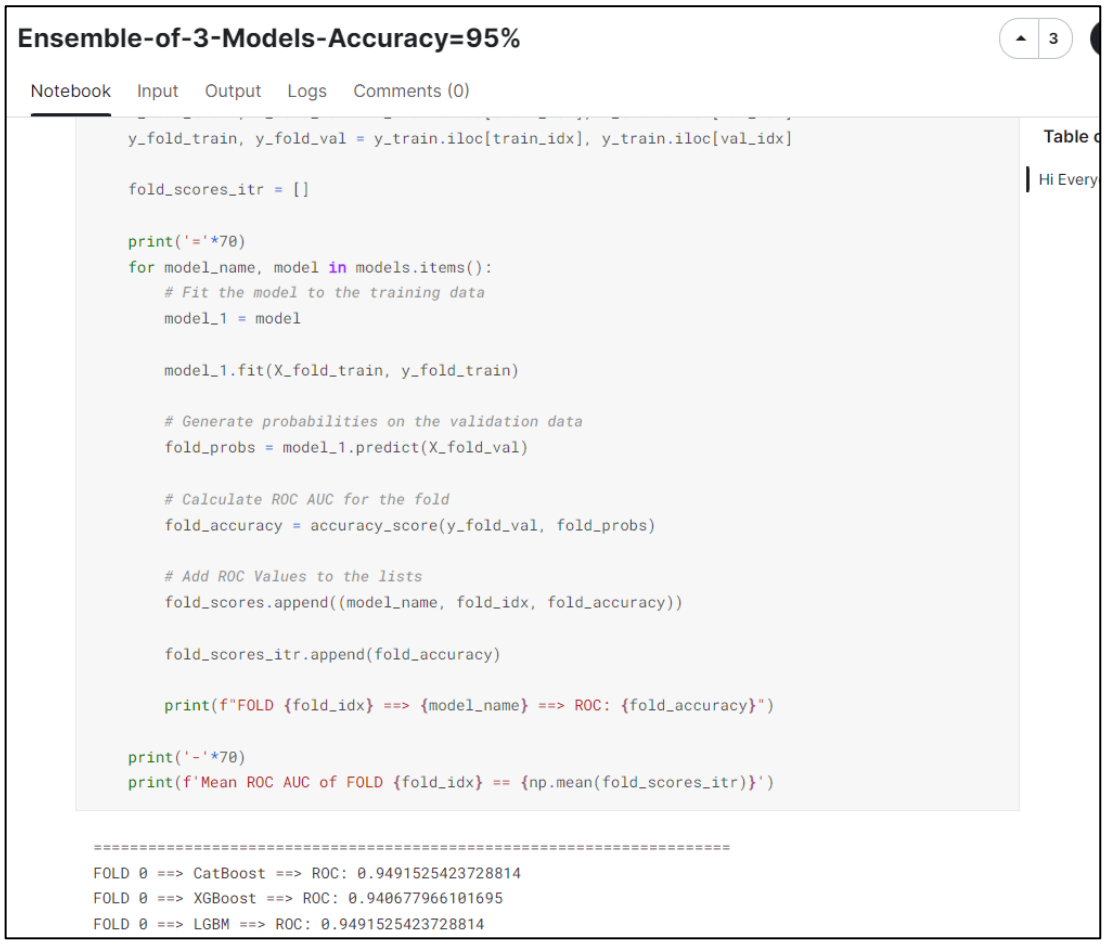


Рисунок 1.7 – Код та результат побудови моделей машинного навчання.

Незважаючи на успішні результати, можна відзначити, що значення ROC для моделей CatBoost і LGBM є високими, проте існує простір для покращень в точності передбачень. Також, хоча модель XGBoost продемонструвала добрі показники, існує можливість розглядати альтернативні архітектури або налаштування для досягнення ще кращих результатів [10].

Важливо зазначити, що в даному вирішенні відсутня діаграма важливості ознак або будь-який опис, який розкривав би важливість конкретних аспектів. Відсутність цього елементу обмежує повноту розуміння усієї картини та може ускладнити аналіз. При наявності відповідної інформації про важливість різних ознак можна було б ефективніше оцінити вплив кожного з них на рішення та зробити більш обґрунтовані висновки. Рекомендується доповнити або уточнити рішення за допомогою додаткової

інформації про важливість факторів, щоб забезпечити більш глибоке та повне розуміння представленої аналітики.

Однією з вже успішно реалізованих та ефективних інформаційних технологій є дослідження, яке отримало назву «EDA and hepatitis C prediction using RFs» (рис.1.8). Цей проект відзначився позитивними результатами.

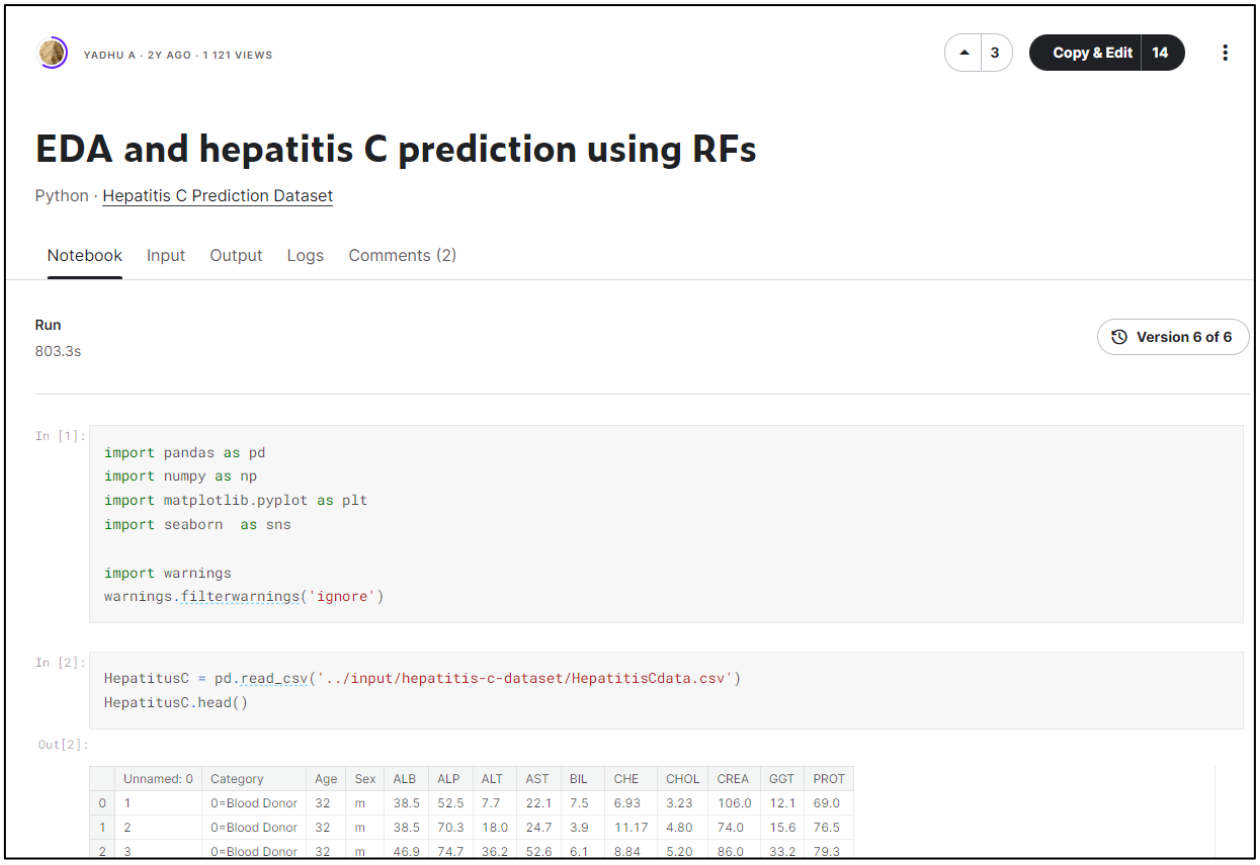


Рисунок 1.8 – Сторінка готового рішення «EDA and hepatitis C prediction using RFs».

Результат моделі Random Forest, отриманий за метрикою r2_score на рівні 0.92, свідчить про високу точність прогнозування. Однак у дослідженні було використано лише одну модель машинного навчання, що може обмежити рівень деталізації та універсальність результатів[11].

Крім того, аналіз важливості окремих ознак у моделі був проведений недостатньо докладно. Недолік цієї аналітики може призвести до недостатньої розглянутості ключових аспектів, що впливають на прогнозування (рис. 1.9).

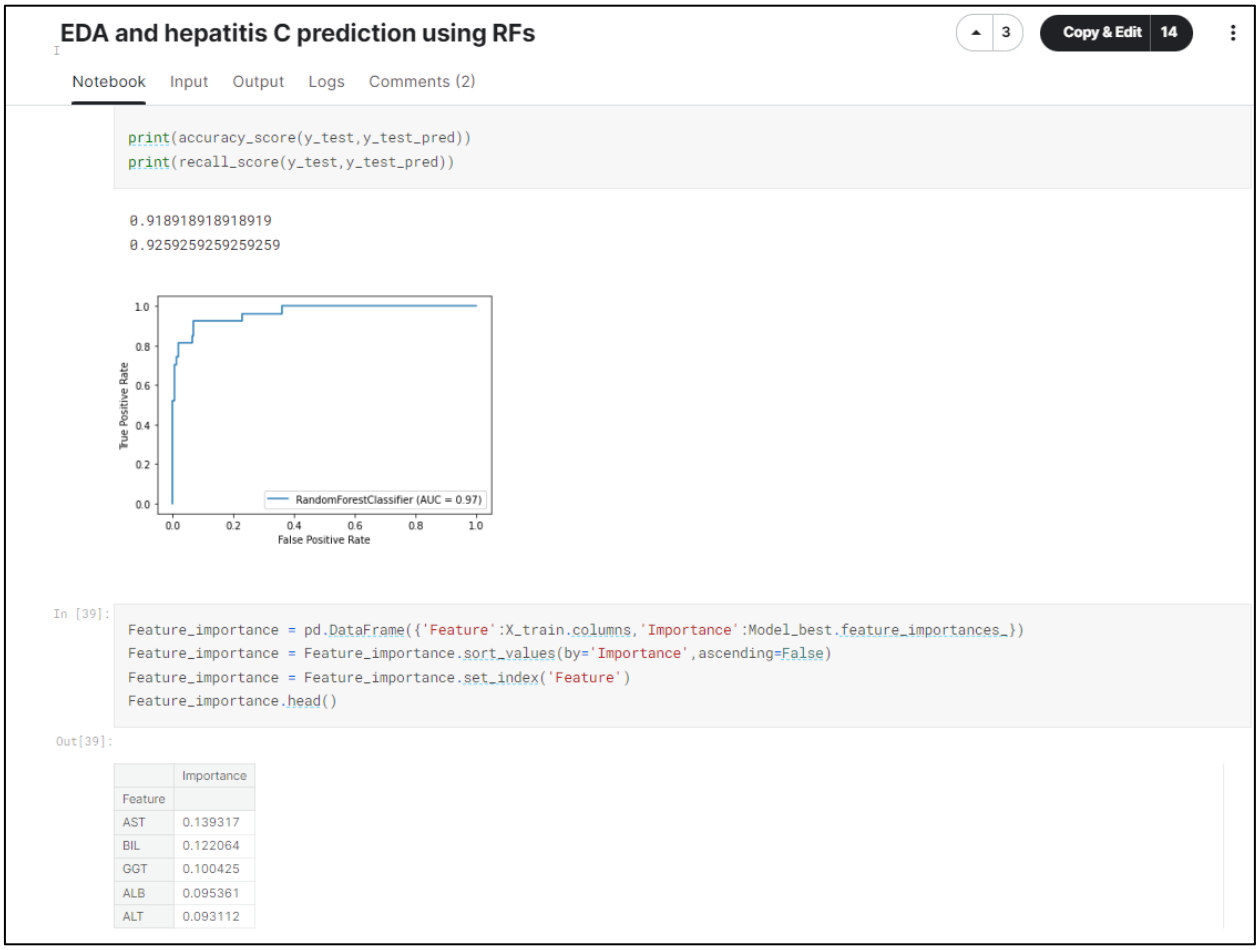


Рисунок 1.9 – Результати класифікації та таблиця важливості ознак інформаційної технології «EDA and hepatitis C prediction using RFs».

З метою поліпшення дослідження може бути корисним розглянути використання не лише однієї моделі машинного навчання, але й порівняти результати різних підходів. Також важливо більш докладно вивчити важливість кожної ознаки для забезпечення глибшого розуміння та оптимізації прогностичних моделей.

Наступною з інформаційних технологій на огляді є «Hepatitis C Prediction» (рис. 1.10).

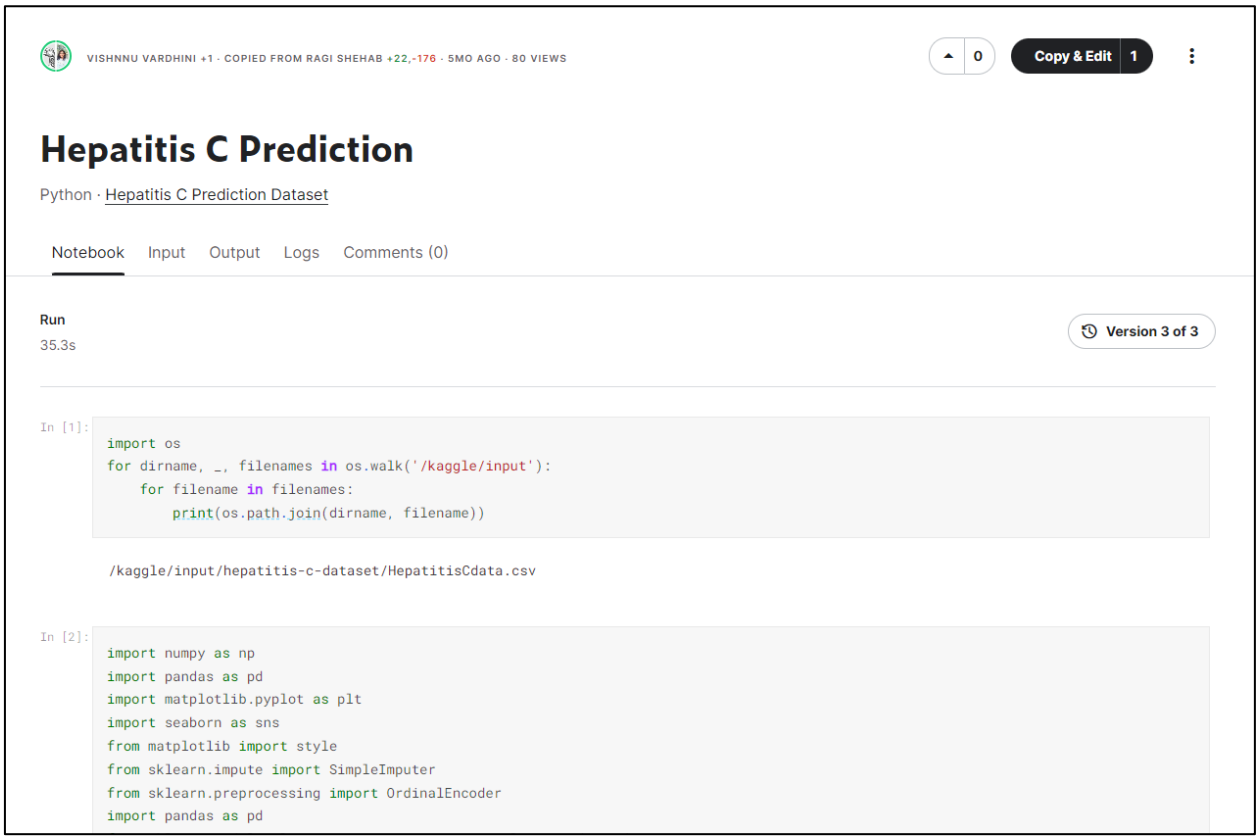


Рисунок 1.10 – Сторінка готового рішення «Hepatitis C Prediction».

Після ретельного аналізу цієї інформаційної технології було виявлено, що, так само, як і в минулому дослідженні, використовується лише одна модель машинного навчання — "Logistic Regression". Незважаючи на вражаючий показник точності, який склав 0.90 за метрикою r2_score, цей обмежений вибір моделі може внести обмеження у розгляді широкого спектру варіантів та можливостей (рис. 1.11) [12].

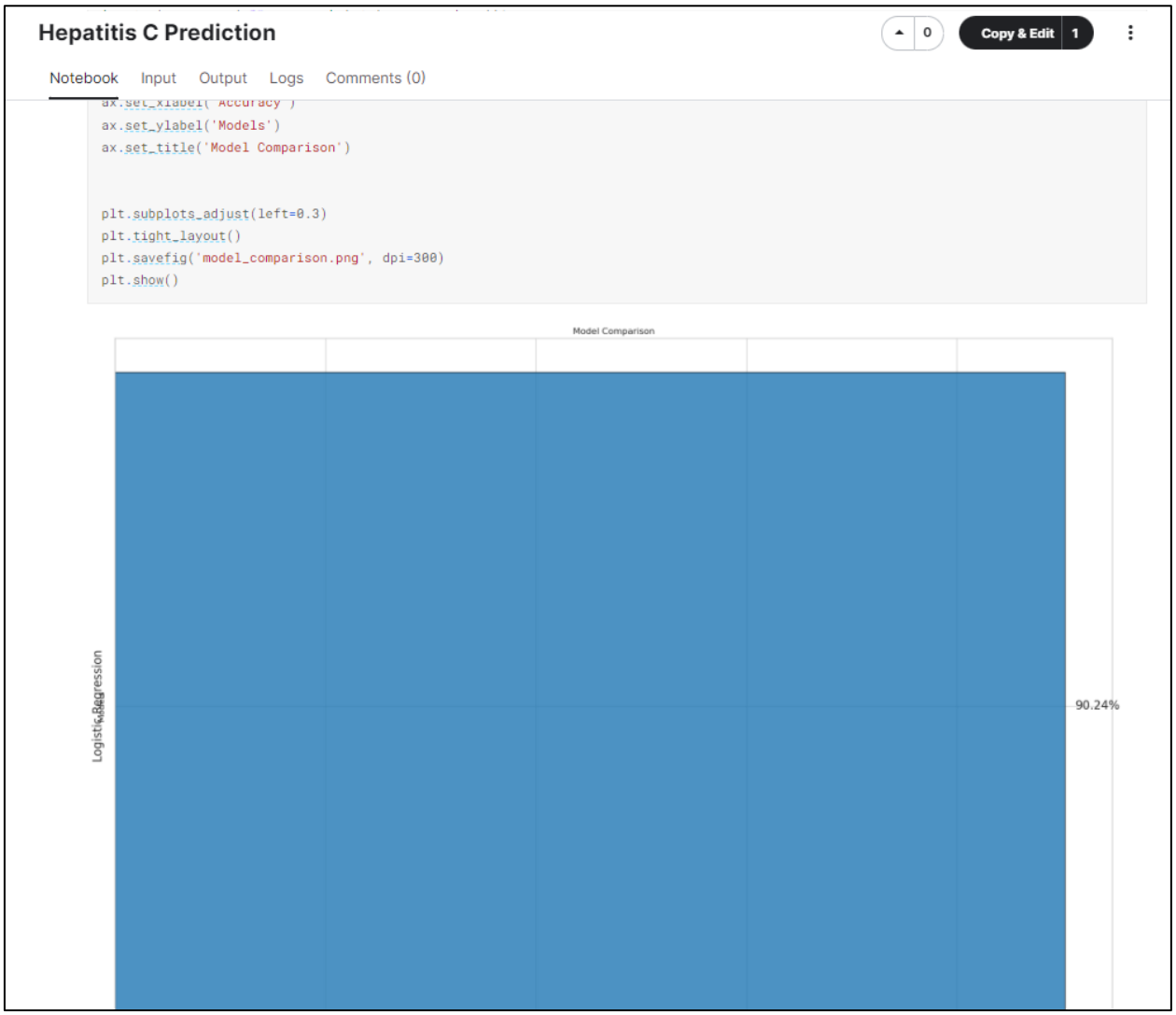


Рисунок 1.11 – Результати класифікації та таблиця важливості ознак інформаційної технології «Hepatitis C Prediction».

Для забезпечення більшої повноти та надійності дослідження рекомендується розглядати використання різних моделей машинного навчання. Це може допомогти виявити потенційні відмінності, які можуть бути важливими для повного розуміння та інтерпретації даних.

Додатково, більш глибокий аналіз та порівняння моделей може розкрити їхню ефективність у різних умовах та виявити оптимальні підходи до розв'язання конкретних завдань.

Останнім розглянутим рішенням буде «Hepatitis C Visualize Importance» (рис. 1.12).

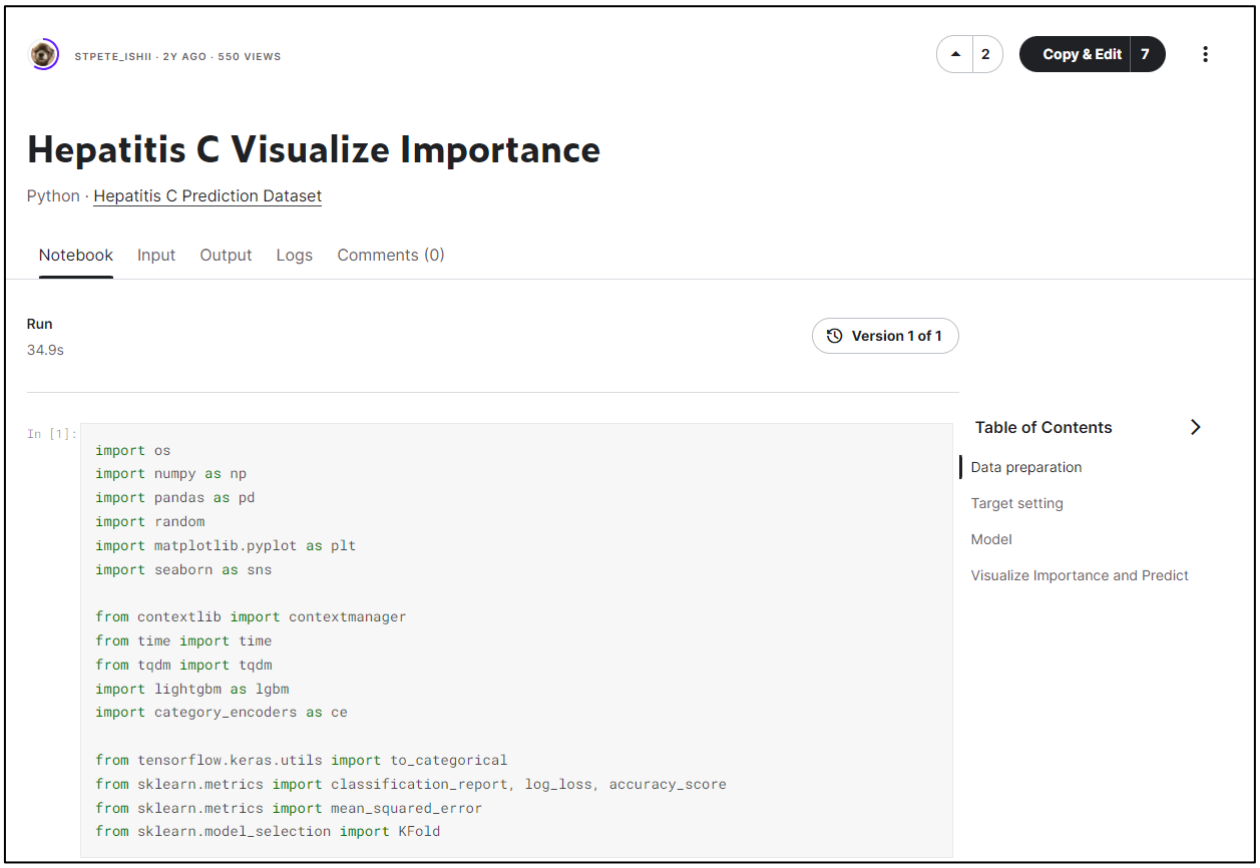


Рисунок 1.12 - Сторінка готового рішення «Hepatitis C Visualize Importance».

В розглянутому рішенні використовується модель машинного навчання LGBMClassifier, яка, за результатами, надає точність на рівні 0.88 за метрикою r2_score. Це вказує на те, що вибір іншої моделі привів до зменшення точності порівняно із попередніми результатами, де використовувалася модель Logistic Regression з точністю 0.92 (рис. 1.13) [13].

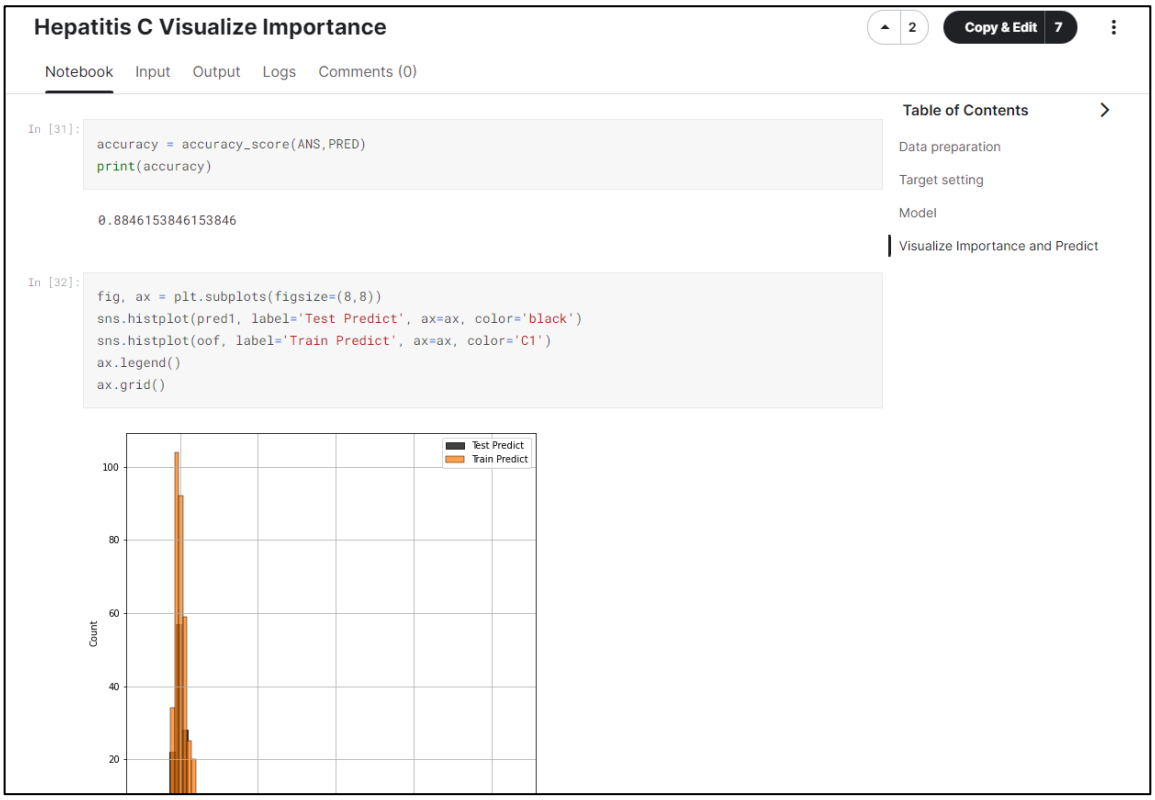


Рисунок 1.13 – Результати класифікації та таблиця важливості ознак інформаційної технології «Hepatitis C Visualize Importance».

Варто вказати на використання лише однієї моделі машинного навчання, що не лише обмежує, а й зводить на невеликий рівень розвідувальний аналіз.

Суттєвий недолік у розвиненні розвідувального аналізу породжує потребу у вдосконаленні та розширенні інструментарію для збору та аналізу даних, щоб забезпечити більш комплексний та ефективний підхід до вирішення завдань.

1.4 Висновки

Перший розділ дослідження присвячений глибокому аналізу предметної області, ретельному огляду існуючих аналогів і вивченню різноманітних технологій для виявлення гепатиту з метою обґрунтованого вибору найбільш відповідних для успішного вирішення поставленої завдання. В цьому розділі висвітлено ключові аспекти, що стосуються обраної тематики, зокрема

враховуючи актуальні тенденції та інноваційні рішення, які можуть вплинути на результати нашого дослідження.

Після проведених досліджень аналогів було визначено, що більшість з них приділяють мало уваги попередній обробці даних, використанню кількох моделей передбачення та пошуку аномалій, що буде враховано при розробленні інформаційної системи.

2 АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Вибір оптимального середовища розробки

Під час вибору інструментів для виконання поставленого завдання було обрано платформу Kaggle, визнавши її ефективність у вирішенні подібних завдань завдяки великому співтовариству фахівців, доступу до потужних обчислювальних ресурсів та інтегрованим інструментам для обробки даних. Ця платформа також надає можливість участі в конкурсах, що сприяє обміну ідеями та розвитку навичок у сфері аналізу даних. Впевненість у стабільності та надійності Kaggle робить її відмінним вибором для цього проекту [14].

Kaggle - це веб-платформа для проведення змагань з машинного навчання та аналізу даних. На Kaggle зібрані дані, завдання та спільнота фахівців з усього світу. Учасники можуть змагатися в різних випробуваннях, вирішувати завдання та долучатися до спільноти для обміну досвідом. Kaggle також надає інструменти для аналізу даних, візуалізації та роботи з моделями машинного навчання. Платформа активно використовується в галузі науки про дані та розв'язання реальних проблем за допомогою інноваційних підходів до аналізу і обробки інформації (рис. 2.1).

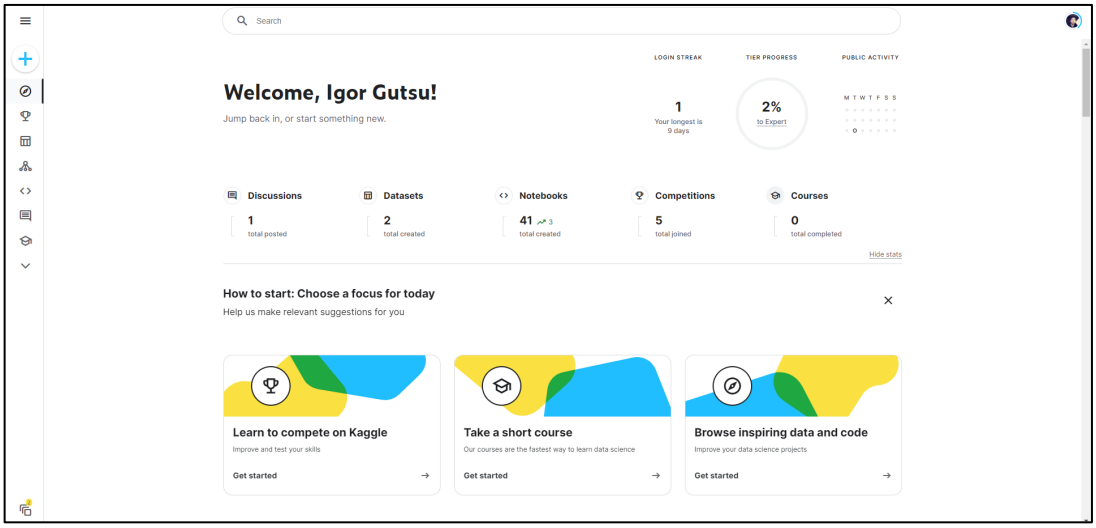


Рисунок 2.1 – Головна сторінка Kaggle

На Kaggle користувачі можуть завантажувати власні дані або використовувати надані набори даних для розв'язання конкретних завдань. Платформа підтримує різні мови програмування, що дозволяє учасникам використовувати їх улюблені інструменти та бібліотеки для розробки моделей.

Крім того, Kaggle надає можливість створення та обговорення ядер (kernels), які є віртуальними робочими середовищами, де користувачі можуть виконувати код, аналізувати дані та ділитися своїми результатами та дослідженнями з іншими.

За допомогою Kaggle, фахівці отримують можливість не лише змагатися в конкуренціях, а й вдосконалювати свої навички, спілкуватися з колегами по галузі та вчитися на прикладах реальних проектів. Крім того, платформа дозволяє компаніям проводити спільні змагання та залучати талановитих аналітиків для вирішення своїх завдань у сфері обробки даних та машинного навчання [15].

Конкурси на Kaggle є однією з ключових функцій платформи, що привертає учасників з усього світу. Ці змагання часто спонсоруються компаніями або організаціями, які ставлять перед учасниками конкретні завдання у галузі машинного навчання, аналізу даних чи інших суміжних областей (рис. 2.2).

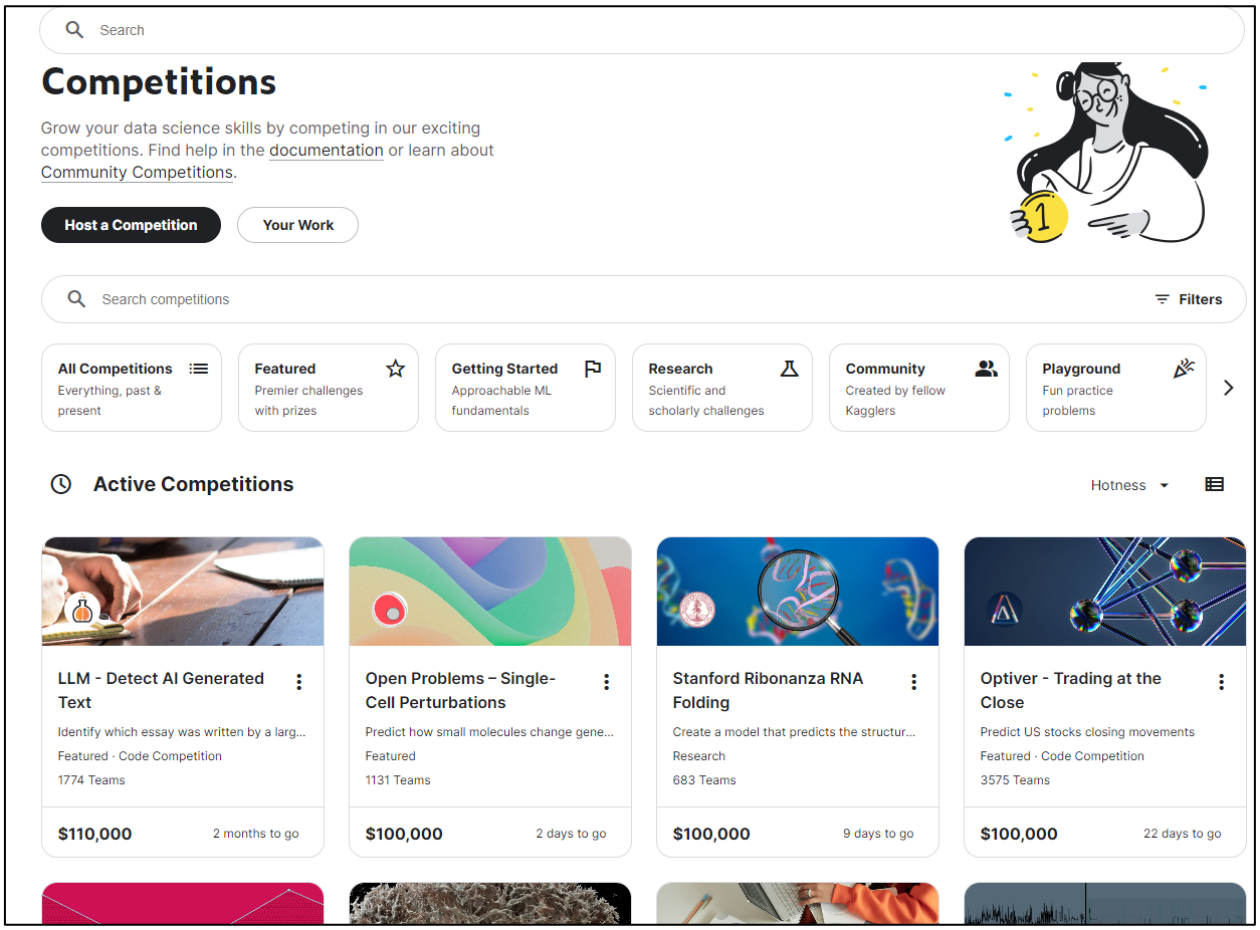


Рисунок 2.2 – Сторінка конкурсів у Kaggle.

Учасники формують свої команди або працюють індивідуально, намагаючись розробити моделі, які найточніше вирішать поставлену задачу. Зазвичай, конкурси тривають кілька місяців, і переможці отримують винагороду у вигляді грошових призів, можливостей працевлаштування або інших стимулів, що залежать від спонсорів.

Такі конкурси не лише сприяють розвитку нових та ефективних методів у сфері машинного навчання, а й допомагають вирішувати реальні проблеми, які стоять перед бізнесом та науковою спільнотою. Участь в конкурсах також дозволяє учасникам побачити різноманітні підходи до вирішення тієї ж задачі та вивчити найкращі практики у галузі аналізу даних.

Як мову програмування було обрано Python, оскільки він має найбільшу кількість інструментів та бібліотек призначених для побудови та використання моделей машинного навчання [16].

Python - це високорівнева, інтерпретована мова програмування загального призначення. Вона була створена Guido van Rossum і вперше випущена у 1991 році. Python має простий синтаксис, що дозволяє легко вивчати мову новачкам. Мова підтримує об'єктно-орієнтоване, імперативне та функціональне програмування.

Основні риси Python включають динамічне визначення типів, автоматичне управління пам'яттю та велику стандартну бібліотеку. Мова активно використовується для веб-розробки, аналізу даних, штучного інтелекту, машинного навчання та інших областей програмування. Розробники також відзначають широку підтримку спільноти та доступність безкоштовних ресурсів для вивчення Python.

Python є мовою з великою активністю розвитку, з новими версіями, що надають покращення та нові можливості. Мова відома своєю простотою, читабельністю коду та можливістю швидкої розробки програм. Python дозволяє писати компактний код, що сприяє продуктивності розробників [17].

Інтерпретованість Python робить його платформонезалежним, що означає, що програми, написані на Python, можуть працювати на різних операційних системах без змін. Python підтримує багато парадигм програмування, що дає розробникам велику гнучкість у виборі стилю програмування.

Багата стандартна бібліотека Python забезпечує велику кількість готових модулів для різних задач, що спрощує розробку і дозволяє швидше створювати функціональні програми. Подальший успіх Python пов'язаний з його використанням у великому спектрі галузей, від науки до веб-розробки та великих корпоративних систем.

Загальна філософія мови, відома як "The Zen of Python", визначає керівні принципи, які сприяють чистоті та ефективності коду. Python став популярним вибором для початківців і досвідчених розробників завдяки своїй простоті та потужності.

Python є популярним вибором у сфері машинного навчання завдяки своїй простоті та широкому спектру бібліотек, спеціалізованих на аналізі даних та машинному навчанні. Бібліотеки, такі як NumPy, Pandas, і Matplotlib, надають зручний інтерфейс для роботи з даними, а scikit-learn та TensorFlow дозволяють легко використовувати алгоритми машинного навчання [18].

Простий синтаксис Python робить його доступним для новачків у галузі машинного навчання, а велика та активна спільнота розробників допомагає вирішувати проблеми та обмінюватися ідеями. Більше того, багато дослідників та практикуючих фахівців в машинному навчанні обирають Python для своїх проектів, що сприяє обміну знаннями та розвитку галузі.

Можливість використовувати Python для розробки прототипів та експериментів зменшує час, необхідний для створення та вдосконалення моделей машинного навчання. Python також інтегрується з іншими популярними мовами та інструментами, що робить його варіантом для великих та складних проектів у сфері штучного інтелекту та аналізу даних.

Оскільки це дослідження є задачею класифікації, доцільним буде обрати такі моделі машинного навчання для виконання поставленої задачі є Logistic Regression, Random Forest Classifier, Gradient Boosting, SVM, Decision Tree Classifier.

2.2 Модель машинного навчання «Decision Tree»

Дерево рішень (Decision Tree) — це модель машинного навчання, яка використовується для прийняття рішень на основі умовних правил. Це деревоподібна структура, де кожен вузол представляє собою тест, що визначає певну характеристику вхідних даних, а кожне гілля вузла вказує на можливий результат цього тесту [19].

Основна ідея рішучого дерева полягає у рекурсивному поділі набору даних на підмножини на основі атрибутів, що мають найбільший вплив на прогнозовану змінну. Процес тренування дерева включає в себе вибір

оптимальних атрибутів для поділу та побудову дерева досягнення критерію зупинки, такого як максимальна глибина дерева або мінімальна кількість вибірок у вузлі.

На рисунку 2.3 показано принцип роботи алгоритму.

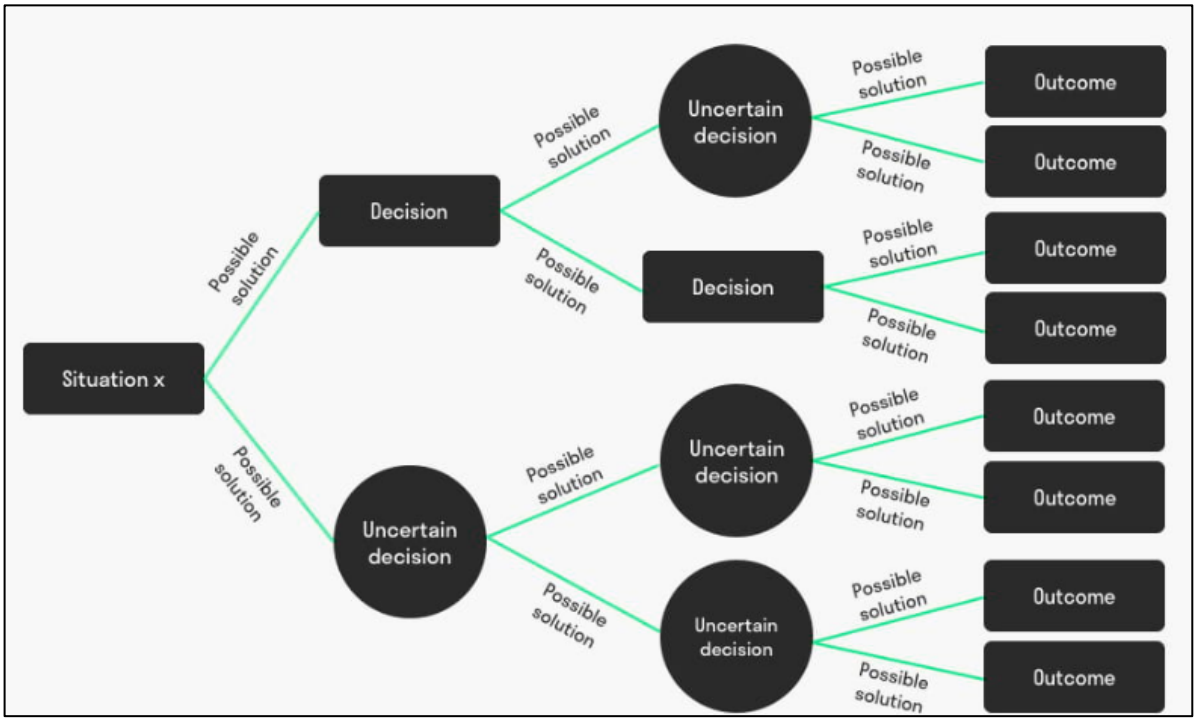


Рисунок 2.3 – Приклад роботи алгоритму Decision Tree.

Однією з ключових переваг рішучих дерев є їх інтерпретованість. Легко зрозуміти, як воно приймає рішення, оскільки це можна представити у вигляді простого "якісного" дерева. Однак дерева можуть бути схильні до перенавчання, особливо, якщо їх не обмежувати.

Існують різні алгоритми для побудови рішучих дерев, такі як CART (Classification and Regression Trees), ID3 (Iterative Dichotomiser 3), C4.5, та інші. Кожен з них може мати свої особливості та параметри.

Рішучі дерева можуть використовуватися для класифікації (прогнозування категорій) та регресії (прогнозування числових значень). Ці моделі застосовуються в різних областях, таких як медицина, фінанси, маркетинг та інші, де важливо приймати рішення на основі великої кількості вхідних даних [19].

Важливо відзначити, що Decision Tree допомагає уникнути проблеми перенавчання, завдяки вбудованому механізму обмеження глибини дерева та інших параметрів. Це дозволяє підтримувати стабільність моделі і уникнути втрати узагальнювальної здатності на нових даних.

Крім того, Decision Tree є дуже гнучким алгоритмом, оскільки може легко адаптуватися до змін у вхідних даних та умовах завдань. Ця гнучкість робить його ефективним в різноманітних галузях, від медицини до фінансів, де важливо миттєво реагувати на зміни у даних та отримувати точні прогнози.

2.3 Модель машинного навчання «Support Vector Machine Classifier»

SVM - алгоритм для класифікації та регресії. Основна мета - знайти гіперплощину, що максимізує відстань між класами. Гіперплощина оптимізується для найкращого розділення даних, де маржа представляє собою відстань від гіперплощини до найближчого екземпляра даних. SVM шукає найбільшу маржу для кращої узагальненості на нових даних.

SVM використовується для вирішення як лінійних, так і не лінійних задач. У випадку не лінійних просторів дані переносяться в вищий розмір, де можлива ефективна лінійна роздільність. Особливість SVM полягає в використанні опорних векторів — це дані, що визначають положення межі рішення. Метод здатний ефективно працювати з невеликою кількістю опорних векторів, що полегшує обробку великих обсягів даних. Крім того, SVM добре справляється з уникненням перенавчання завдяки регулюючим параметрам, які контролюють маржу та вплив даних на модель [20].

SVM виявляється ефективним у випадках, коли дані мають складні форми розділення між класами або коли вони знаходяться в високорозмірних просторах. Метрика, за якою відбувається оптимізація, дозволяє моделі знаходити оптимальні рішення, що робить SVM корисним у багатьох галузях, включаючи комп'ютерне зорове сприйняття, біоінформатику та фінансовий аналіз.

Недоліком може бути складність в роботі з великою кількістю ознак або об'ємних даних, а також чутливість до шуму в даних. Не зважаючи на це, правильно налаштований SVM може досягати вражаючих результатів у вирішенні складних задач класифікації та регресії.

SVM також може бути використаний для вирішення задачі класифікації з більш ніж двома класами, використовуючи метод один проти всіх (One-vs-All). У цьому випадку для кожного класу створюється окрема модель, яка відрізняє цей клас від всіх інших. При класифікації нового прикладу вхідних даних відбувається голосування між всіма моделями, і клас з найбільшою кількістю голосів визначається як кінцевий результат.

Інша важлива риса SVM - можливість використовувати різні ядра (kernels), такі як лінійне, поліноміальне, та радіально-базисне (RBF) ядро. Це дозволяє адаптувати модель до різних типів даних і форм розділення. Загалом, SVM залишається популярним інструментом у сфері машинного навчання завдяки своїй ефективності та гнучкості в різноманітних завданнях [20].

Важливою характеристикою SVM є його здатність ефективно працювати з невеликими обсягами даних. Це зроблено завдяки тому, що при навчанні моделі враховуються лише опорні вектори, тобто дані, які визначають межу прийняття рішення. Такий підхід дозволяє уникнути перенавчання та робить модель більш стійкою до змін у вихідних даних.

Крім того, SVM може бути успішно використаний у задачах виявлення аномалій, де основна маса даних представлена одним класом, а відхилення від цього класу розглядається як аномалія. Це робить SVM важливим інструментом для вирішення завдань, пов'язаних із виявленням непередбачених аномальних подій.

У підсумку, SVM є потужним і гнучким алгоритмом, який може вирішувати різноманітні завдання машинного навчання, зокрема в області класифікації та регресії, і залишається популярним інструментом серед дослідників та практиків у галузі інтелектуального аналізу даних.

2.4 Модель машинного навчання «Gradient Boosting Classifier»

Gradient Boosting Classifier (GBC) - це алгоритм машинного навчання, який належить до сімейства ансамблевих методів. Основна ідея полягає в тому, щоб створювати послідовність слабких класифікаторів, зазвичай рішаючих дерев, і поетапно коригувати їх прогнози для поліпшення точності моделі. Градієнтний бустінг використовує градієнтні методи оптимізації для мінімізації функції втрат.

Під час навчання GBC будує новий класифікатор, який виправляє помилки, зроблені попередніми класифікаторами у послідовності. Цей новий класифікатор додається до моделі з вагами, залежними від того, наскільки добре він виправляє помилки. Важливою характеристикою є введення концепції кроку (learning rate), який контролює внесок кожного нового класифікатора.

Градієнтний бустінг широко використовується для вирішення задач класифікації та регресії і відомий своєю високою точністю. Проте важливо враховувати, що цей метод може бути схильним до перенавчання, і налаштування гіперпараметрів грає ключову роль у досягненні оптимальної продуктивності моделі [21].

Градієнтний бустінг використовується для створення сильної прогнозної моделі за допомогою комбінації багатьох слабких моделей. При кожному новому кроці, модель фокусується на тих прикладах, які попередні моделі погано класифікували, і намагається покращити їх прогнози. Цей процес ітеративно повторюється, що призводить до створення потужної моделі з високою точністю.

Однією з важливих переваг градієнтного бустінгу є його здатність автоматично виявляти важливі функції для класифікації або регресії, що робить його ефективним для роботи з складними даними. Також, він добре впорається з великою кількістю даних та може адаптуватися до різноманітних завдань машинного навчання.

Важливо враховувати, що градієнтний бустінг витребовує обчислювальних ресурсів і тривалого процесу навчання, але при правильному налаштуванні параметрів він може забезпечити вражаючі результати в різних областях застосування, таких як фінанси, медицина чи аналіз даних.

Градієнтний бустінг також володіє деякими вбудованими механізмами регуляризації, які допомагають управляти склонністю моделі до перенавчання. Це робить алгоритм менш чутливим до випадкових викидів або шуму в даних, сприяючи стабільній роботі в реальних умовах.

При роботі з категоріальними ознаками градієнтний бустінг може використовувати кодування, такі як кодування категорій з вагами (Weight of Evidence) або кодування категорій за допомогою внутрішніх змінних (Target Encoding). Це дозволяє алгоритму ефективно враховувати категоріальні дані, що є важливим аспектом в багатьох реальних задачах [21].

Загалом, градієнтний бустінг став популярним і ефективним інструментом в сфері машинного навчання, завдяки його здатності до вирішення різноманітних завдань та надійності в роботі з різними типами даних.

2.5 Модель машинного навчання «Random Forest Classifier»

Random Forest Classifier - це алгоритм машинного навчання, який використовується для класифікації даних. Він є типом ансамблю дерев рішень, де кілька дерев приймають рішення разом для забезпечення більш точного та стійкого результату. Алгоритм використовує випадковий вибір підмножини ознак для побудови кожного дерева, що допомагає уникнути перенавчання та забезпечує різноманітність в ансамблі.

Random Forest працює, об'єднуючи прогнози кількох дерев рішень, які обробляють різні частини даних. Коли потрібно зробити прогноз для нового прикладу, кожне дерево в ансамблі вносить свій внесок, і результат визначається голосуванням чи узгодженням прогнозів дерев [22].

Основна ідея Random Forest полягає в тому, що він комбінує сильні сторони багатьох дерев рішень, зменшуючи при цьому їхні недоліки, такі як перенавчання. Це дозволяє алгоритму ефективно працювати з різноманітними типами даних і забезпечує високий рівень точності при класифікації.

Random Forest також може надавати важливість ознакам, враховуючи те, як часто вони використовуються для прийняття рішення в ансамблі дерев. Це може бути корисно для аналізу впливу конкретних ознак на прогнози моделі.

Random Forest Classifier є особливо ефективним в обробці великих обсягів даних і враховує велику кількість можливих шляхів прийняття рішень. Його здатність працювати з великою кількістю ознак робить його придатним для різних завдань, включаючи задачі з обробки текстів, зображень та числових даних.

Однією з важливих переваг Random Forest є його стійкість до перенавчання, оскільки кожне дерево рішень побудоване на випадковому підмножині даних та ознак. Це робить його менш вразливим до шуму та випадкових аномалій у навчальних даних [22].

Крім того, Random Forest дозволяє оцінювати важливість ознак у завданні класифікації. Це надає можливість отримати інсайти щодо того, які аспекти даних найбільше впливають на результати моделі.

Random Forest Classifier також має вбудовані механізми виявлення аномалій та важкозрозумілих зв'язків у даних. Це може бути особливо корисним у випадках, коли взаємодії між ознаками складні чи коли деякі з ознак мають велику важливість лише в конкретних умовах.

Крім того, Random Forest може ефективно працювати з відсутніми значеннями в даних, оскільки він може працювати з неповними або неперевіреними наборами даних без значного втрати точності.

Ще однією важливою характеристикою є можливість використання Random Forest для вирішення завдань регресії, де модель може прогнозувати числове значення замість класифікації. Це робить його універсальним

інструментом, здатним вирішувати широкий спектр завдань у сфері прогнозування.

Загалом, Random Forest Classifier вирізняється високою гнучкістю, надійністю та здатністю до роботи з різноманітними типами даних, роблячи його одним із популярних інструментів у сфері машинного навчання.

2.6 Модель машинного навчання «Logistic Regression»

Логістична регресія - це статистичний метод, що використовується для моделювання ймовірностей виникнення події залежно від одного чи кількох змінних. Назва "логістична" вказує на використання логістичної функції для моделювання ймовірностей.

Мета логістичної регресії полягає в тому, щоб знаходити оптимальні ваги для вхідних змінних так, щоб максимізувати ймовірність вірного класифікації. Логістична регресія широко використовується в задачах бінарної класифікації, де потрібно визначити, до якого з двох класів належить об'єкт.

Модель логістичної регресії використовує логістичну функцію, яка перетворює лінійну комбінацію вхідних змінних і їх ваг у значення між 0 і 1. Логістична функція дозволяє моделі виражати ймовірність події, що належить до певного класу. Ваги моделі оптимізуються за допомогою методу максимальної ймовірності або інших оптимізаційних методів [23].

У випадку бінарної класифікації, логістична регресія визначає границю прийняття рішення, де значення функції більше або рівне 0.5 призначається одному класу, а значення менше 0.5 - іншому.

Логістична регресія має широкий спектр застосувань, включаючи медичну діагностику, фінансовий аналіз, соціальні науки та інші галузі, де важливо прогнозувати ймовірність виникнення певної події.

Логістична регресія є ефективним методом, особливо там, де даним є два взаємовиключних класи. Вона часто порівнюється з лінійною регресією, але вона підходить для задач класифікації, де вихідна змінна є категоріальною.

Однією з ключових переваг логістичної регресії є можливість отримання ймовірностей для приналежності до класу, що дозволяє здійснювати більш гнучкий аналіз та враховувати невизначеність [23].

Метод логістичної регресії добре пристосований для роботи з різними видами змінних, включаючи кількісні та категоріальні. Його можна легко розширити для включення взаємодій між змінними, що робить його корисним для моделювання складних взаємодій в даних.

Також важливо відзначити, що логістична регресія добре справляється з великими наборами даних та не вимагає великої обчислювальної потужності. Це робить її популярним вибором для широкого спектру завдань машинного навчання та статистичного моделювання.

Важливим етапом в роботі з логістичною регресією є валідація моделі, яка дозволяє оцінити її точність та генералізаційну здатність до нових даних. Це допомагає уникнути перенавчання та підтримує надійність моделі в реальних сценаріях застосування.

Логістична регресія є легко інтерпретованою моделлю, оскільки ваги, які вона використовує для кожної вхідної змінної, можуть бути розглянуті як вплив цих змінних на ймовірність належності до певного класу. Це дозволяє дослідникам та фахівцям легко зрозуміти внесок кожної змінної у прогнозування.

У додаток до бінарної класифікації, логістична регресія також може бути розширена для вирішення багатокласових задач класифікації. Одним із способів реалізації цього є використання одного проти всіх (One-vs-All або OvA) або одного проти одного (One-vs-One або OvO) підходів [24].

Важливою особливістю логістичної регресії є вона стійкість до викидів (outliers) у даних, оскільки вона використовує ймовірнісний підхід, що дозволяє їй працювати добре навіть у випадках, коли дані містять шум чи аномалії.

2.7 Висновки

У цьому розділі розглянуто різноманітні аспекти сучасних інформаційних технологій, які можуть бути застосовані у контексті вирішення визначеної задачі. Зокрема, проаналізовано ключові характеристики та переваги різних підходів до машинного навчання, що включають у себе нейронні мережі, методи класифікації та кластеризації даних.

Вибір оптимальної моделі машинного навчання здійснювався на підставі врахування специфіки завдання, обсягу та характеру наявних даних. Кожна обрана технологія вивчалася з точки зору її придатності для розв'язання конкретних завдань, а також врахування ймовірних переваг та обмежень.

В результаті проведеного в цьому розділі дослідження було визначено що найкращими моделями класифікації для виконання поставленої задачі є Logistic Regression, Random Forest Classifier, Gradient Boosting, SVM, Decision Tree Classifier.

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ СТАНУ ХВОРИХ ТА ГЕПАТИТ

3.1 Аналіз обраних бібліотек та побудова схеми алгоритму

Першим кроком у виконанні поставленої задачі було розроблено певну послідовність дій, яких необхідно дотримуватись для успішного виконання завдання.

На рисунку 3.1 показано схему алгоритму який буде задіяно в подальшій роботі для передбачення стану хворих на гепатит.

- Показана схема алгоритму включає в себе такі кроки:
- Імпорт бібліотек;
- Імпорт датасету;
- Розвідувальний аналіз;
- Визначення цільової ознаки класифікації;
- Розбиття датасету на тренувальний та тестовий;
- Побудова моделей класифікації;
- Оцінка та порівняння результатів класифікації;
- У випадку, якщо результати не задовільняють потреби, провести заміну параметрів моделей та додаткову обробку даних і повернутись до кроку «Розбиття датасету на тренувальний та тестовий».

Дотримання визначеного алгоритму забезпечить успішне виконання задачі.

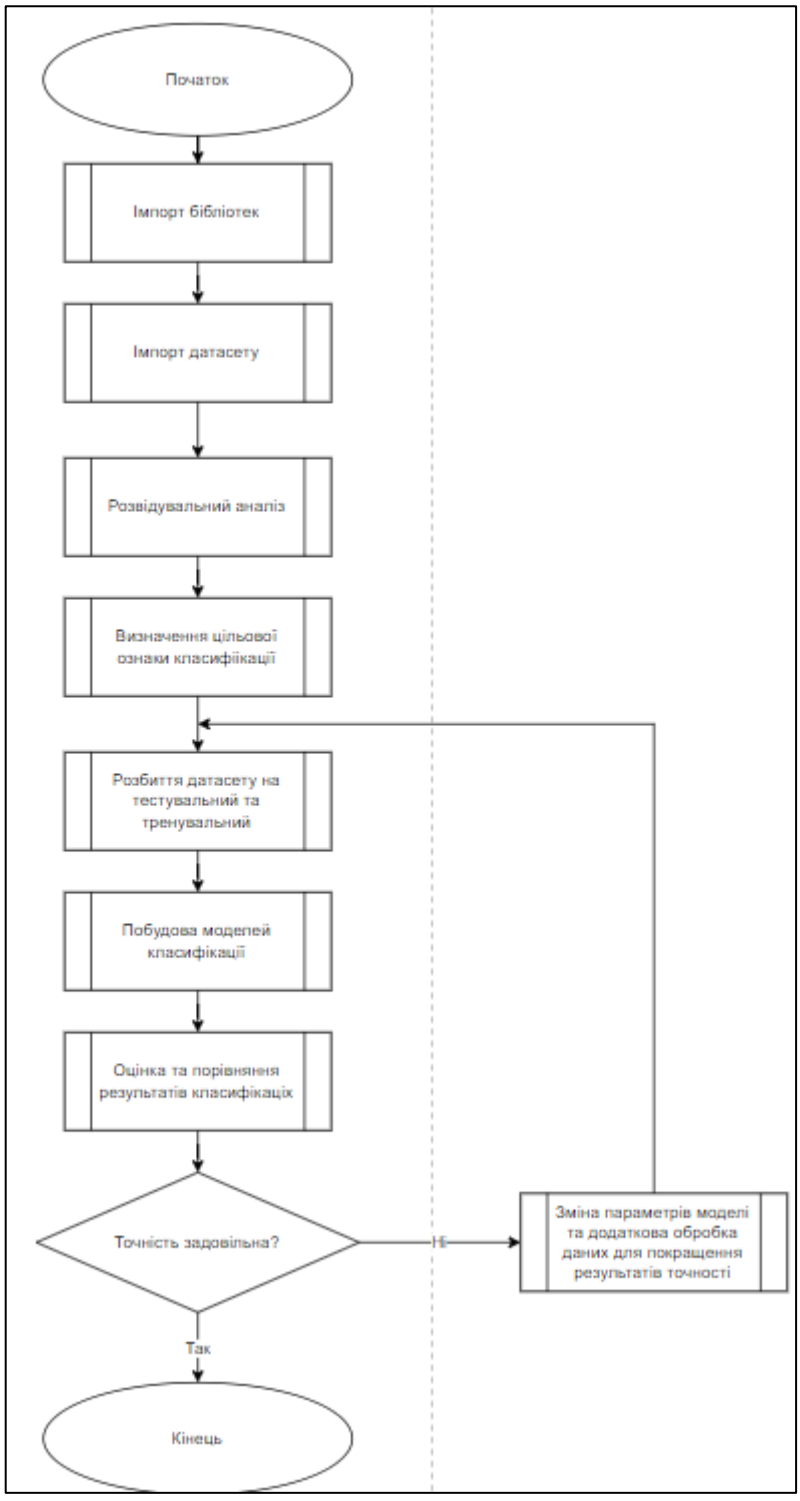


Рисунок 3.1 – Схема роботи алгоритму.

Першим кроком для виконання поставленого завдання було створено новий ноутбук у Kaggle, з використанням датасету «Hepatitis C Prediction».

Далі, було обрано та імпортовано необхідний перелік бібліотек. Ці бібліотеки будуть використовуватися для проведення аналізу даних, дозволяючи ефективно виконати завдання.

Зазначений перелік бібліотек включає інструменти, які забезпечують широкі можливості для обробки та вивчення даних, що дозволить здійснити більш глибокий аналіз з наявних даних (рис. 3.1).

```
[450]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import style
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OrdinalEncoder
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import warnings
import joblib

style.use("fivethirtyeight")
```

Рисунок 3.1 – Список імпортованих бібліотек.

Бібліотека NumPy в Python є потужним інструментом для виконання числових обчислень. Забезпечує зручні структури даних, такі як масиви та матриці, і високопродуктивні функції для їх обробки. NumPy знаходить широке використання в областях наукових обчислень, обробки сигналів та статистики. При імпортуванні використовується псевдонім np для скорочення викликів функцій та методів.

Бібліотека Pandas, також використовувана в імпорті під іменем pd, є потужним інструментом для маніпулювання та аналізу даних. Забезпечує структури даних, такі як DataFrame, які спрощують обробку та аналіз табличних даних. Pandas використовується для фільтрації, групування,

об'єднання та агрегації даних, роблячи його невід'ємною частиною роботи з даними в Python [25].

Matplotlib та Seaborn є бібліотеками для візуалізації даних. Matplotlib дозволяє створювати різноманітні графіки, діаграми та інші візуальні елементи. Seaborn, ґрунтуючись на Matplotlib, надає високорівневий інтерфейс для створення статистично змістовних графіків. Ці бібліотеки важливі для відображення та розуміння залежностей в даних [26].

Бібліотеки scikit-learn та SciPy пропонують інструменти для машинного навчання та наукових обчислень. Scikit-learn містить реалізації різноманітних моделей машинного навчання, таких як логістична регресія та випадковий ліс. SciPy, серед іншого, містить інструменти для обробки статистичних даних, такі як функція zscore, яка використовується для стандартизації даних за допомогою зсуву та масштабування [27-28].

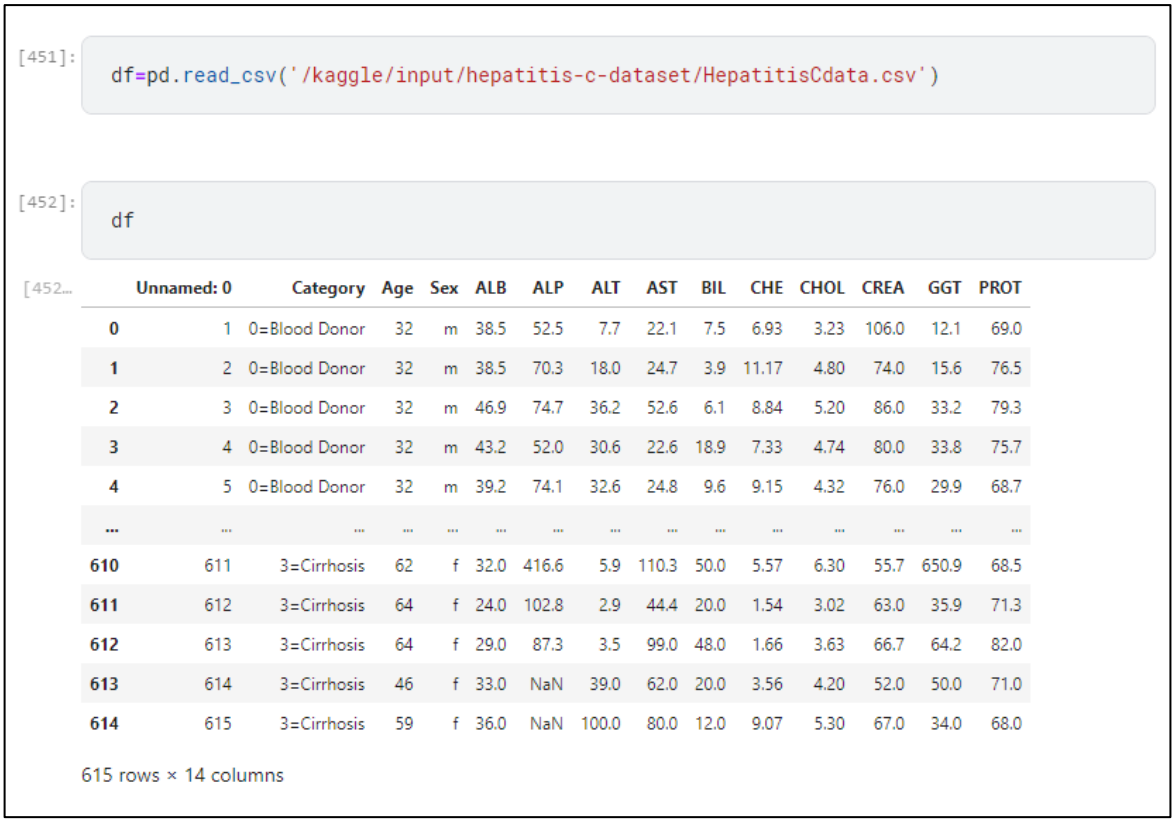
Бібліотека warnings в Python використовується для керування попередженнями в програмах. Вона надає можливість управляти виведенням та приглушувати певні види попереджень за допомогою контекст-менеджера warnings.filterwarnings(). Це корисно для управління виведенням попереджень та забезпечення чистого виведення програми [29].

Бібліотека joblib використовується для ефективного збереження та завантаження об'єктів Python. Вона дозволяє кешувати результати викликів функцій для подальшого використання, що особливо корисно в задачах обчислювального навчання та обробки даних. joblib сприяє прискоренню обчислень шляхом уникнення повторного обчислення та забезпечення зручного зберігання обчислювальних результатів[30].

3.2 Підготовка даних до розвідувального аналізу

Першим етапом у підготовці даних для розвідувального аналізу було імпортовано датасет. Після цього було виведено його на екран для детального огляду та аналізу. Це дозволило отримати загальне уявлення про структуру та

зміст набору даних, і визначити можливі шляхи оптимізації та обробки інформації перед подальшим аналізом (рис. 3.2).



```
[451]: df=pd.read_csv('/kaggle/input/hepatitis-c-dataset/HepatitisCdata.csv')
```

```
[452]: df
```

[452...

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
0	1	0=Blood Donor	32	m	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0
1	2	0=Blood Donor	32	m	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5
2	3	0=Blood Donor	32	m	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3
3	4	0=Blood Donor	32	m	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7
4	5	0=Blood Donor	32	m	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
610	611	3=Cirrhosis	62	f	32.0	416.6	5.9	110.3	50.0	5.57	6.30	55.7	650.9	68.5
611	612	3=Cirrhosis	64	f	24.0	102.8	2.9	44.4	20.0	1.54	3.02	63.0	35.9	71.3
612	613	3=Cirrhosis	64	f	29.0	87.3	3.5	99.0	48.0	1.66	3.63	66.7	64.2	82.0
613	614	3=Cirrhosis	46	f	33.0	NaN	39.0	62.0	20.0	3.56	4.20	52.0	50.0	71.0
614	615	3=Cirrhosis	59	f	36.0	NaN	100.0	80.0	12.0	9.07	5.30	67.0	34.0	68.0

615 rows × 14 columns

Рисунок 3.2 – Імпорт та огляд датасету.

Після цього проведено аналіз датасету на наявність нульових значень. Цей етап дозволяє виявити пропуски в даних та визначити необхідні кроки для їхнього усунення або обробки (рис.3.3).

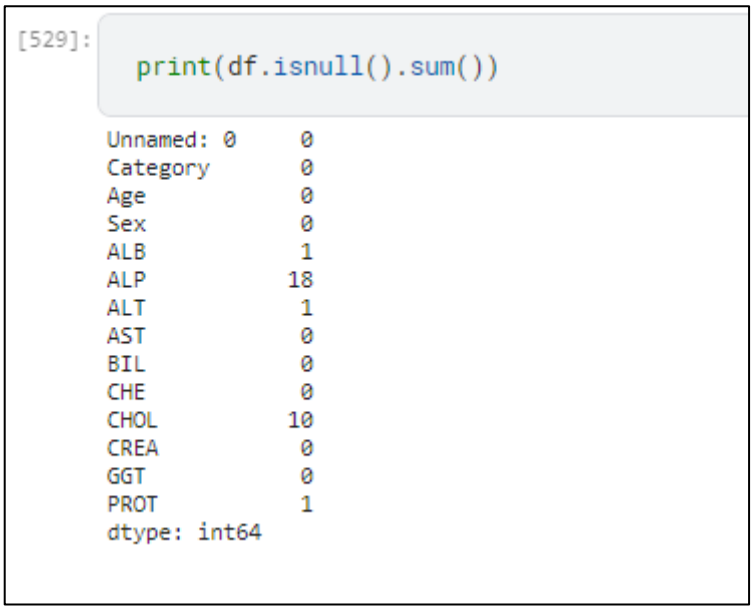


Рисунок 3.3 – Виведення кількості пустих значень.

З використанням функції "mean" нульові значення було замінено на середнє арифметичне їхніх сусідніх значень. Цей підхід спрямований на уникнення можливих неточностей та забезпечення більш стабільного та репрезентативного набору даних для подальшого аналізу (рис. 3.4).

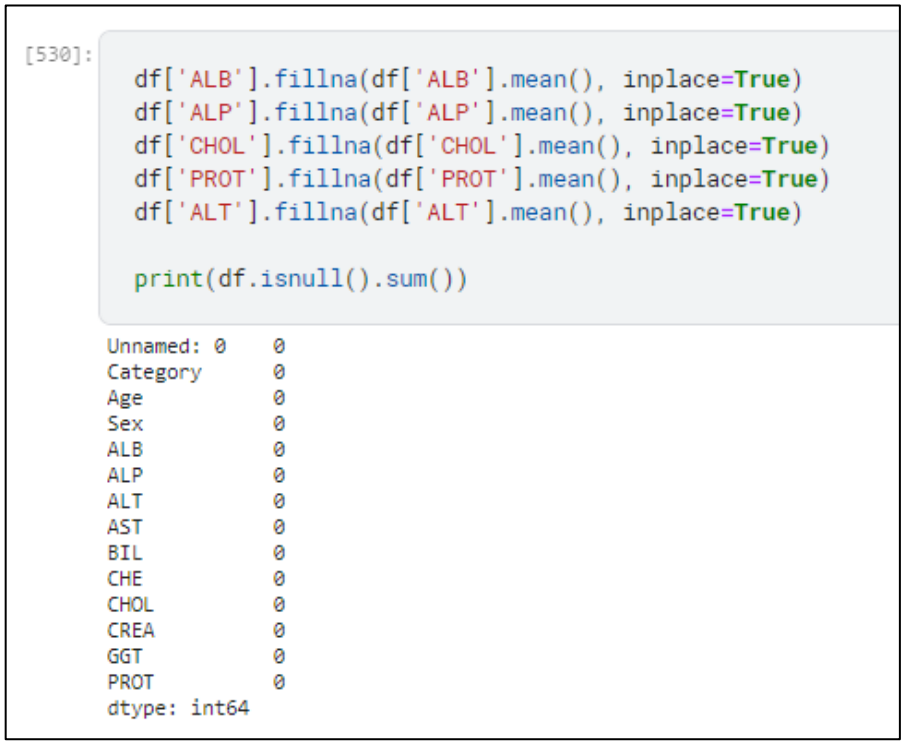


Рисунок 3.4 – Код заміни нулевих значень

На основі рисунку 3.4 можна зазначити, що нульові значення в датасеті більше не існує. Однак, наступним кроком буде виявлення та видалення аномальних значень в наборі даних. Цей процес спрямований на покращення точності та надійності аналізу, усунення потенційних викривлень та забезпечення якісної підготовки даних для подальших етапів дослідження (рис. 3.5).

```
]:
from scipy.stats import zscore
# Визначення порогу для Z-оцінки (зазвичай поріг 3 вважається значущим)
z_threshold = 3

# Обчислення Z-оцінок для кожної колонки
z_scores = zscore(df.select_dtypes(include=['float64']))

# Виявлення аномалій
anomalies = (z_scores > z_threshold).any(axis=1)

# Виведення кількості та видалення аномалій
print("Кількість аномалій:", anomalies.sum())
df = df[~anomalies].reset_index(drop=True)

Кількість аномалій: 53
```

Рисунок 3.5 – Пошук та видалення аномальних значень.

У зв'язку з тим, що датасет включає дані про здорових осіб та пацієнтів із фіброзом, цирозом та гепатитом, було вирішено об'єднати їх у дві категорії: "здорові" та "хворі на гепатит" (оскільки це, по суті, є стадії гепатиту). Додатково, для поліпшення якості подальшого аналізу проведено об'єднання трьох колонок в одну категорію – "гепатит".

Для покращення зручності аналізу в стовпці "Sex" значення були замінені на числові: "m" замінено на 0, а "f" на 1. Ці кроки спрямовані на зменшення кількості категорій та уніфікацію даних, що полегшить їхню обробку та інтерпретацію на наступних етапах аналізу. (рис. 3.6).

```
[532]: df['Category'] = df['Category'].replace({'0=Blood Donor': 0, '0s=suspect Blood Donor': 0, '1=Hepatitis': 1, '2=Fibrosis': 1, '3=Cirrhosis': 1})
       df['Sex'] = df['Sex'].replace({'m': 0, 'f': 1})

[533]: df
```

[533…

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
0	1	0	32	0	38.5	52.50000	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0
1	2	0	32	0	38.5	70.30000	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5
2	3	0	32	0	46.9	74.70000	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3
3	4	0	32	0	43.2	52.00000	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7
4	5	0	32	0	39.2	74.10000	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
557	608	1	52	1	39.0	37.00000	1.3	30.4	21.0	6.33	3.78	158.2	142.5	82.7
558	612	1	64	1	24.0	102.80000	2.9	44.4	20.0	1.54	3.02	63.0	35.9	71.3
559	613	1	64	1	29.0	87.30000	3.5	99.0	48.0	1.66	3.63	66.7	64.2	82.0
560	614	1	46	1	33.0	68.28392	39.0	62.0	20.0	3.56	4.20	52.0	50.0	71.0
561	615	1	59	1	36.0	68.28392	100.0	80.0	12.0	9.07	5.30	67.0	34.0	68.0

562 rows × 14 columns

Рисунок 3.6 – Заміна значень в стовпцях для полегшення подальшого аналізу.

Далі, проведено обчислення кількості викидів у датасеті. Цей етап аналізу дозволяє ідентифікувати та виділити незвичайні або виокремлені значення, які можуть впливати на статистичні результати та здатні внести спотворення у висновки. Виявлення викидів є важливим етапом у підготовці даних для точного та достовірного аналізу (рис. 3.7).

```
[535]: col=['Category', 'Age', 'Sex', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE',
               'CHOL', 'CREA', 'GGT', 'PROT']

[536]: q_low = df[col].quantile(0.01)
       q_hi  = df[col].quantile(0.99)

       df_outliers = df[(df[col] < q_low) | (df[col] > q_hi)]
       outlier_percentage = (df_outliers.sum() / len(df_outliers)) * 100
       print(outlier_percentage)

Unnamed: 0      0.000000
Category        0.000000
Age            97.864769
Sex             0.000000
ALB            80.444840
ALP           158.807829
ALT           106.387900
AST           139.003559
BIL            52.526690
CHE            16.094306
CHOL           11.804270
CREA          164.199288
GGT           186.156584
PROT          143.167260
dtype: float64
```

Рисунок 3.7 – Прорахована кількість викидів у датасеті.

Згідно з рисунком 3.7, виявлено значну кількість викидів у датасеті. З урахуванням цього факту, рекомендовано використовувати метод RobustScaler для маштабування та нормалізації даних. RobustScaler враховує вплив викидів, що робить його ефективним інструментом для обробки даних з аномальними значеннями. Цей метод допоможе забезпечити стабільні та достовірні результати аналізу, попереджаючи вплив великої кількості викидів на статистику (рис. 3.8).

```
from sklearn.preprocessing import RobustScaler

# Create a RobustScaler object
robust_scaler = RobustScaler()

# Define the columns to be scaled using RobustScaler
cols_to_scale = ['ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT']

# Scale the selected columns using RobustScaler
df[cols_to_scale] = robust_scaler.fit_transform(df[cols_to_scale])
```

+ Code + Markdown

```
38]:
q_low = df[col].quantile(0.01)
q_hi  = df[col].quantile(0.99)

df_outliers = df[(df[col] < q_low) | (df[col] > q_hi)]
outlier_percentage = (df_outliers.sum() / len(df_outliers)) * 100
print(outlier_percentage)
```

```
Unnamed: 0     0.000000
Category       0.000000
Age           97.864769
Sex            0.000000
ALB           -1.495526
ALP            0.690197
ALT            3.691243
AST            9.134348
BIL            6.863940
CHE           -0.657482
CHOL           0.277631
CREA          -0.010676
GGT            6.993578
PROT          -1.877531
dtype: float64
```

Рисунок 3.8 – Результат використання методу RobustScaler.

Як можна бачити на рисунку 3.8, після використання методу RobustScaler кількість викидів значно зменшилась, тому можна переходити до розвідувального аналізу.

3.3 Розвідувальний аналіз

Розвідувальний аналіз - це процес систематичного збору, обробки та інтерпретації інформації з метою забезпечення відомостей для прийняття обґрунтованих рішень в різних сферах діяльності. Він використовується в різноманітних галузях, таких як військова справа, бізнес, політика, наука та безпека[25].

Розвідувальний аналіз охоплює велике коло завдань, починаючи від визначення потреби у конкретній інформації і закінчуючи розробкою стратегій збору та обробки даних. Важливою частиною цього процесу є визначення джерел інформації, аналіз їхньої достовірності та значущості.

У розвідувальному аналізі використовуються різні методи та інструменти, такі як збір інформації від відкритих джерел, спостереження, інтерв'ю, аналіз супутникових знімків, криптоаналіз, аналіз соціальних мереж та багато інших. Комбінування різних підходів дозволяє створювати повні та об'єктивні картини ситуацій.

Важливим етапом є аналіз інформації, що була зібрана, з метою винесення висновків та прийняття рішень. Цей процес включає в себе оцінку надійності джерел, перевірку на можливі піддавання дезінформації та інші аспекти, які впливають на достовірність інформації.

Завдання розвідувального аналізу можуть бути дуже різноманітні: від прогнозування дій супротивника до визначення ринкових тенденцій. У будь-якому випадку, ефективний розвідувальний аналіз вимагає високого рівня експертності, креативності та вміння працювати з різноманітними джерелами інформації [28].

Загалом, розвідувальний аналіз є ключовим елементом для різних організацій та установ, допомагаючи їм адаптуватися до змін у навколишньому світі, прогнозувати події та приймати обґрунтовані стратегічні рішення. Однак, при всіх його перевагах, важливо дотримуватися етичних стандартів та забезпечувати захист конфіденційної інформації у процесі проведення розвідувальних дій [29].

У першому етапі розвідувального аналізу було побудовано кругові діаграми для візуалізації співвідношення чоловіків та жінок у датасеті, а також розподілу здорових пацієнтів в порівнянні з пацієнтами з гепатитом. Ці графіки дозволяють швидко отримати уявлення про статевий та хворобливий розподіл у наборі даних, що може бути корисним для подальшого уточнення аналізу та визначення основних тенденцій (рис. 3.9-3.10).

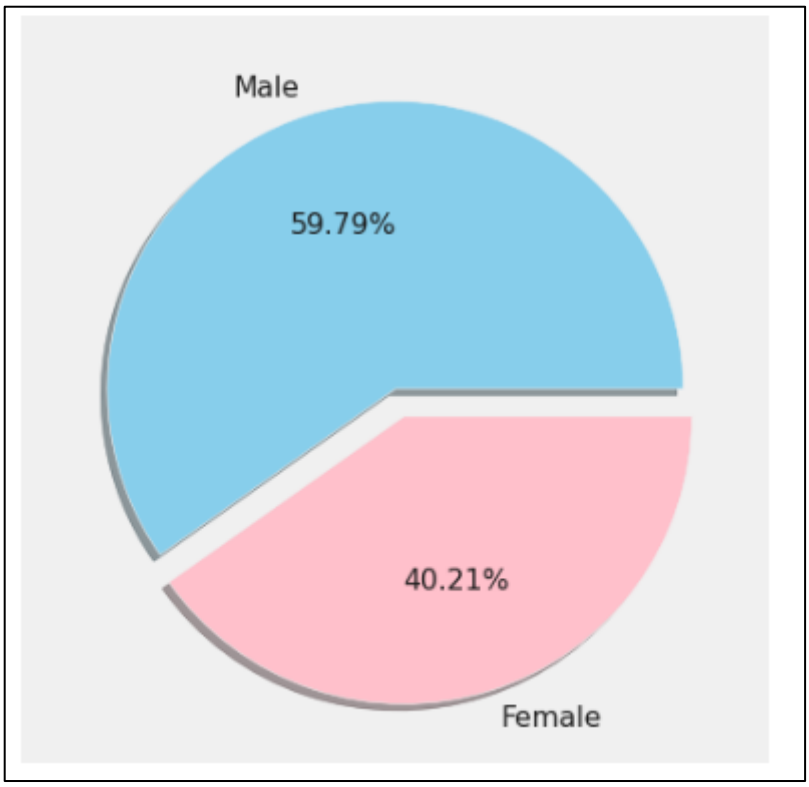


Рисунок 3.9 – Кругова діаграма розподілу за статтю

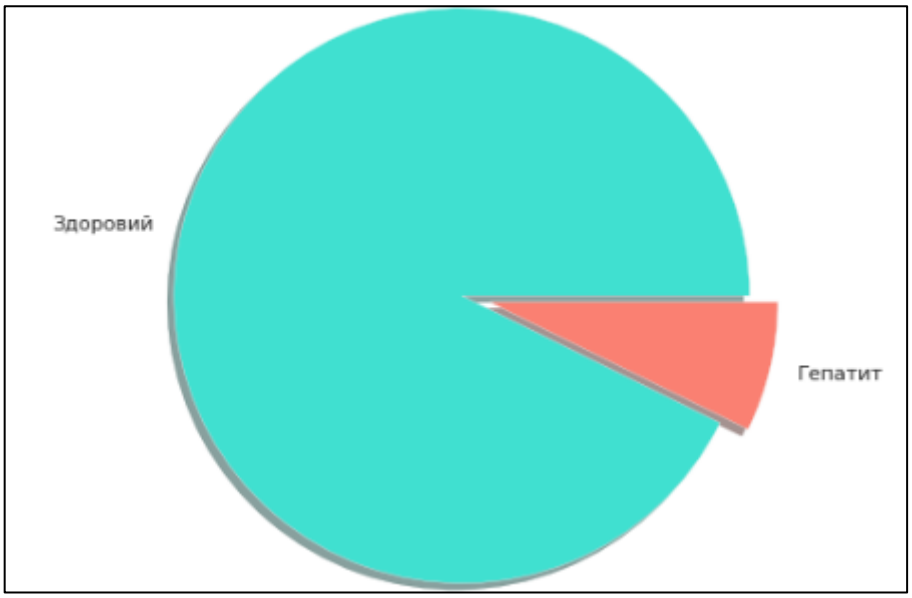


Рисунок 3.10 – Розподіл даних за здоровими та хворими.

Додатково, слід зазначити, що на рисунку 3.11 наведена теплова карта, яку слід розглядати для отримання додаткового уявлення про кореляції між змінними.

Теплова карта відображає ступінь кореляції між різними змінними. Кожна клітина цієї карти представляє собою числове значення, що вказує на величину кореляції між двома конкретними змінними. Значення кореляції може коливатися від -1 до 1, де -1 вказує на негативну кореляцію, 0 - на відсутність кореляції, а 1 - на позитивну кореляцію.

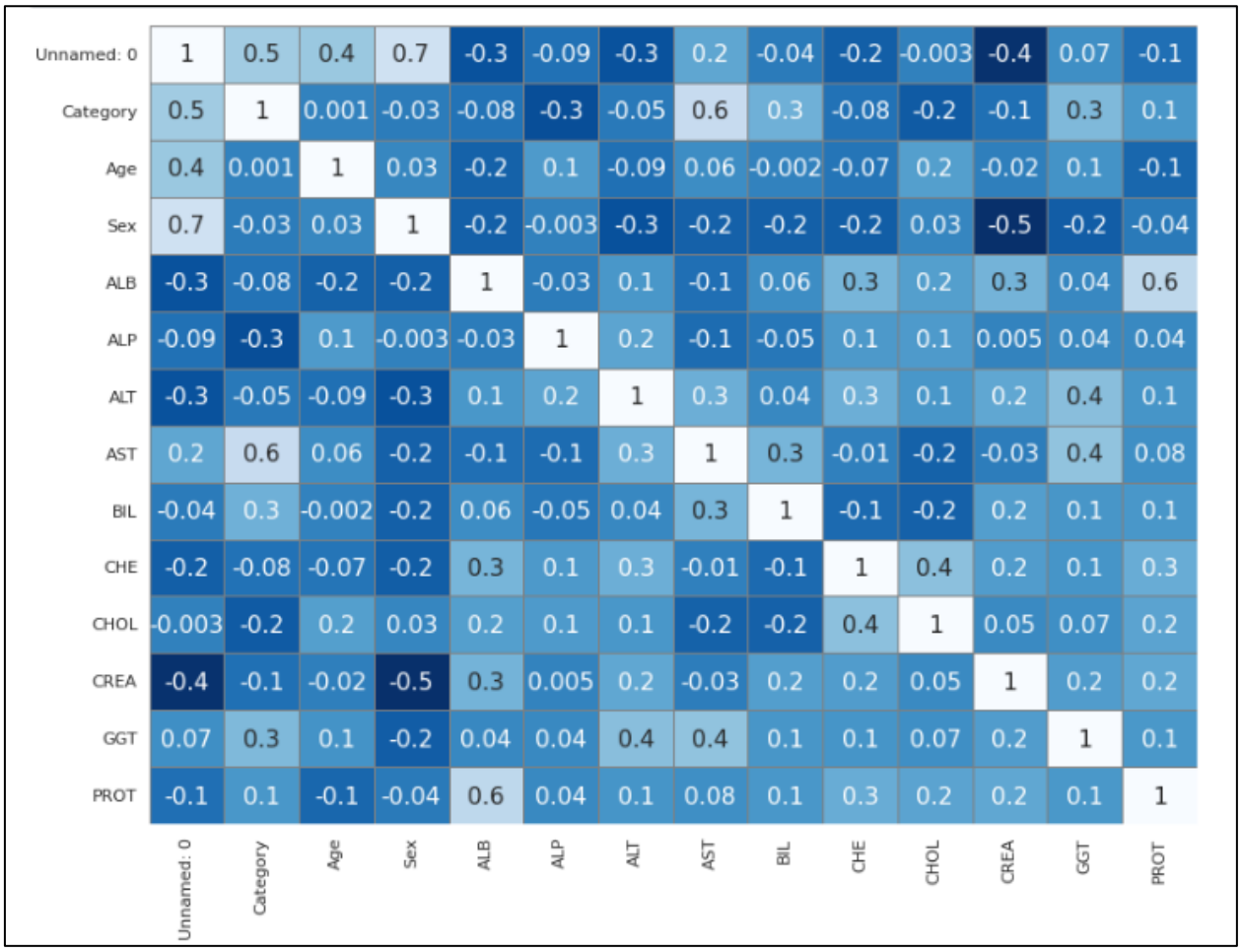


Рисунок 3.11 – Холодна карта показників.

Необхідно враховувати, що діагональні значення завжди дорівнюють 1, оскільки вони відображають кореляцію з тією ж самою змінною. Такий інструмент, як теплова карта, є важливим для аналізу даних, оскільки він дозволяє виявляти залежності між різними змінними та розуміти, як вони взаємодіють між собою.

Подивимося на конкретні результати, які можна отримати з цієї теплової карти:

– Є позитивна кореляція між AST (аспартатамінотрансфераза) та ALT (аланінамінотрансфераза).

– BIL (білірубін) позитивно корелює з ALP (фосфатаза лужна).

– Є негативна кореляція між ALB (альбумін) та BIL (білірубін).

Розглянемо також побудовану діаграму накопичення яка допомагає відобразити зв’язок між віком пацієнта та його діагнозом. Це графік, який показує кількість людей з гепатитом (категорія 1) та без гепатиту (категорія 0) у різних вікових групах (рис. 3.12).

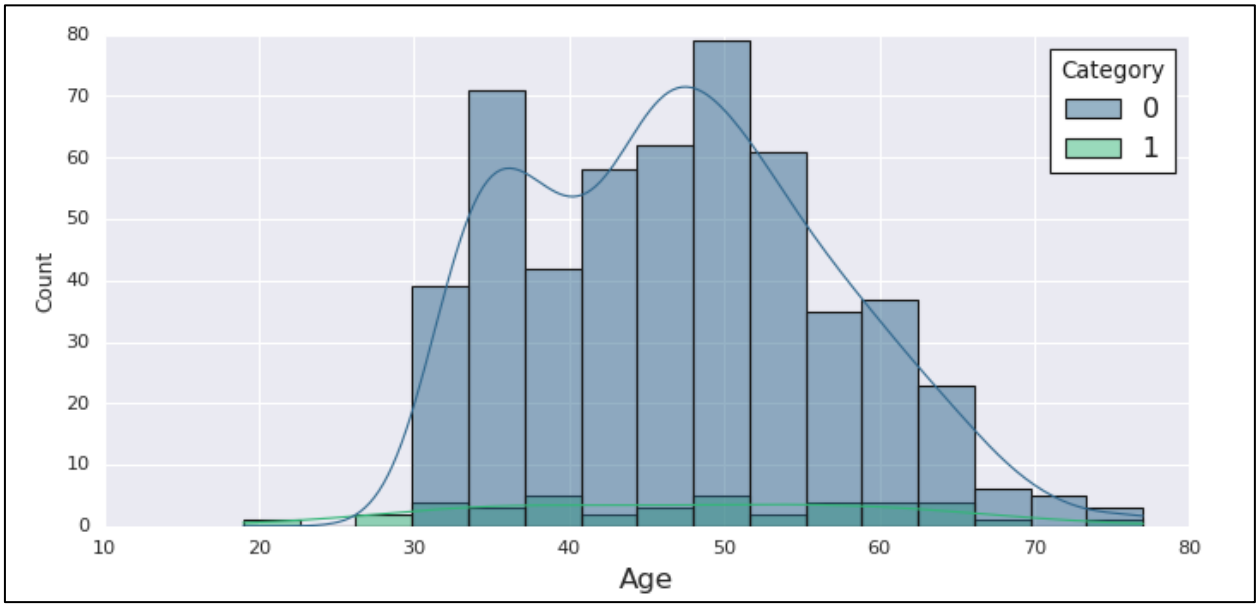


Рисунок 3.12 – Гістограма накопичення що відображає зв’язок між віком та діагнозом

З цього графіка можна зробити висновок, що кількість людей без гепатиту перевищує кількість людей з гепатитом у всіх вікових групах, і кількість людей з гепатитом зменшується зі зростанням віку.

Далі, розглянемо графік зв’язку рівня альбуміну та діагнозу (рис. 3.13).

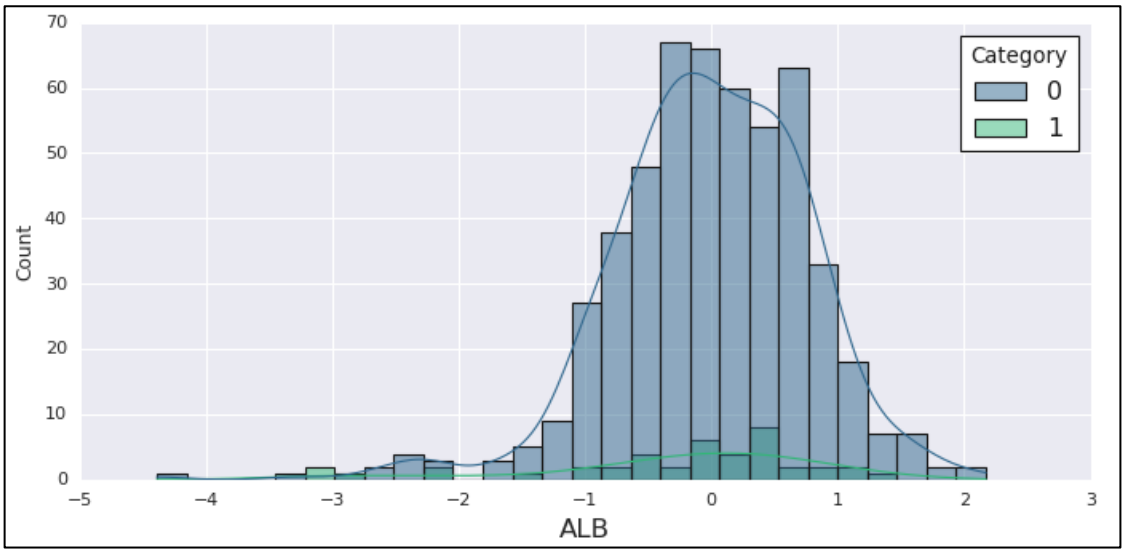


Рисунок 3.13 – Графік зв’язку рівня альбуміну та діагнозу.

За результатами графіка видно, що рівень ALB може бути важливим показником при визначенні наявності гепатиту. Наприклад, вищий рівень альбуміну може вказувати на відсутність гепатиту, тоді як нижчий рівень може бути пов’язаний з гепатитом.

На рисунку 3.14 показано графік зв’язку рівня альбуміну з діагнозом.

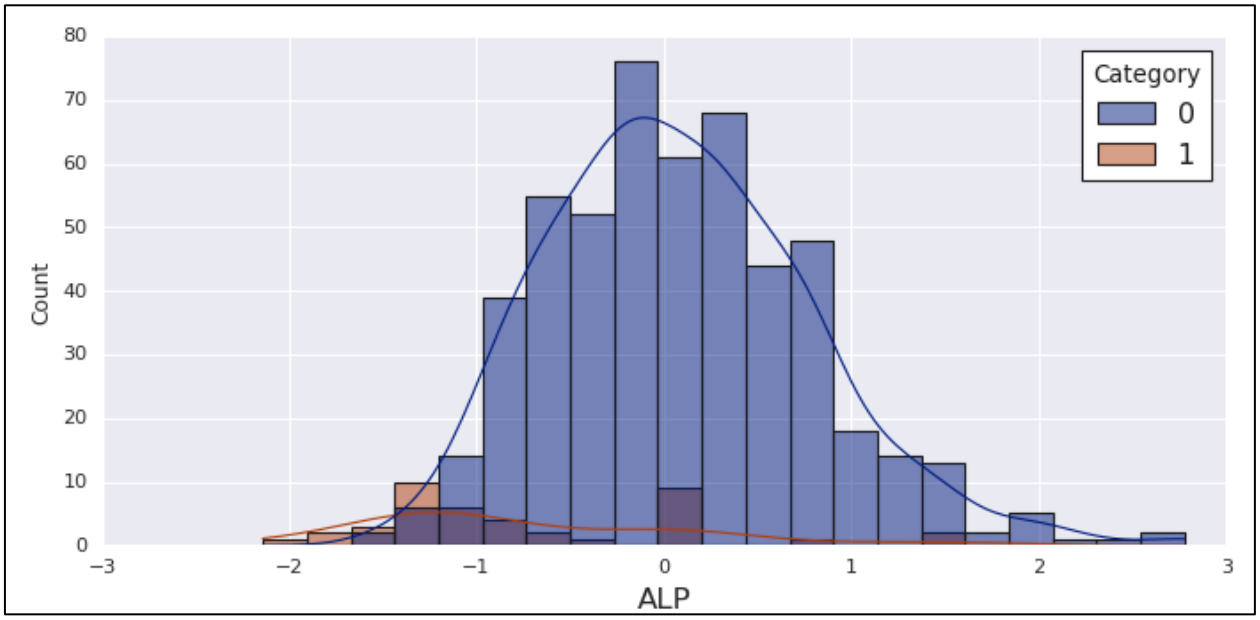


Рисунок 3.14 – Графік зв’язку рівня альбуміну з діагнозом.

Ця гістограма ілюструє зв'язок між рівнем ALP в крові та наявністю діагнозу гепатиту. У графіку сині стовпчики відображають осіб з встановленим діагнозом гепатиту, тоді як помаранчеві стовпчики представляють тих, у кого гепатиту немає.

Важливо відзначити, що показники значення 0 переважають за висотою над показниками 1 в області вищих рівнів ALP. Це свідчить про те, що при підвищених рівнях ALP існує вища ймовірність наявності гепатиту. Зробивши такий аналіз, можна зробити висновок, що підвищений рівень ALP може слугувати індикатором гепатиту.

Рисунок 3.15 відображає графік зв’язку рівня ALT та стану здоров’я пацієнта.

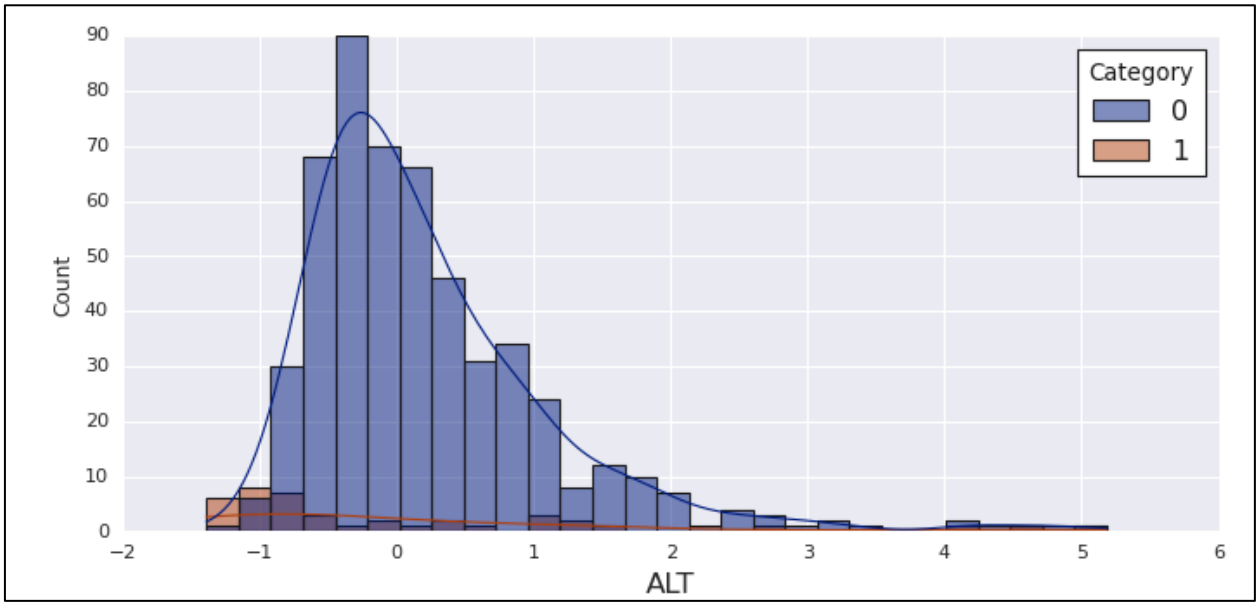


Рисунок 3.15 – Графік зв’язку рівня ALT та стану здоров’я пацієнта.

Аналіз графіку дозволяє зробити висновок, що серед осіб з вищими рівнями ALT переважає кількість осіб з нормальним станом здоров'я порівняно із тими, хто має знижене здоров'я. Це ставить перед нами завдання ретельніше вивчити цю групу осіб, оскільки високі рівні ALT можуть вказувати на підвищену ймовірність гепатиту та інших захворювань.

На рисунку 3.16 можна бачити зв’язок атрибуту AST та стану здоров’я людини.

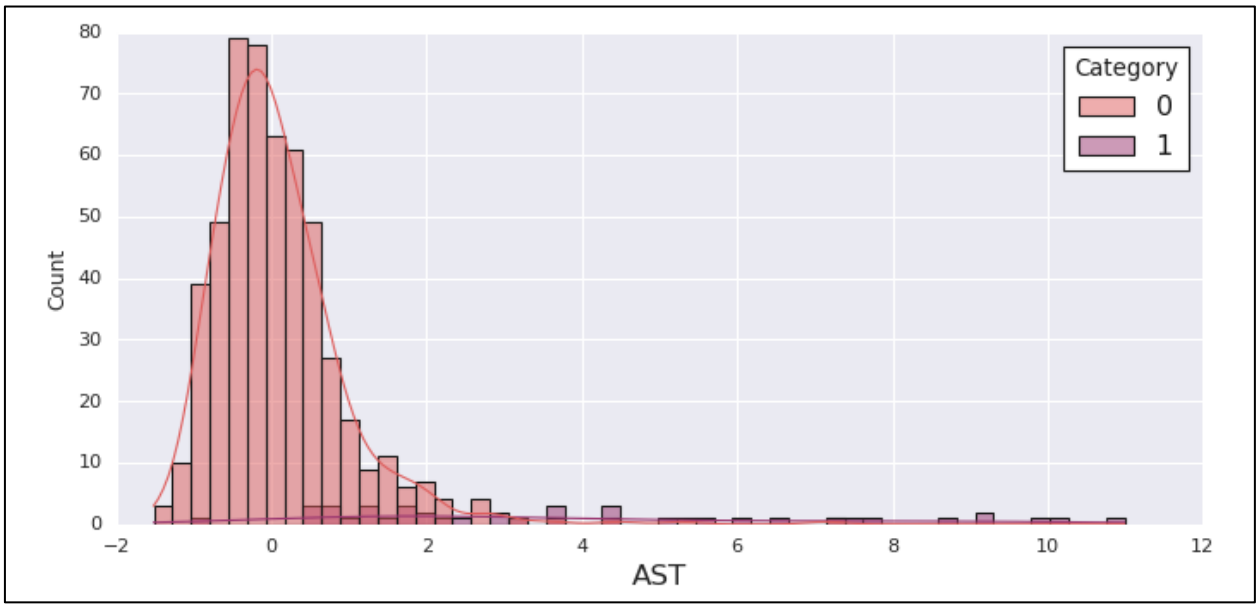


Рисунок 3.16 – Гістограма зв’язку атрибуту AST та стану здоров’я людини.

На зазначеній гістограмі можна відзначити тенденцію, згідно з якою підвищені рівні амінотрансферази (AST) зазвичай є індикатором прогресуючого ураження печінки, особливо у випадках гострого гепатиту. Збільшення концентрації AST може свідчити про збільшену активність цього ферменту в печінці, що може бути пов'язано з ураженням тканин органу.

Рисунок 3.17 є відображенням гістограми зв’язку рівня BIL та присутності у пацієнта гепатиту.

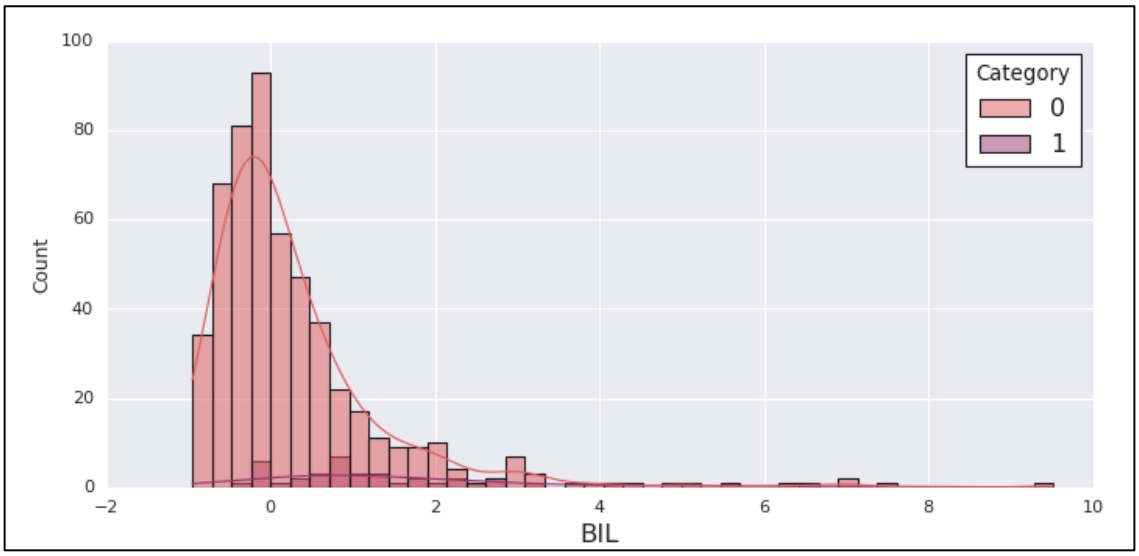


Рисунок 3.17 – Гістограма зв’язку атрибуту BIL та стану здоров’я людини.

Аналіз гістограми показує, що підвищені рівні білірубіну (BIL) можуть бути індикатором прогресуючого ураження печінки, особливо у випадках гострого гепатиту. Високі значення BIL часто вказують на можливість ураження цього органу, оскільки білірубін є продуктом розпаду червоних кров'яних клітин і його високі рівні часто пов'язані з недоліком функціонування печінки.

На рисунку 3.18 показано графік зв’язку атрибуту CHE (холінестераза) з станом здоров’я пацієнта.

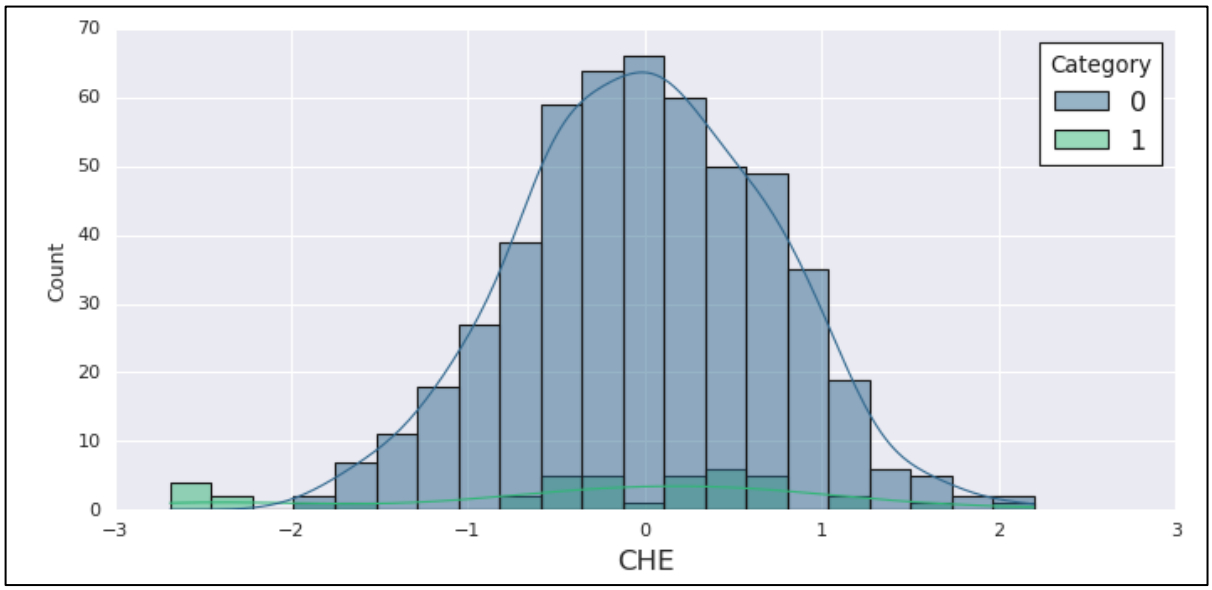


Рисунок 3.18 – Графік зв’язку атрибуту CHE та стану здоров’я пацієнта.

Графік ілюструє, що зниження рівня холінестерази в сироватці крові є суттєвим показником при виявленні гепатиту. Холінестераза, як фермент, є важливим компонентом, відповідальним за регулювання нервово-м'язової передачі. Зменшення активності цього ферменту може вказувати на пошкодження печінки, що є типовою характеристикою гепатиту.

Було також побудовано гістограму зв’язку рівня холестерину та діагнозу пацієнта (рис. 3.19).

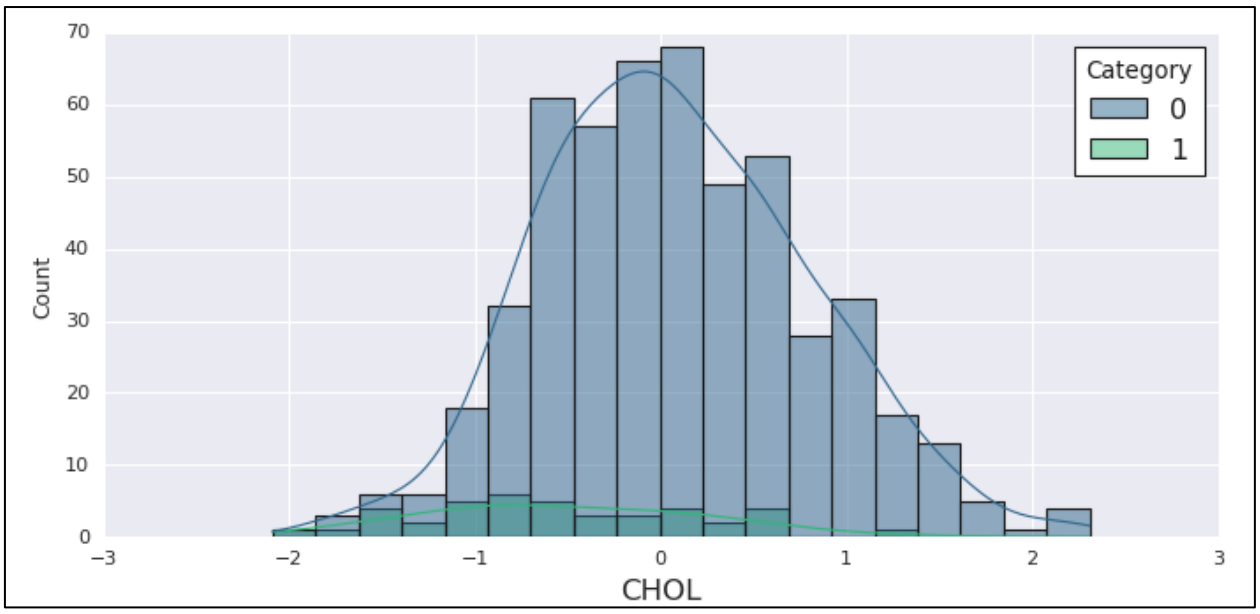


Рисунок 3.19 – Графік зв’язку рівня холестерину та діагнозу.

Аналізуючи низькі значення холестерину на графіку, можна припустити, що це може бути вказівкою на наявність гепатиту.

На рисунку 3.20 показано важливість показнику креатиніну на наявність гепатиту в пацієнта.

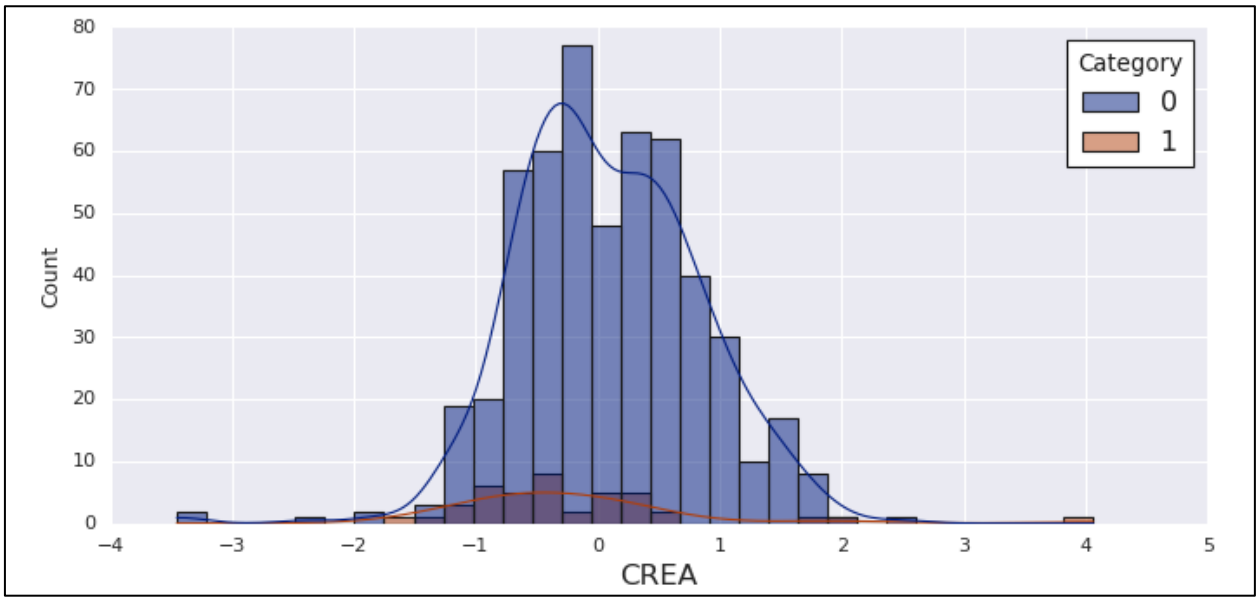


Рисунок 3.20 – Гістограма впливу креатиніну на діагноз.

Побудована діаграма свідчить про те, що найвищий відсоток хворих на гепатит спостерігається серед осіб, у яких рівень креатиніну є близьким до нуля або значно нижчим. Цей висновок може вказувати на можливий зв'язок між низьким рівнем креатиніну і розвитком гепатиту.

Креатинін, як індикатор функції нирок, може відображати ступінь ураження цих органів, а високий ризик гепатиту у пацієнтів із зниженим рівнем креатиніну вимагає уваги та подальших досліджень.

Графік на рисунку 3.21 відображає вплив гамма-глутамілтрансферази на діагноз.

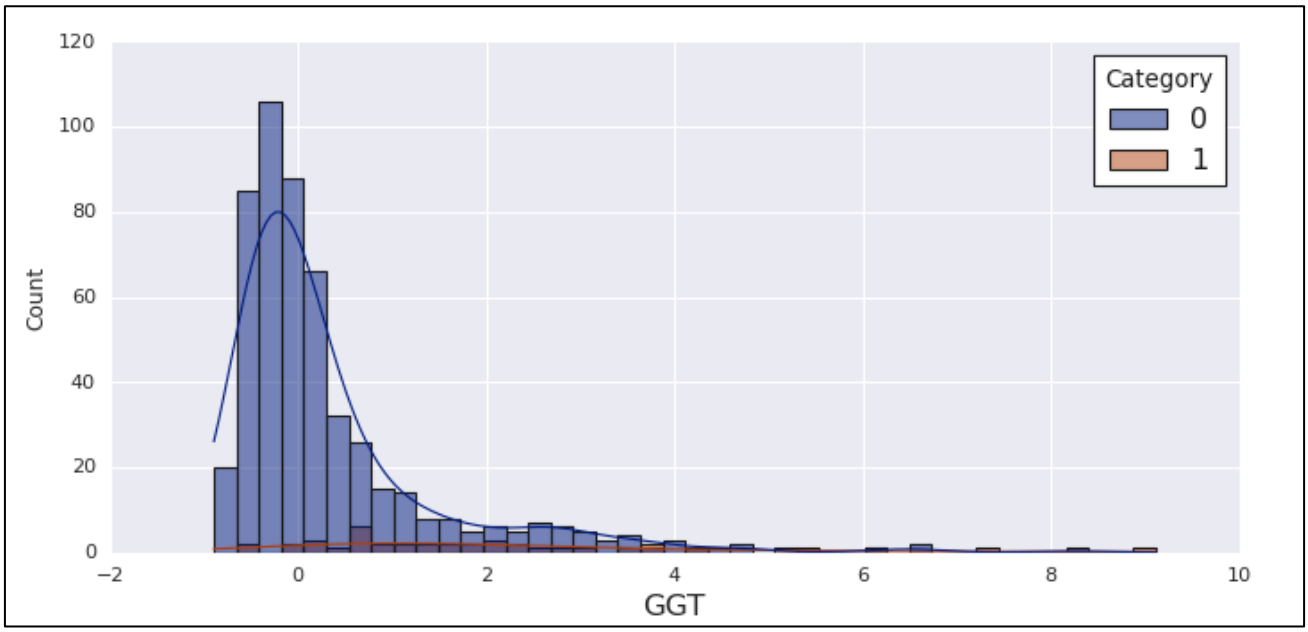


Рисунок 3.21 – Гістограма впливу атрибуту GGT на діагноз.

Аналіз гістограми свідчить, що рівень гамма-глутамілтрансферази (GGT) не виявляє значущого впливу на формування діагнозу. Хоча GGT часто використовується як маркер функції печінки і може підвищуватися при різних печінкових захворюваннях, отримані дані не демонструють вираженої кореляції між рівнем GGT і виникненням конкретного діагнозу.

3.4 Використання моделей машинного навчання для аналізу даних

Першим кроком перед побудовою моделей прогнозування було вирішено розділити датасет на тренувальний та тестовий та проведено нормалізацію даних (рис. 3.22).

```
# Завантаження даних та подальша обробка
df['Category'] = df['Category'].replace({'0=Blood Donor': 'Blood Donor',
                                         '0s=suspect Blood Donor': 'Blood Donor'})

# Вибір колонок для ознак та цільової змінної
columns = ["Age", "ALB", "ALP", "ALT", "AST", "BIL", "CHE", "CHOL", "CREA", "GGT", "Category"]
df_subset = df[columns]

# Розділення даних на тренувальний та тестовий набори
X = df_subset.drop("Category", axis=1)
y = df_subset["Category"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=42, stratify=y)

# Нормалізація даних
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Рисунок 3.22 – Код розділення датасету та нормалізації даних.

Код показаний на рисунку 3.22 визначає список ознак (колонок), які будуть використовуватися для навчання моделі, включаючи "Age", "ALB", "ALP", "ALT", "AST", "BIL", "CHE", "CHOL", "CREA", "GGT". Також включається цільова змінна "Category".

Дані розбиваються на тренувальний і тестовий набори за допомогою train_test_split. Розмір тестового набору встановлено на 70% від загальної кількості даних. Використовується стратифікація за змінною "Category" для забезпечення рівномірного розподілу категорій у тренувальному та тестовому наборах.

Використовується стандартний масштабувальник (StandardScaler), щоб нормалізувати дані. Ознаки тренувального та тестового наборів масштабуються, щоб мати середнє значення 0 та стандартне відхилення 1 [30].

Наступним кроком було побудовано моделі класифікації. На рисунку 3.23 показано код побудови моделі «LinearRegression».

```
# Логістична регресія
logistic_regression_model = LogisticRegression(random_state=42, max_iter=1000)
logistic_regression_model.fit(X_train_scaled, y_train)
y_pred_lr = logistic_regression_model.predict(X_test_scaled)

# Оцінка моделі
print("Логістична регресія:")
print("Accuracy:", accuracy_score(y_test, y_pred_lr))
print("Classification Report:\n", classification_report(y_test, y_pred_lr))

# Виведення кількості пацієнтів для кожної категорії окремо
category_counts = df['Category'].value_counts()

from sklearn.metrics import confusion_matrix

conf_matrix_lr = confusion_matrix(y_test, y_pred_lr)

# Plotting the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_lr, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Donor', 'Liver Disease']
plt.title('Confusion Matrix for Logistic Regression')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Рисунок 3.23 – Код побудови моделі «LogisticRegression».

У даному коді використовується логістична регресія для класифікації пацієнтів на категорії "Blood Donor" та "Liver Disease" на основі ряду біохімічних показників. Модель обучена на тренувальному наборі даних та оцінена на тестовому наборі.

Виведено основні метрики ефективності моделі, такі як точність (Accuracy) та звіт про класифікацію (Classification Report). Також, для кращого розуміння результатів, побудована теплова карта матриці плутанини (Confusion Matrix), яка візуалізує відповідність між прогнозованими та реальними класами.

Отримані метрики та графічне представлення допомагають зрозуміти ефективність логістичної регресії у класифікації пацієнтів за їхнім станом здоров'я (рис. 3.24).

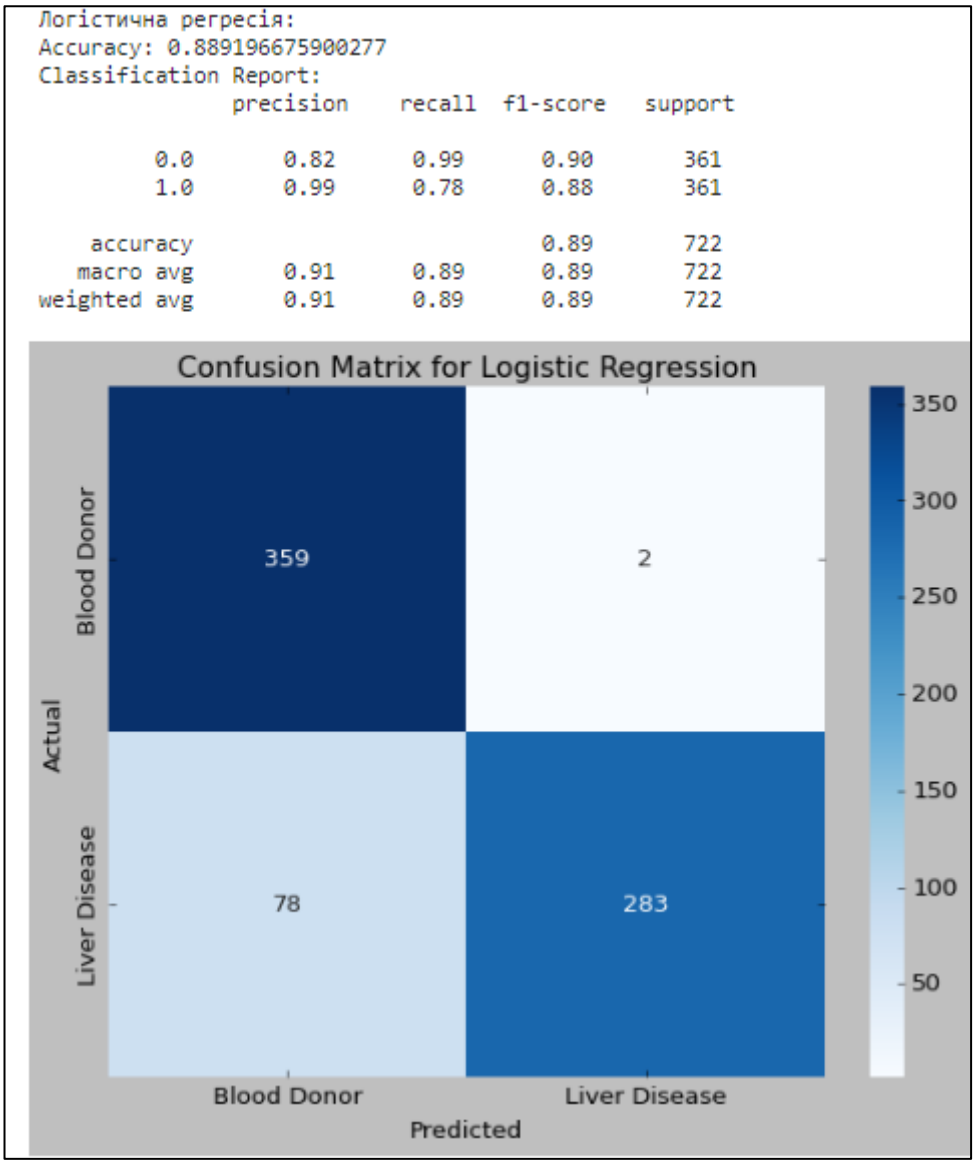


Рисунок 3.24 – Результат виконання моделі «Logistic Regression»

Як можна бачити на рисунку 3.24, модель «Logistic Regression» показала результат у 0.89 за шкалою r2_score, та виявила 283 людини з гепатитом і 359 здорових людини.

Далі було також побудовано модель «Random Forest Classifier», код побудови моделі показано на рисунку 3.25.

```
# RandomForestClassifier зі збільшеною кількістю дерев
random_forest_model = RandomForestClassifier(random_state=42, n_estimators=200, max_depth=None,
                                             min_samples_split=2, min_samples_leaf=1)
random_forest_model.fit(X_train, y_train)
y_pred_rf = random_forest_model.predict(X_test)

# Оцінка моделі
print("Random Forest Classifier:")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print("Classification Report:\n", classification_report(y_test, y_pred_rf))

# Виведення кількості пацієнтів для кожної категорії окремо
category_counts = df['Category'].value_counts()

conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)

# Plotting the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_rf, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Donor', 'Liver Disease'])
plt.title('Confusion Matrix for RFC')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Рисунок 3.25 – Код для побудови моделі «Random Forest Classifier»

У цьому фрагменті коду використовується класифікатор RandomForestClassifier зі збільшеною кількістю дерев у моделі. Модель навчена на тренувальному наборі даних, і її ефективність оцінена за допомогою ряду метрик.

Виведені результати включають точність та звіт про класифікацію (Classification Report), що надають інформацію про ефективність моделі у визначенні категорій "Blood Donor" та "Liver Disease". Знову виведена кількість пацієнтів для кожної категорії, щоб дати загальний огляд розподілу класів у тестовому наборі.

Результат виконання моделі «Random Forest Classifier» показано на рисунку 3.26.

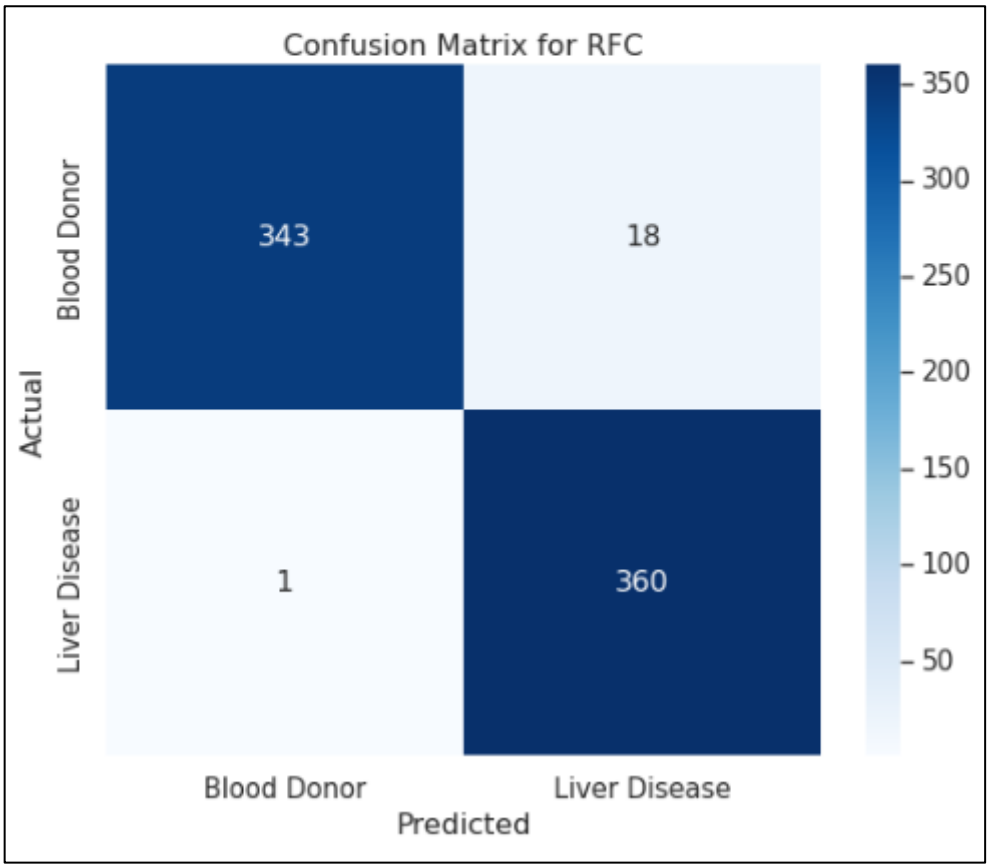


Рисунок 3.26 – Результат виконання моделі «Random Forest Classifier»

На рисунку 3.26, показано що модель «Random Forest Classifier» показала трошки гірший результат в 0,974 за шкалою r2_score, та виявила 332 людей з гепатитом і 361 здорових людини.

Наступним було використано модель «Gradient Boosting Classifier», код побудови моделі можна бачити на рисунку 3.27.

```
# Gradient Boosting Classifier
gradient_boosting_model = GradientBoostingClassifier(random_state=42, n_estimators=100, learning_rate=0.1, max_depth=3)
# Використовуємо крос-валідацію для оцінки точності
cv_scores = cross_val_score(gradient_boosting_model, X_train, y_train, cv=5)
print("Cross-Validation Accuracy:", cv_scores.mean())

# Навчання моделі на всій тренувальній вибірці
gradient_boosting_model.fit(X_train, y_train)
y_pred_gb = gradient_boosting_model.predict(X_test)

# Оцінка моделі
print("Gradient Boosting Classifier:")
print("Accuracy:", accuracy_score(y_test, y_pred_gb))
print("Classification Report:\n", classification_report(y_test, y_pred_gb))

# Виведення кількості пацієнтів для кожної категорії окремо
category_counts = df['Category'].value_counts()
```

Рисунок 3.27 – Код побудови моделі «Gradient Boosting Classifier».

У цьому фрагменті коду використовується класифікатор Gradient Boosting для класифікації пацієнтів на основі ряду біохімічних показників. Модель навчається на тренувальному наборі даних та оцінюється на тестовому наборі. Результати виводяться на екран у вигляді точності (Accuracy) та звіту про класифікацію (Classification Report).

Результат роботи моделі «Gradient Boosting» показано на рисунку 3.28.

```
Gradient Boosting Classifier:
Accuracy: 0.9487534626038782
Classification Report:
              precision    recall  f1-score   support

         0.0       0.91      1.00      0.95       361
         1.0       1.00      0.90      0.95       361

    accuracy                           0.95       722
   macro avg       0.95      0.95      0.95       722
weighted avg       0.95      0.95      0.95       722
```

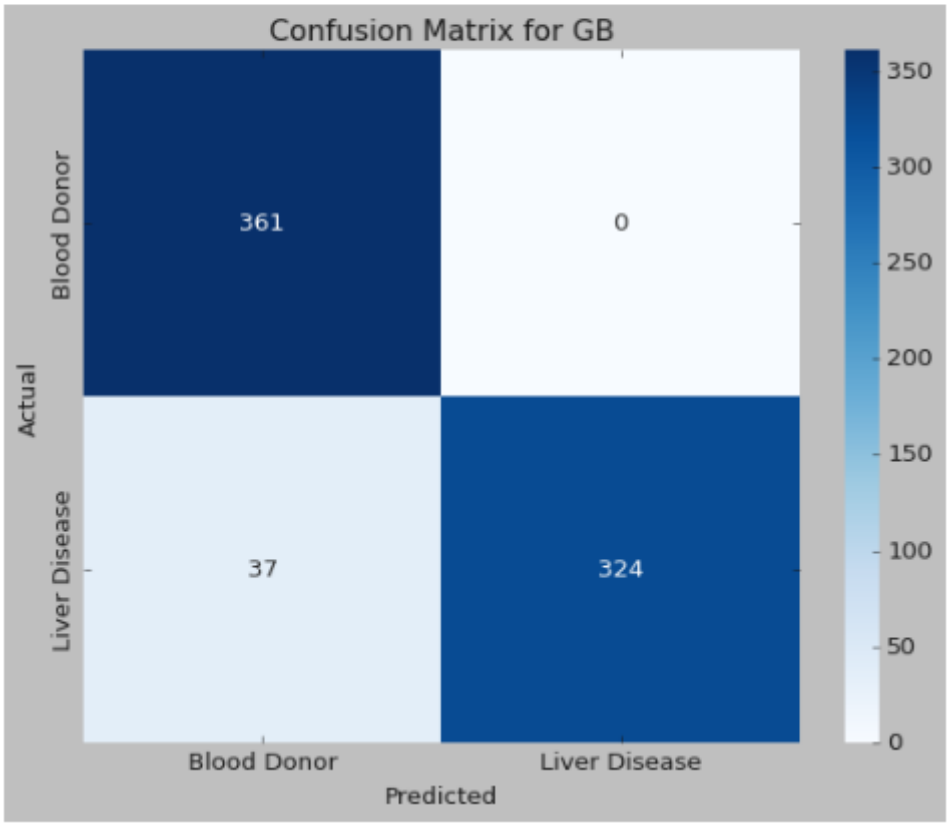


Рисунок 3.28 – Результат роботи моделі «Gradient Boosting»

Точність класифікатора становить 0.948 за шкалою r2_score що означає, що він правильно визначив 95 % донорів. Як видно з матриці плутанини, класифікатор правильно визначив 361 донора крові, які не мають захворювання печінки, і 324 донора крові, які мають захворювання печінки. Ці результати вказують на те, що класифікатор є ефективним для класифікації донорів крові за наявності захворювання печінки, але він не є ідеальним.

Наступним було використано модель «SVM», код побудови моделі можна бачити на рисунку 3.29.

```
# Support Vector Machine (SVM) Classifier
svm_model = SVC(random_state=42, kernel='linear')

# Train the model on the entire training set
svm_model.fit(X_train, y_train)
y_pred_svm = svm_model.predict(X_bal)

# Evaluate the model
# Note: R2 score is not suitable for classification problems; using accuracy for demonstration
print("Support Vector Machine Classifier:")
print("Accuracy:", accuracy_score(y_bal, y_pred_svm))
print("Classification Report:\n", classification_report(y_bal, y_pred_svm))

conf_matrix_svm = confusion_matrix(y_bal, y_pred_svm)

# Plotting the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_svm, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Do
nor', 'Liver Disease'])
plt.title('Confusion Matrix for SVM')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Рисунок 3.29 – Код побудови моделі «SVM».

В даному коді використовується метод опорних векторів (SVM) для класифікації даних. Модель навчається на тренувальному наборі і потім використовується для передбачення класів на збалансованому наборі.

Після цього проводиться оцінка точності моделі за допомогою метрик та виводяться результати вигляді точності та звіту про класифікацію. Також побудована теплова карта матриці плутанини для візуалізації результатів класифікації.

Результат роботи моделі «SVM» показано на рисунку 3.30.

```
Support Vector Machine Classifier:
Accuracy: 0.9238227146814404
Classification Report:
              precision    recall  f1-score   support

         0.0       0.88      0.99      0.93       361
         1.0       0.99      0.86      0.92       361

    accuracy                           0.92       722
   macro avg       0.93      0.92      0.92       722
weighted avg       0.93      0.92      0.92       722
```

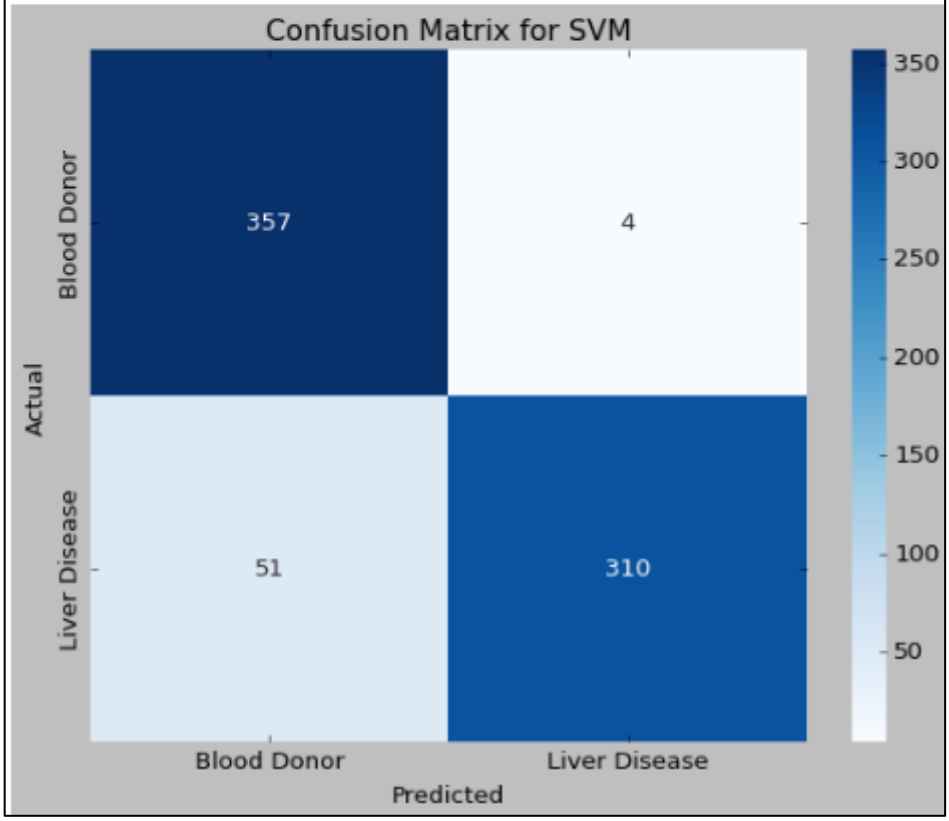


Рисунок 3.30 – Результат роботи моделі «SVM»

Абсолютна точність класифікації становить 0,92 за шкалою r2_score, що означає, що система правильно класифікувала 92% даних. Цей результат є значним, оскільки він значно вищий за випадкову точність, яка становить 50%.

Загалом, результати класифікації за допомогою SVM є дуже хорошими. Система може точно класифікувати донорів крові та захворювання печінки.

Цей результат є значним, оскільки він може бути використаний для підвищення якості медичного обслуговування.

На рисунку 3.31 показано код побудови моделі «DecisionTreeClassifier».

```
from sklearn.tree import DecisionTreeClassifier

# Decision Tree Classifier
decision_tree_model = DecisionTreeClassifier(random_state=42, max_depth=3)

# Train the model on the entire training set
decision_tree_model.fit(X_train, y_train)
y_pred_dt = decision_tree_model.predict(X_bal)

# Evaluate the model
print("Decision Tree Classifier:")
print("Accuracy:", accuracy_score(y_bal, y_pred_dt))
print("Classification Report:\n", classification_report(y_bal, y_pred_dt))

conf_matrix_dt = confusion_matrix(y_bal, y_pred_dt)

# Plotting the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_dt, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Don
or', 'Liver Disease'])
plt.title('Confusion Matrix for Decision Tree Classifier')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Рисунок 3.31 – Код побудови моделі «DecisionTreeClassifier»

У цьому фрагменті коду використовується класифікатор дерева рішень для розв'язання задачі класифікації. Модель навчається на тренувальному наборі даних з обмеженням глибини дерева. Після навчання модель використовується для передбачення класів на тестовому наборі.

Виводяться метрики оцінки моделі, такі як точність та звіт про класифікацію. Також будується теплова карта матриці плутанини для візуалізації результатів класифікації.

Результат роботи моделі «DecisionTreeClassifier» показано на рисунку 3.32.

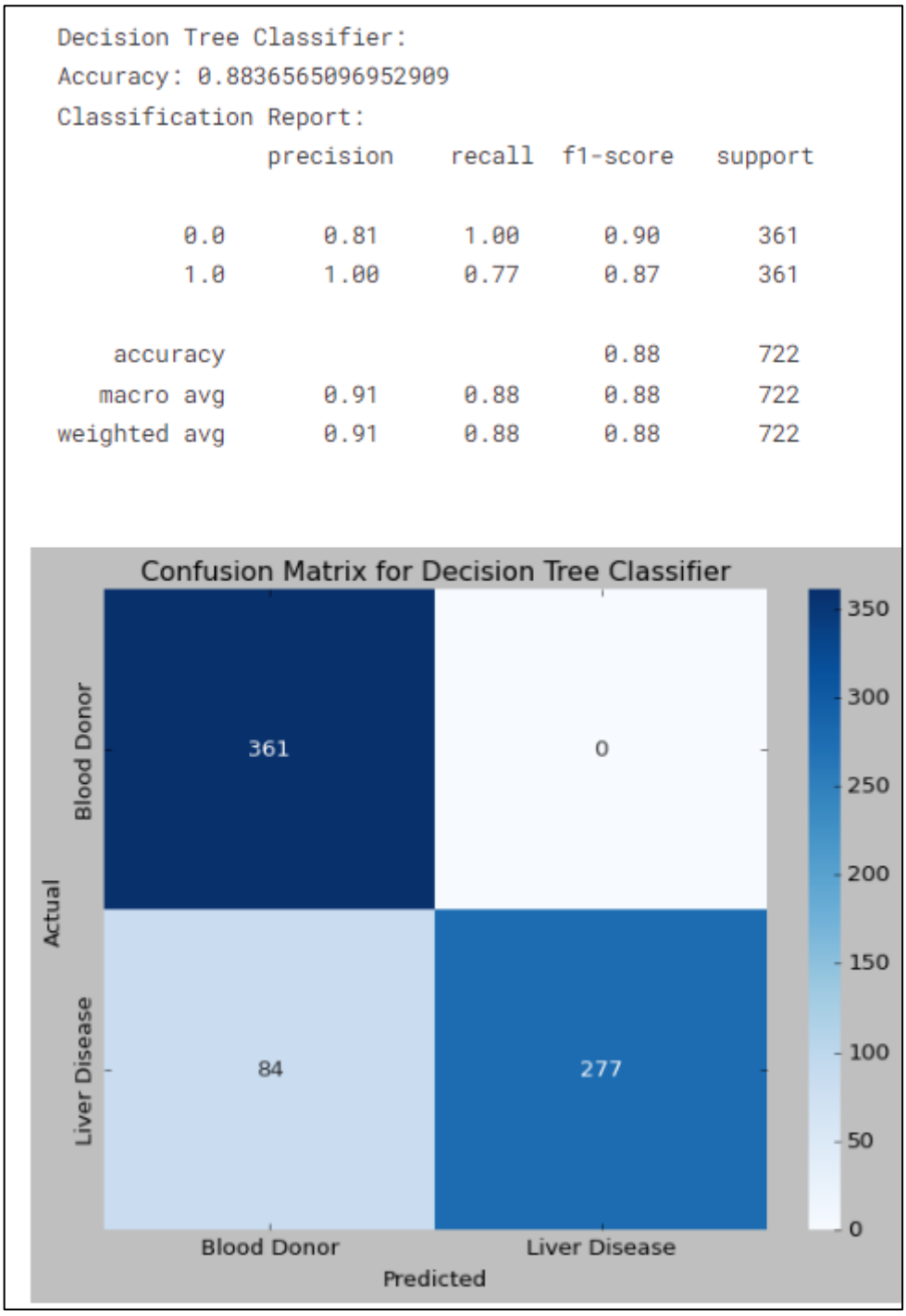


Рисунок 3.32 – Результат роботи моделі «DecisionTreeClassifier»

Точність класифікації становить 0,88, що означає, що дерево правильно класифікувало 88,36% зразків. Це досить високий показник, що свідчить про те, що дерево є ефективним класифікатором.

Точність класифікації для кожного класу також є високою. Для класу "донор крові" точність становить 0,81, а для класу "хвороба печінки" - 0,87. Це

означає, що дерево добре класифікує як зразки кровних донорів, так і зразки з хворобою печінки.

Матриця розбіжності показує, як дерево класифікувало кожен зразок. З матриці видно, що дерево правильно класифікувало 361 донора крові і 277 пацієнтів з хворобою печінки.

На рисунку 3.33 показано горизонтальну гістограму порівняння результатів класифікації.

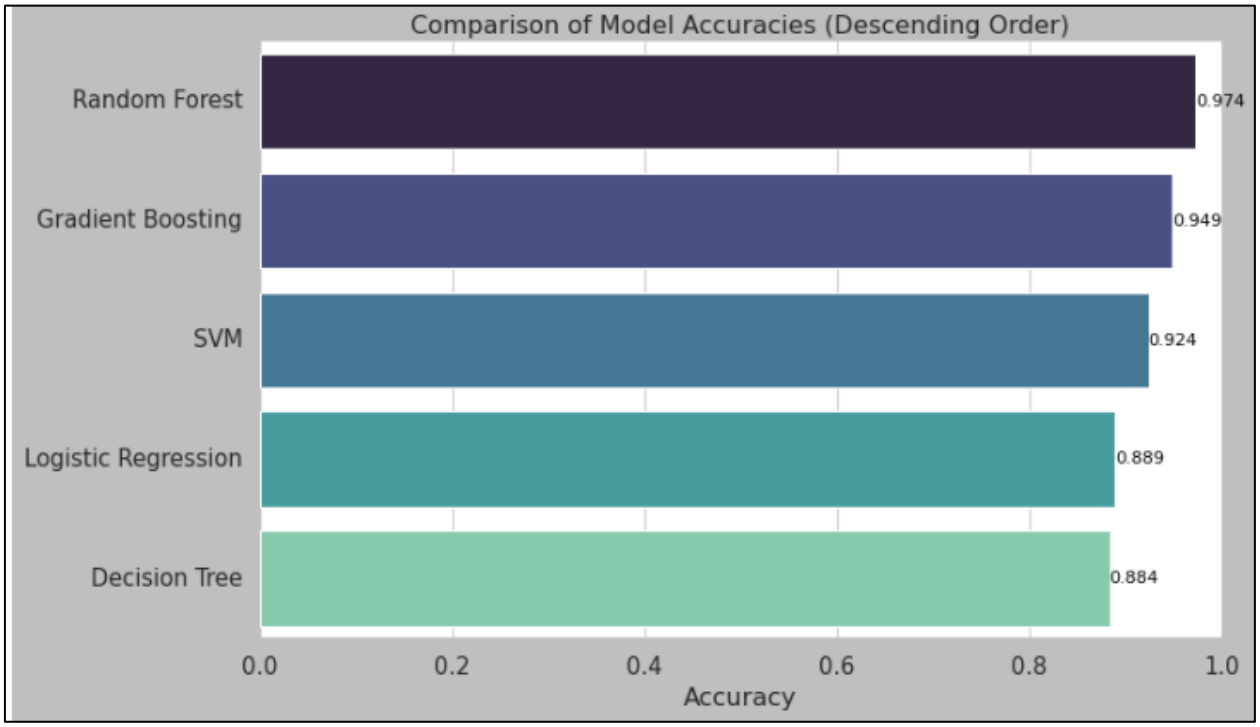


Рисунок 3.33 – Графік порівняння результатів класифікації.

На рисунку 3.34 показано таблицю порівняння результатів моделей.

	Модель	Точність на тренувальних даних	Точність на тестових даних
4	Decision Tree Classifier	0.989822	0.883657
0	Логістична регресія	0.974555	0.889197
3	Support Vector Machine Classifier	0.977099	0.923823
2	Gradient Boosting Classifier	0.918575	0.952941
1	Random Forest Classifier	0.954198	0.973684

Рисунок 3.34 – Таблиця порівняння результатів.

З результатів таблиці та горизонтальної гістограми видно, що найвищу точність має модель Random Forest Classificator (0,973). Вона генерує кілька простих моделей, які потім об'єднуються для отримання більш точного прогнозу.

На другому місці за точністю знаходиться модель градієнтного бустингу (0,949). Вона також генерує кілька простих моделей, але вони вдосконалюються з кожним новим циклом навчання.

На третьому місці знаходиться модель опорного вектора (0,924). Вона вибирає лінію або гіперплощину, яка найкраще розділяє дані на два класи. На четвертому місці знаходиться логістична регресія (0,889). Вона використовується для прогнозування дискретних величин, таких як "так" або "ні". На останньому місці знаходиться модель дерева рішень (0,884). Вона генерує дерево, яке представляє правила, які можна використовувати для прогнозування значень.

Загалом, можна сказати, що всі моделі, представлені на графіку, мають високу точність прогнозування. Найбільш точними є модель випадкового лісу та модель градієнтного бустингу.

3.5 Висновки

У проведеному дослідженні ефективність різних моделей класифікації була оцінена на основі їхньої точності прогнозування. За результатами аналізу можна визначити, що найвищу точність (0,973) за метрикою r2_score продемонструвала Random Forest Classifier. Ця модель використовує підхід, що базується на комбінації простих моделей для отримання більш точного прогнозу. Модель,

На другому місці за точністю знаходиться модель градієнтного бустингу, яка досягла значення точності 0,952 за метрикою r2_score. Градієнтний бустинг, також використовуючи ансамбль простих моделей,

вдосконалює їх з кожним новим циклом навчання, підвищуючи загальну точність.

На третьому місці розташувалась модель Support Vector Machine (SVM) із точністю 0,924. Ця модель вибирає оптимальну лінію або гіперплощину для розділення даних на два класи. Логістична регресія, з точністю 0,889 за метрикою r2_score, займає четверте місце.

4 ЕКОНОМІЧНА ЧАСТИНА

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

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

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- розраховано витрати на здійснення науково-технічної розробки;
- розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія аналізу та передбачення стану хворих на

гепатит» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

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

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

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

Продовження таблиці 4.1

6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій
12	Необхідна розробка регламентних документів та отримання великої	Необхідно отримання великої кількості дозвільних документів на	Процедура отримання дозвільних документів для виробництва та реалізації	Необхідно тільки повідомлення відповідним органам про виробництво	Відсутні будь-які регламентні обмеження на виробництво та реалізацію

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

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

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	4	3
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	3	4	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	2	2	2
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2

Продовження таблиці 4.2

10. Практична здійсненність (необхідність нових матеріалів)	2	2	2
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	33	37	34
Середньоарифметична сума балів $СБ_c$	34,7		

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

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

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

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та передбачення стану хворих на гепатит» становить 34,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

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

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

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

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

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

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

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

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

$$\beta_i = \frac{I_{ні}}{I_{аі}} , \tag{4.2}$$

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

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

$$\beta_i = \frac{I_{аі}}{I_{ні}} . \tag{4.3}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі

отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість використаних моделей машинного навчання	одиниць	3	4	1,33	0,2
Швидкість попередньої обробки та очистки даних	бал	4	8	2	0,25
Точність прогнозу	%	92	95	1,03	0,15
Кількість графіків розвідувального аналізу	одиниць	8	19	2,2	0,2
Кількість алгоритмів нормалізації даних	одиниць	1	3	3	0,2

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

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1{,}33 \cdot 0{,}2 + 2 \cdot 0{,}25 + 1{,}03 \cdot 0{,}15 + 2{,}2 \cdot 0{,}2 + 3 \cdot 0{,}2 = 1{,}96.$$

Отже, за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,96 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

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

4.3.1 Витрати на оплату праці

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

Основна заробітна плата дослідників

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

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

де k – кількість посад дослідників залучених до процесу досліджень;

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

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=22 дні.

$$З_о = 17000{,}00 \cdot 22 / 22 = 17000{,}00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник розробки	17000,00	772,73	22	17000,00
Інженер-аналітик (системний аналіз)	16500,00	750,00	22	16500,00
Консультант (лікар-терапевт вищої категорії)	15000,00	681,82	5	3409,09
Всього				36909,09

Основна заробітна плата робітників

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

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

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

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

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

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

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

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

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих

об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_р$ – середнє число робочих днів в місяці, приблизно $T_р$ = 22 дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 6700{,}00 \cdot 1{,}50 \cdot 1{,}35 / (22 \cdot 8) = 77{,}09 \text{ (грн)}.$$
$$З_{р1} = 77{,}09 \cdot 5{,}00 = 385{,}44 \text{ (грн)}.$$

На таблиці 4.6 показано величину витрат на основну заробітну плату робітників.

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Монтаж обчислювального обладнання та серверних блоків	5,00	4	1,50	77,09	385,44
Підготовка робочого місця дослідника-розробника інформаційної технології	7,00	2	1,10	56,53	395,72
Інсталяція програмного забезпечення розробки (моделювання) інформаційної технології аналізу	5,00	5	1,70	87,37	436,83
Всього					1217,99

Додаткова заробітна плата дослідників та робітників

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

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

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод} = (36909{,}09 + 1217{,}99) \cdot 11 / 100\% = 4193{,}98$ *грн.*

4.3.2 Відрахування на соціальні заходи

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

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

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

Зн = (36909,09 + 1217,99 + 4193,98) · 22 / 100% = 9310,63 (грн).

4.3.3 Сировина та матеріали

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

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

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

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

n – кількість видів матеріалів;

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

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

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

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

$М_1$ = 2,000 · 139,00 · 1,03 - 0,0 · 0,0 = 286,34 (грн).

Проведені розрахунки зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
USB-пам'ять	139,00	2,000	0,0	0,0	286,34
Диск оптичний	25,00	4,000	0,0	0,0	103,00
Картридж для принтера	950,00	1,000	0,0	0,0	978,50
Начиння канцелярське	195,00	3,000	0,0	0,0	602,55
Органайзер офісний	183,00	3,000	0,0	0,0	565,47
Папір для заміток (А5)	116,00	4,000	0,0	0,0	477,92
Папір канцелярський офісний (А4)	225,00	2,000	0,0	0,0	463,50
Всього					3477,28

4.3.4 Розрахунок витрат на комплектуючі

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

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

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

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

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

$К_в = 1 \cdot 3079,00 \cdot 1,03 = 3171,37$ *грн.*

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Зовнішній жорсткий диск 2.5" 2TB Seagate (STGD2000200)	1	3079,00	3171,37
Концентратор Defender SEPTIMA SLIM (83505)	1	400,00	412,00
Кабель для передачі даних USB to COM 1.0m Patron (CAB-PN-USB-COM)	1	354,00	364,62
Всього			3947,99

4.3.5 Спецустаткування для наукових (експериментальних) робіт

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

Балансову вартість спецустаткування розраховуємо за формулою:

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

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

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

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

k – кількість найменувань устаткування.

$$В_{спец} = 7818,00 \cdot 1 \cdot 1,03 = 8052,54 \text{ (грн)}.$$

Отримані результати зведемо до таблиці 4.9.

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Маршрутизатор MikroTik RB4011iGS+RM	1	7818,00	8052,54
Ноутбук ASUS RU451-UJ, оперативна пам'ять 4gb RAM, процесор intel core i5 2.5Ггц, Пам'ять на жорсткому диску 40mb	1	39460,00	40643,80
Серверне обладнання обробки та збереження DATA BASE на основі Artline Gaming X75 v17 (X75v17) Intel Core i7-10700F / RAM 16ГБ / HDD 2ТБ + SSD 480ГБ / nVidia GeForce RTX 3060 Ti 8ГБ	1	40299,00	41507,97
Всього			90204,31

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

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

Балансову вартість програмного забезпечення розраховуємо за формулою:

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

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

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

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

k – кількість найменувань програмних засобів.

$$В_{прг} = 7910,00 \cdot 1 \cdot 1,03 = 8147,30 \text{ (грн)}.$$

Отримані результати зведемо до таблиці 4.10.

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Прикладне програмне забезпечення розробки системи аналізу	1	7910,00	8147,30
Платформа Kaggle	1	4129,00	4252,87
Доступ до мережі Internet (високошвидкісний) грн/місяць	2	239,00	492,34
Hepatitis C Prediction Dataset	1	2350,00	2420,50
Всього			15313,01

4.3.7 Амортизація обладнання, програмних засобів та приміщень

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

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

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

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

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

$A_{обл} = (7300,00 \cdot 2) / (4 \cdot 12) = 304,17$ *грн.*

Проведені розрахунки зведемо до таблиці 4.11.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Блоки зовнішньої памяті серверного обладнання (зберігання бази даних)	7300,00	4	2	304,17
Дослідницька лабораторія	500000,00	20	2	4166,67
Місце оператора спеціалізоване	8200,00	5	2	273,33
Офісна оргтехніка	9600,00	4	2	400,00
Пристрої виведення інформації	6520,00	5	2	217,33
Програмне забезпечення Microsoft Windows, Offise 2021	9340,00	2	2	778,33
Програмно-обчислювальний комплекс розробки системи аналізу даних	35830,00	2	2	2985,83
Всього				9125,67

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

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

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

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

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

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

$$B_e = 0{,}03 \cdot 160{,}0 \cdot 7{,}50 \cdot 0{,}95 / 0{,}97 = 36{,}00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.12.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Маршрутизатор MikroTik RB4011iGS+RM	0,03	160,0	36,00
Ноутбук ASUS RU451-UJ, оперативна пам'ять 4gb RAM, процесор intel core i5 2.5Ггц, Пам'ять на жорсткому диску 40mb	0,06	160,0	72,00
Серверне обладнання обробки та збереження DATA BASE на основі Artline Gaming X75 v17 (X75v17) Intel Core i7-10700F / RAM 16ГБ / HDD 2ТБ + SSD 480ГБ / nVidia GeForce RTX 3060 Ti 8ГБ	0,42	160,0	504,00
Блоки зовнішньої памяті серверного обладнання (зберігання бази даних)	0,08	160,0	96,00
Місце оператора спеціалізоване	0,10	160,0	120,00
Офісна оргтехніка	0,50	1,5	5,63
Пристрої виведення інформації	0,40	2,3	6,90
Програмно-обчислювальний комплекс розробки системи аналізу даних	0,32	100,0	240,00
Всього			1080,53

4.3.9 Службові відрядження

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

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

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

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

$$В_{св} = (36909,09 + 1217,99) \cdot 20 / 100\% = 7625,42 \text{ (грн)}.$$

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

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

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

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

$$В_{сп} = (36909,09 + 1217,99) \cdot 30 / 100\% = 11438,12 \text{ (грн)}.$$

4.3.11 Інші витрати

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

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

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

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$$І_в = (36909,09 + 1217,99) \cdot 50 / 100\% = 19063,54 \text{ (грн)}.$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

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

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

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

$$В_{нзв} = (36909,09 + 1217,99) \cdot 100 / 100\% = 38127,08 \text{ (грн)}.$$

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

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

$B_{заг}$= 36909,09 +1217,99 +4193,98 +9310,63 +3477,28 +3947,99 +90204,31 + 15313,01 + 9125,67 +1080,53 +7625,42 +11438,12 +19063,54 +38127,08 = 251034,64 (грн).

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

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

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 251034,64 / 0,9 = 278927,38 (грн).

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

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

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

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

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

На таблиці 4.13 показано збільшення споживачів.

Таблиця 4.13 – Збільшення споживачів.

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	900	1800	1200	600

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 26300 осіб;

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 650,00 грн;

$\pm\Delta Ц_o$ – зміна вартості послуги від впровадження результатів, приймемо 224,33 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [32]:

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

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

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 38\%$;

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

Збільшення чистого прибутку 1-го року:

$$\Delta П_1 = (224{,}33 \cdot 26300{,}00 + 874{,}33 \cdot 900) \cdot 0{,}83 \cdot 0{,}38 \cdot (1 - 0{,}18/100\%) = 1729352{,}33 \text{ (грн)}.$$

Збільшення чистого прибутку 2-го року:

$$\Delta П_2 = (224{,}33 \cdot 26300{,}00 + 874{,}33 \cdot 2700) \cdot 0{,}83 \cdot 0{,}38 \cdot (1 - 0{,}18/100\%) = 2136377{,}20 \text{ (грн)}.$$

Збільшення чистого прибутку 3-го року:

$$\Delta П_3 = (224{,}33 \cdot 26300{,}00 + 874{,}33 \cdot 3900) \cdot 0{,}83 \cdot 0{,}38 \cdot (1 - 0{,}18/100\%) = 2407727{,}11 \text{ (грн)}.$$

Збільшення чистого прибутку 4-го року:

$$\Delta П_4 = (224{,}33 \cdot 26300{,}00 + 874{,}33 \cdot 4500) \cdot 0{,}83 \cdot 0{,}38 \cdot (1 - 0{,}18/100\%) = 2543402{,}06 \text{ (грн)}.$$

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

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

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

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

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,2;

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

$$ПП = 1729352{,}33/(1+0{,}2)^1 + 2136377{,}20/(1+0{,}2)^2 + 2407727{,}11/(1+0{,}2)^3 + 2543402{,}06/(1+0{,}2)^4 = 1441126{,}94 + 1483595{,}28 + 1393360{,}60 + 1226563{,}50 = 5544646{,}31 \text{ (грн)}.$$

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

$$PV = k_{інв} \cdot ЗВ, \tag{4.23}$$

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

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 278927,38 грн.

$$PV = k_{інв} \cdot ЗВ = 2{,}1 \cdot 278927{,}38 = 585747{,}50 \text{ (грн)}.$$

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

$$E_{абс} = ПП - PV \tag{4.24}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 5544646,31 грн;

PV – теперішня вартість початкових інвестицій, 585747,50 грн.

$$E_{абс} = ПП - PV = 5544646{,}31 - 585747{,}50 = 4958898{,}81 \text{ (грн)}.$$

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

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

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 4958898,81 грн;

PV – теперішня вартість початкових інвестицій, 585747,50 грн;

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

$$E_{в} = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1 = (1+4958898,81/585747,50)^{1/4} = 0,75.$$

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

$$\tau_{мін} = d + f, \tag{4.26}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,09;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,35.

$$\tau_{мін} = 0,09+0,35 = 0,44.$$

$\tau_{мін} < 0,75$ свідчить про те, що внутрішня економічна дохідність інвестицій $E_{в}$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія аналізу та передбачення стану хворих на гепатит» доцільно.

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

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

де $E_{в}$ – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,75 = 1,33 \text{ (року)}.$$

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

4.5 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та передбачення стану хворих на гепатит» становить 34,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,96 рази.

Також термін окупності становить 1,33 року, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія аналізу та передбачення стану хворих на гепатит».

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена інформаційній технології аналізу та прогнозування стану хворих на гепатит.

Проведене дослідження виявило значущі аспекти в сучасних інформаційних технологіях, зокрема в області машинного навчання. Вибір оптимальної моделі для розв'язання конкретної задачі виконувався з урахуванням специфіки завдання, обсягу та характеру наявних даних.

В результаті глибокого аналізу предметної області було визначено вже існуючі аналоги та різноманітні технології. Акцент робився на обґрунтованому виборі технологій, найбільш відповідних для успішного вирішення задачі. У цьому контексті були враховані актуальні тенденції та інноваційні рішення.

Проведено детальний огляд моделей машинного навчання, що використовувались для передбачення стану хворих на гепатит. Створено новий ноутбук та підключено датасет з медичними показниками людей, в яких спостерігались симптоми захворювань печінки. Проведено ретельний аналіз даних. Здійснено необхідні маніпуляції з даними аби привести їх до необхідного формату для подальшої роботи. Здійснено розвідувальний аналіз даних, побудовано графіки для візуального відображення значення показників. Підготовлено тренувальний та тестові датасети для подальшого передбачення.

У результаті побудови моделей машинного навчання виявлено, що Random Forest Classifier продемонстрував найвищу точність прогнозування (0,973). Його ефективність ґрунтується на комбінації простих моделей, що дозволяє отримати точні прогнози.

Отже, можна зробити висновок про успішне створення інформаційної технологія аналізу та прогнозування стану хворих на гепатит.

Результати роботи опубліковано в матеріалах LIII Всеукраїнської науково-технічної підрозділів Вінницького національного технічного університету (2023-2024) [1].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гуцу І.П., Жуков С.О. Інформаційна технологія аналізу та передбачення стану хворих на гепатит. *LIII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024). Вінниця, 2024.* URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/19792/16384 (дата звернення 19.12.2023)

2. Hepatitis C - WHO | World Health Organization. URL: https://www.who.int/news-room/fact-sheets/detail/hepatitis-c. (date of access: 01.10.2023).

3. Hepatitis C - Symptoms and causes - Mayo Clinic URL: https://www.mayoclinic.org/diseases-conditions/hepatitis-c/symptoms-causes/syc-20354278 (date of access: 03.10.2023).

4. Hepatitis C - FAQs, Statistics, Data, & Guidelines | CDC URL: https://www.cdc.gov/hepatitis/hcv/index.html (date of access: 05.11.2023).

5. Гепатит С, гострий - MSD Manual Professional Edition [Електронний ресурс]: URL: https://www.msdmanuals.com/uk/professional/hepatic-and-biliary-disorders/hepatitis/hepatitis-c-acute (date of access: 09.11.2023).

6. Мобільний додаток «Liver Health». URL: https://play.google.com/store/apps/datasafety?id=com.worthyworks.liverhealthapppremium&hl=uk&gl=US&pli=1 (date of access: 12.11.2023).

7. ConnQuER HepC App: Educational and Care Companion App. URL: https://aidsetc.org/resource/connquer-hepc-app-educational-and-care-companion-app. (date of access: 13.11.2023).

8. Мобільний додаток «My GI Health». URL: https://mygi.health/app. (date of access: 14.11.2023).

9. Сервіс «HepVu». URL: https://hepvu.org/services/#/ . (date of access: 13.11.2023).

10. «Ensemble-of-3-Models-Accuracy=95%» Kaggle Notebook. URL: https://www.kaggle.com/code/abhijeetsinghmeena/ensemble-of-3-models-accuracy-95 (date of access: 16.11.2023).

11. «EDA and hepatitis C prediction using RFs» Kaggle Notebook. URL: https://www.kaggle.com/code/yadhua/eda-and-hepatitis-c-prediction-using-rfs (date of access: 16.11.2023).

12. «Hepatitis C Prediction» Kaggle Notebook. URL: https://www.kaggle.com/code/mahmoudelhusseni/hepatitis-c-prediction (date of access: 17.11.2023).

13. «Hepatitis C Visualize Importance» Kaggle Notebook. URL: https://www.kaggle.com/code/stpeteishii/hepatitis-c-visualize-importance (date of access: 17.11.2023).

14. Kaggle. URL: https://www.kaggle.com/. (date of access: 17.11.2023).

15. «Що таке кагл? - визначення з техопедії - Розвиток 2023». URL: https://uk.theastrologypage.com/kaggle (date of access: 18.11.2023).

16. Офіційний сайт мови програмування Python. URL: https://www.python.org/ (date of access: 18.11.2023).

17. Python Introduction URL: https://www.w3schools.com/python/python_intro.asp (date of access: 22.11.2023).

18. Top 30 Python Libraries To Know in 2024 - Great Learning. URL: https://www.mygreatlearning.com/blog/open-source-python-libraries/ (date of access: 22.11.2023).

19. What is a Decision Tree | IBM. URL: https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes (date of access: 22.11.2023).

20. Support Vector Machine (SVM) Algorithm. URL: https://www.geeksforgeeks.org/support-vector-machine-algorithm/ (date of access: 23.11.2023).

21. A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. URL: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/ (date of access: 23.11.2023).

22. Understand Random Forest Algorithms With Examples (Updated 2023). URL: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ (date of access: 25.11.2023).

23. Logistic Regression - an overview | ScienceDirect Topics. URL: https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on (date of access: 26.11.2023).

24. One-vs-Rest and One-vs-One for Multi-Class Classification. URL: https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/ (date of access: 28.11.2023).

25. Pandas - Python Data Analysis Library. URL: https://pandas.pydata.org/ (date of access: 28.11.2023).

26. Matplotlib — Visualization with Python. URL: https://matplotlib.org/ (date of access: 30.11.2023).

27. Scikit-learn: machine learning in Python. URL: https://scikit-learn.org/stable/ (date of access: 02.12.2023).

28. SciPy documentation — SciPy v1.13.0.dev Manual. URL: https://scipy.github.io/devdocs/ (date of access: 02.12.2023).

29. Warning control — Python 3.12.1 documentation. URL: https://docs.python.org/3/library/warnings.html (date of access: 05.12.2023).

30. Joblib: running Python functions as pipeline jobs. URL: https://joblib.readthedocs.io/en/stable/ (date of access: 04.12.2023).

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

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

Додаток А

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

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

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

Керівник: к.т.н., доц.
_____ Сергій ЖУКОВ
«___» _____ 2023 р.

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

Вінниця 2023

1. Підстава для проведення робіт

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

2. Джерела розробки:

1. One-vs-Rest and One-vs-One for Multi-Class Classification. URL: https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/ (date of access: 28.11.2023).
2. Deep Learning | Introduction to Long Short Term Memory – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (date of access: 01.12.2023).

3. Мета і призначення роботи: Метою даної магістерської кваліфікаційної роботи є підвищення точності передбачення стану хворих на гепатит.

4. Вихідні дані для проведення робіт:

Датасет Kaggle «Hepatitis C Prediction Dataset» з даними для передбачення ймовірності гепатиту у пацієнтів.

5.Методи дослідження:

- Підготовка даних;
- Розвідувальний аналіз;
- Моделі машинного навчання для передбачення даних;

6. Етапи роботи і терміни їх виконання:

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

7. Очікувані результати та порядок реалізації:

Очікуваним результатом є розроблення інформаційної технології аналізу та передбачення стану хворих на гепатит.

8. Вимоги до розробленої документації

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

9. Порядок приймання роботи

Публічний захист «____» _____ 2023 р.

Початок розробки «____» _____ 2023 р.

Граничні терміни виконання МКР «____» _____ 2023 р.

Розробив студент групи 2ІСТ-22м _____ Ігор ГУЦУ

Додаток Б
Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія аналізу та передбачення стану хворих на гепатит»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 94.8% Схожість 5.01 %

Аналіз звіту подібності (відмітити потрібне)

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

Особа, відповідальна за перевірку (підпис) Сергій ЖУКОВ

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

Автор роботи (підпис) Ігор ГУЦУ

Керівник роботи Сергій ЖУКОВ

Додаток В
Лістинг програми

```
# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.281663Z","iopub.execute_input":"2023-12-17T16:43:46.282977Z","iopub.status.idle":"2023-12-17T16:43:46.292107Z","shell.execute_reply.started":"2023-12-17T16:43:46.282939Z","shell.execute_reply":"2023-12-17T16:43:46.291034Z"}}

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import random

from contextlib import contextmanager

from time import time

from tqdm import tqdm

import lightgbm as lgbm

import category_encoders as ce

from tensorflow.keras.utils import to_categorical

from sklearn.metrics import classification_report, log_loss, accuracy_score

from sklearn.metrics import mean_squared_error

from sklearn.model_selection import KFold

from matplotlib import style

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OrdinalEncoder

import pandas as pd

from sklearn.model_selection import train_test_split, GridSearchCV

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier

from sklearn.svm import SVC

from sklearn.neural_network import MLPClassifier

from sklearn.metrics import accuracy_score, confusion_matrix

import warnings

import joblib

styluse("fivethirtyeight")

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.294247Z","iopub.execute_input":"2023-12-17T16:43:46.294715Z","iopub.status.idle":"2023-12-17T16:43:46.314318Z","shell.execute_reply.started":"2023-12-17T16:43:46.294673Z","shell.execute_reply":"2023-12-17T16:43:46.313179Z"}}

df=pd.read_csv('/kaggle/input/hepatitis-c-dataset/HepatitisCdata.csv')
```

%% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.315905Z","iopub.execute_input":"2023-12-17T16:43:46.316596Z","iopub.status.idle":"2023-12-17T16:43:46.349042Z","shell.execute_reply.started":"2023-12-17T16:43:46.316556Z","shell.execute_reply":"2023-12-17T16:43:46.348025Z"}}

```
df
```

%% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.352368Z","iopub.execute_input":"2023-12-17T16:43:46.352808Z","iopub.status.idle":"2023-12-17T16:43:46.360370Z","shell.execute_reply.started":"2023-12-17T16:43:46.352771Z","shell.execute_reply":"2023-12-17T16:43:46.359278Z"}}

```
print(df.isnull().sum())
```

%% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.361983Z","iopub.execute_input":"2023-12-17T16:43:46.362509Z","iopub.status.idle":"2023-12-17T16:43:46.375137Z","shell.execute_reply.started":"2023-12-17T16:43:46.362471Z","shell.execute_reply":"2023-12-17T16:43:46.374485Z"}}

```
df['ALB'].fillna(df['ALB'].mean(), inplace=True)

df['ALP'].fillna(df['ALP'].mean(), inplace=True)

df['CHOL'].fillna(df['CHOL'].mean(), inplace=True)

df['PROT'].fillna(df['PROT'].mean(), inplace=True)

df['ALT'].fillna(df['ALT'].mean(), inplace=True)

print(df.isnull().sum())
```

%% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.376355Z","iopub.execute_input":"2023-12-17T16:43:46.377229Z","iopub.status.idle":"2023-12-17T16:43:46.388104Z","shell.execute_reply.started":"2023-12-17T16:43:46.377198Z","shell.execute_reply":"2023-12-17T16:43:46.387412Z"}}

```
from scipy.stats import zscore

# Визначення порогу для Z-оцінки (зазвичай поріг 3 вважається значущим)

z_threshold = 3

# Обчислення Z-оцінок для кожної колонки

z_scores = zscore(df.select_dtypes(include=['float64']))

# Виявлення аномалій

anomalies = (z_scores > z_threshold).any(axis=1)

# Виведення кількості та видалення аномалій

print("Кількість аномалій:", anomalies.sum())

df = df[~anomalies].reset_index(drop=True)
```

%% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.389259Z","iopub.execute_input":"2023-12-17T16:43:46.389948Z","iopub.status.idle":"2023-12-17T16:43:46.399402Z","shell.execute_reply.started":"2023-12-17T16:43:46.389903Z","shell.execute_reply":"2023-12-17T16:43:46.398212Z"}}

```
df['Category'] = df['Category'].replace({'0=Blood Donor': 0, '0s=suspect Blood Donor': 0, '1=Hepatitis': 1, '2=Fibrosis': 1, '3=Cirrhosis': 1})

df['Sex'] = df['Sex'].replace({'m': 0, 'f': 1})
```

%% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.401012Z","iopub.execute_input":"2023-12-17T16:43:46.401415Z","iopub.status.idle":"2023-12-17T16:43:46.434225Z","shell.execute_reply.started":"2023-12-17T16:43:46.401362Z","shell.execute_reply":"2023-12-17T16:43:46.433199Z"}}

```
df
```

```
# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.435710Z","iopub.execute_input":"2023-12-17T16:43:46.436410Z","iopub.status.idle":"2023-12-17T16:43:46.444788Z","shell.execute_reply.started":"2023-12-17T16:43:46.436349Z","shell.execute_reply":"2023-12-17T16:43:46.443787Z"}}

# Value counts of categorical variables

print(df['Category'].value_counts())

print(df['Sex'].value_counts())

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.495997Z","iopub.execute_input":"2023-12-17T16:43:46.496420Z","iopub.status.idle":"2023-12-17T16:43:46.519760Z","shell.execute_reply.started":"2023-12-17T16:43:46.496360Z","shell.execute_reply":"2023-12-17T16:43:46.518667Z"}}

col=['Category', 'Age', 'Sex', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE',

     'CHOL', 'CREA', 'GGT', 'PROT']

q_low = df[col].quantile(0.01)

q_hi  = df[col].quantile(0.99)

df_outliers = df[(df[col] < q_low) | (df[col] > q_hi)]

outlier_percentage = (df_outliers.sum() / len(df_outliers)) * 100

print(outlier_percentage)

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.522105Z","iopub.execute_input":"2023-12-17T16:43:46.522770Z","iopub.status.idle":"2023-12-17T16:43:46.542059Z","shell.execute_reply.started":"2023-12-17T16:43:46.522735Z","shell.execute_reply":"2023-12-17T16:43:46.540986Z"}}

from sklearn.preprocessing import RobustScaler

# Create a RobustScaler object

robust_scaler = RobustScaler()

# Define the columns to be scaled using RobustScaler

cols_to_scale = ['ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT']

# Scale the selected columns using RobustScaler

df[cols_to_scale] = robust_scaler.fit_transform(df[cols_to_scale])

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.543622Z","iopub.execute_input":"2023-12-17T16:43:46.544221Z","iopub.status.idle":"2023-12-17T16:43:46.575647Z","shell.execute_reply.started":"2023-12-17T16:43:46.544182Z","shell.execute_reply":"2023-12-17T16:43:46.574656Z"}}

q_low = df[col].quantile(0.01)

q_hi  = df[col].quantile(0.99)

df_outliers = df[(df[col] < q_low) | (df[col] > q_hi)]

outlier_percentage = (df_outliers.sum() / len(df_outliers)) * 100

print(outlier_percentage)

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.577285Z","iopub.execute_input":"2023-12-17T16:43:46.577921Z","iopub.status.idle":"2023-12-17T16:43:46.703052Z","shell.execute_reply.started":"2023-12-17T16:43:46.577889Z","shell.execute_reply":"2023-12-17T16:43:46.702007Z"}}

fig, ax = plt.subplots(figsize=(6,6))

plt.pie(x=df["Sex"].value_counts(),

        colors=["skyblue","pink"],
```

```
        labels=["Чоловік","Жінка"],
        shadow = True,
        autopct="%1.2f%%",
        explode = (0, 0.1)
        )
plt.show()


# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.706095Z","iopub.execute_input":"2023-12-17T16:43:46.706631Z","iopub.status.idle":"2023-12-17T16:43:46.910521Z","shell.execute_reply.started":"2023-12-17T16:43:46.706601Z","shell.execute_reply":"2023-12-17T16:43:46.909176Z"}}
# Comparação entre pacientes Saudáveis e suspeitos:
plt.style.use('seaborn')
fig,ax = plt.subplots(figsize=(6,6))
plt.pie(x= df['Category'].value_counts(),
        colors=['turquoise','salmon'],
        labels=['Здоровий',"Гепатит"],
        shadow= True,
        explode = (0.1,0)


        )
plt.show()
# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:46.912361Z","iopub.execute_input":"2023-12-17T16:43:46.913497Z","iopub.status.idle":"2023-12-17T16:43:47.821666Z","shell.execute_reply.started":"2023-12-17T16:43:46.913440Z","shell.execute_reply":"2023-12-17T16:43:47.820703Z"}}
fig, ax = plt.subplots(figsize=(10,8))
sns.heatmap(df.corr(), annot=True, fmt='.1g', cmap="Blues_r", cbar=False, linewidths=0.5, linecolor='grey');
# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:47.823035Z","iopub.execute_input":"2023-12-17T16:43:47.823443Z","iopub.status.idle":"2023-12-17T16:43:52.243500Z","shell.execute_reply.started":"2023-12-17T16:43:47.823406Z","shell.execute_reply":"2023-12-17T16:43:52.242404Z"}}
fig, ax =plt.subplots(5,2, figsize=(20,25))
plt.style.use("classic")
sns.histplot(x = df["Age"], hue = df["Category"], palette="viridis", kde=True, ax=ax[0,0]);
ax[0,0].set_xlabel("Age",fontsize=15)
sns.histplot(x = df["ALB"], hue = df["Category"], palette="viridis", kde=True, ax=ax[0,1]);
ax[0,1].set_xlabel("ALB",fontsize=15)
sns.histplot(x = df["ALP"], hue = df["Category"], palette="dark", kde=True, ax=ax[1,0]);
ax[1,0].set_xlabel("ALP",fontsize=15)
sns.histplot(x = df["ALT"], hue = df["Category"], palette="dark", kde=True, ax=ax[1,1]);
```

```
ax[1,1].set_xlabel("ALT",fontsize=15)
sns.histplot(x = df["AST"], hue = df["Category"], palette="flare", kde=True, ax=ax[2,0]);
ax[2,0].set_xlabel("AST",fontsize=15)
sns.histplot(x = df["BIL"], hue = df["Category"], palette="flare", kde=True, ax=ax[2,1]);
ax[2,1].set_xlabel("BIL",fontsize=15)
sns.histplot(x = df["CHE"], hue = df["Category"], palette="viridis", kde=True, ax=ax[3,0]);
ax[3,0].set_xlabel("CHE",fontsize=15)
sns.histplot(x = df["CHOL"], hue = df["Category"], palette="viridis", kde=True, ax=ax[3,1]);
ax[3,1].set_xlabel("CHOL",fontsize=15);
sns.histplot(x = df["CREA"], hue = df["Category"], palette="dark", kde=True, ax=ax[4,0]);
ax[4,0].set_xlabel("CREA",fontsize=15)
sns.histplot(x = df["GGT"], hue = df["Category"], palette="dark", kde=True, ax=ax[4,1]);
ax[4,1].set_xlabel("GGT",fontsize=15);
# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:52.245222Z","iopub.execute_input":"2023-12-17T16:43:52.245925Z","iopub.status.idle":"2023-12-17T16:43:52.267436Z","shell.execute_reply.started":"2023-12-17T16:43:52.245893Z","shell.execute_reply":"2023-12-17T16:43:52.266103Z"}}
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
# Assuming df is loaded somewhere before this point
# Завантаження даних та подальша обробка
df['Category'] = df['Category'].replace({'0=Blood Donor': 'Blood Donor',
                                         '0s=suspect Blood Donor': 'Blood Donor'})
# Вибір колонок для ознак та цільової змінної
columns = ["Age", "ALB", "ALP", "ALT", "AST", "BIL", "CHE", "CHOL", "CREA", "GGT", "Category"]
df_subset = df[columns]
# Розділення даних на тренувальний та тестовий набори
X = df_subset.drop("Category", axis=1)
y = df_subset["Category"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=42, stratify=y)
# Нормалізація даних
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:52.268802Z","iopub.execute_input":"2023-12-17T16:43:52.269662Z","iopub.status.idle":"2023-12-17T16:43:52.289936Z","shell.execute_reply.started":"2023-12-17T16:43:52.269621Z","shell.execute_reply":"2023-12-17T16:43:52.289058Z"}}

import numpy as np

from sklearn.model_selection import train_test_split

from imblearn.over_sampling import SMOTE

# Convert y to float values

conditions = [y == 'Healthy', y == 'Hepatitis']

choices = [0.0, 1.0]

y_float = np.select(conditions, choices, default=y)

# Split data into train, validation, and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y_float, test_size=0.3, random_state=8)

X_val, X_test, y_val, y_test = train_test_split(X_test, y_test, test_size=0.5, random_state=8)

# SMOTE for class balancing

sm = SMOTE(random_state=8)

# Create a new training set with SMOTE object

X_bal, y_bal = sm.fit_resample(X_train, y_train)

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:52.291325Z","iopub.execute_input":"2023-12-17T16:43:52.292420Z","iopub.status.idle":"2023-12-17T16:43:52.616239Z","shell.execute_reply.started":"2023-12-17T16:43:52.292352Z","shell.execute_reply":"2023-12-17T16:43:52.615136Z"}}

# Логістична регресія

logistic_regression_model = LogisticRegression(random_state=42, max_iter=1000)

logistic_regression_model.fit(X_train, y_train)

y_pred_lr = logistic_regression_model.predict(X_bal)

# Оцінка моделі

print("Логістична регресія:")

print("Accuracy:", accuracy_score(y_bal, y_pred_lr))

print("Classification Report:\n", classification_report(y_bal, y_pred_lr))

# Виведення кількості пацієнтів для кожної категорії окремо

category_counts = df['Category'].value_counts()

from sklearn.metrics import confusion_matri

conf_matrix_lr = confusion_matrix(y_bal, y_pred_lr)

# Plotting the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_lr, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Donor', 'Liver Disease'])
```

```
plt.title('Confusion Matrix for Logistic Regression')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:52.617821Z","iopub.execute_input":"2023-12-17T16:43:52.618821Z","iopub.status.idle":"2023-12-17T16:43:53.446369Z","shell.execute_reply.started":"2023-12-17T16:43:52.618780Z","shell.execute_reply":"2023-12-17T16:43:53.444881Z"}}

# RandomForestClassifier зі збільшеною кількістю дерев

random_forest_model = RandomForestClassifier(random_state=42, n_estimators=200, max_depth=None,

                                             min_samples_split=2, min_samples_leaf=1)

random_forest_model.fit(X_train, y_train)

y_pred_rf = random_forest_model.predict(X_bal)

# Оцінка моделі

print("Random Forest Classifier:")

print("Accuracy:", accuracy_score(y_bal, y_pred_rf))

print("Classification Report:\n", classification_report(y_bal, y_pred_rf))

conf_matrix_rf = confusion_matrix(y_bal, y_pred_rf)

# Plotting the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_rf, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Donor', 'Liver Disease'])

plt.title('Confusion Matrix for RFC')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:53.448635Z","iopub.execute_input":"2023-12-17T16:43:53.450792Z","iopub.status.idle":"2023-12-17T16:43:53.955356Z","shell.execute_reply.started":"2023-12-17T16:43:53.450730Z","shell.execute_reply":"2023-12-17T16:43:53.953951Z"}}

# Gradient Boosting Classifier

gradient_boosting_model = GradientBoostingClassifier(random_state=42, n_estimators=100, learning_rate=0.1, max_depth=3)

# Навчання моделі на всій тренувальній вибірці

gradient_boosting_model.fit(X_train, y_train)

y_pred_gb = gradient_boosting_model.predict(X_bal)

# Оцінка моделі

print("Gradient Boosting Classifier:")

print("Accuracy:", accuracy_score(y_bal, y_pred_gb))

print("Classification Report:\n", classification_report(y_bal, y_pred_gb))
```

```
conf_matrix_gb = confusion_matrix(y_bal, y_pred_gb)

# Plotting the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_gb, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Donor', 'Liver Disease'])

plt.title('Confusion Matrix for GB')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:53.957205Z","iopub.execute_input":"2023-12-17T16:43:53.957799Z","iopub.status.idle":"2023-12-17T16:43:54.266101Z","shell.execute_reply.started":"2023-12-17T16:43:53.957759Z","shell.execute_reply":"2023-12-17T16:43:54.264716Z"}}

# Support Vector Machine (SVM) Classifier

svm_model = SVC(random_state=42, kernel='linear')

# Train the model on the entire training set

svm_model.fit(X_train, y_train)

y_pred_svm = svm_model.predict(X_bal)

# Evaluate the model

# Note: R2 score is not suitable for classification problems; using accuracy for demonstration

print("Support Vector Machine Classifier:")

print("Accuracy:", accuracy_score(y_bal, y_pred_svm))

print("Classification Report:\n", classification_report(y_bal, y_pred_svm))

conf_matrix_svm = confusion_matrix(y_bal, y_pred_svm)

# Plotting the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_svm, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Donor', 'Liver Disease'])

plt.title('Confusion Matrix for SVM')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:54.271341Z","iopub.execute_input":"2023-12-17T16:43:54.271967Z","iopub.status.idle":"2023-12-17T16:43:54.574555Z","shell.execute_reply.started":"2023-12-17T16:43:54.271934Z","shell.execute_reply":"2023-12-17T16:43:54.573026Z"}}

from sklearn.tree import DecisionTreeClassifier

# Decision Tree Classifier

decison_tree_model = DecisionTreeClassifier(random_state=42, max_depth=3)

# Train the model on the entire training set
```

```
decision_tree_model.fit(X_train, y_train)

y_pred_dt = decision_tree_model.predict(X_bal)

# Evaluate the model

print("Decision Tree Classifier:")

print("Accuracy:", accuracy_score(y_bal, y_pred_dt))

print("Classification Report:\n", classification_report(y_bal, y_pred_dt))

conf_matrix_dt = confusion_matrix(y_bal, y_pred_dt)

# Plotting the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_dt, annot=True, fmt='g', cmap='Blues', xticklabels=['Blood Donor', 'Liver Disease'], yticklabels=['Blood Donor', 'Liver Disease'])

plt.title('Confusion Matrix for Decision Tree Classifier')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:50:48.968058Z","iopub.execute_input":"2023-12-17T16:50:48.968515Z","iopub.status.idle":"2023-12-17T16:50:49.226658Z","shell.execute_reply.started":"2023-12-17T16:50:48.968480Z","shell.execute_reply":"2023-12-17T16:50:49.225457Z"}}

models = ['Logistic Regression', 'Random Forest', 'Gradient Boosting', 'SVM', 'Decision Tree']

accuracies = [accuracy_score(y_bal, y_pred_lr),

        accuracy_score(y_bal, y_pred_rf),

        accuracy_score(y_bal, y_pred_gb),

        accuracy_score(y_bal, y_pred_svm),

        accuracy_score(y_bal, y_pred_dt)]

# Створення гістограми

plt.figure(figsize=(10, 6))

sns.set(style="whitegrid", font_scale=1.2)

colors = sns.color_palette("mako", n_colors=len(models))  # Змінено палітру кольорів

# Створення списку кортежів для моделей та їхніх точностей

model_accuracies = list(zip(models, accuracies))

# Сортування за спаданням точностей

model_accuracies.sort(key=lambda x: x[1], reverse=True)

# Розпакування відсортованого списку

sorted_models, sorted_accuracies = zip(*model_accuracies)

# Створення гістограми з впорядкованими точностями та відповідними назвами моделей

bar_plot = sns.barplot(x=list(sorted_accuracies), y=list(sorted_models), palette=colors)

# Додавання значень до барів
```

```
for index, value in enumerate(sorted_accuracies):
    bar_plot.text(value, index, f'{value:.3f}', ha="left", va="center", color='black', fontsize=10)
# Додаткові налаштування
plt.xlabel('Accuracy')
plt.title('Comparison of Model Accuracies (Descending Order)')
plt.xlim(0, 1.0)
# Додаткові лінії сітки
bar_plot.xaxis.grid(True)
# Видалення рамок
sns.despine(left=True, bottom=True)
# Відображення графіку
plt.show()
# %% [code] {"execution":{"iopub.status.busy":"2023-12-17T16:43:54.756031Z","iopub.status.idle":"2023-12-17T16:43:54.756877Z","shell.execute_reply.started":"2023-12-17T16:43:54.756584Z","shell.execute_reply":"2023-12-17T16:43:54.756613Z"}}
# Створення DataFrame з результатами
data = {'Model': models, 'Accuracy': accuracies}
df_results = pd.DataFrame(data)

# Виведення таблиці у вигляді, схожому на Excel
df_results
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ СТАНУ ХВОРИХ НА ГЕПАТИТ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2023 р.

Вінниця 2023

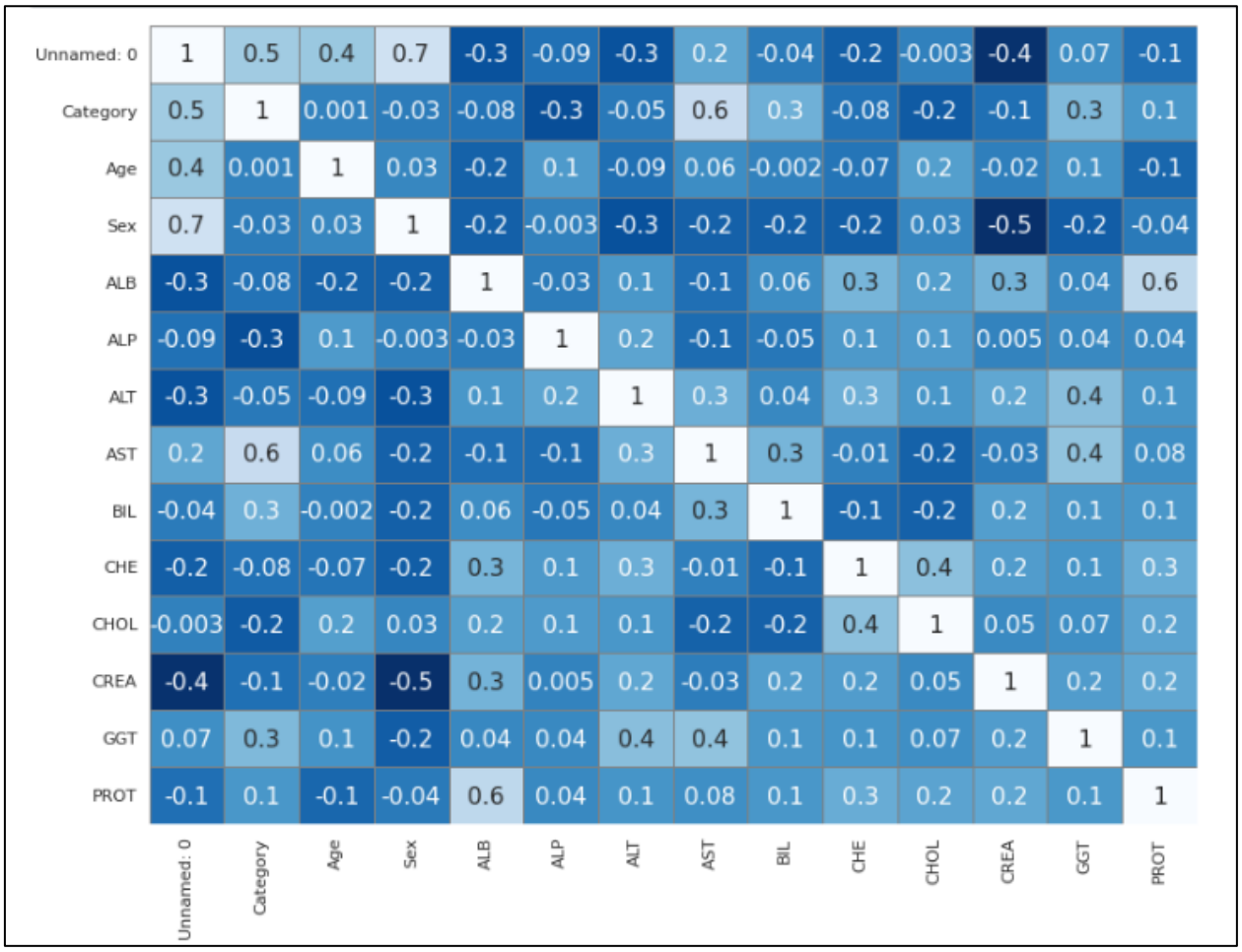


Рисунок Г.1 – Кореляційна матриця показників.

```
Логістична регресія:
Accuracy: 0.889196675900277
Classification Report:
              precision    recall  f1-score   support

         0.0       0.82      0.99      0.90       361
         1.0       0.99      0.78      0.88       361

    accuracy                           0.89       722
   macro avg       0.91      0.89      0.89       722
weighted avg       0.91      0.89      0.89       722
```

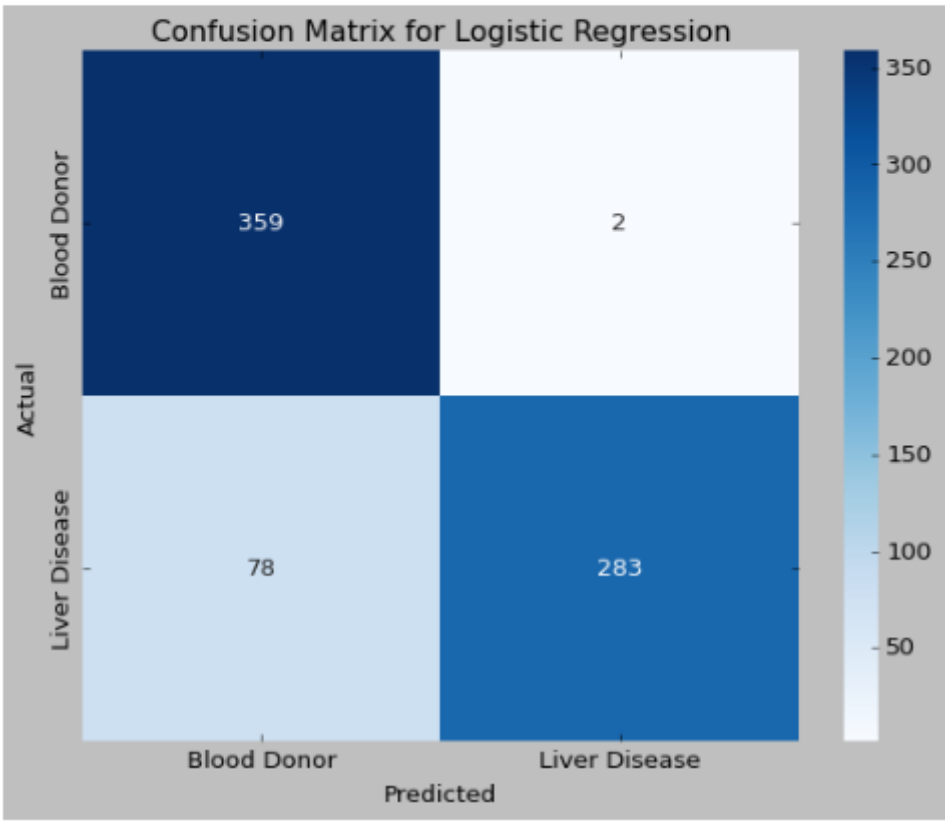


Рисунок Г.2 – Матриця плутанини моделі «Logistic Regression»

```
Random Forest Classifier:
Accuracy: 0.9736842105263158
Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.95      0.97       361
         1.0       0.95      1.00      0.97       361

    accuracy                           0.97       722
   macro avg       0.97      0.97      0.97       722
weighted avg       0.97      0.97      0.97       722
```

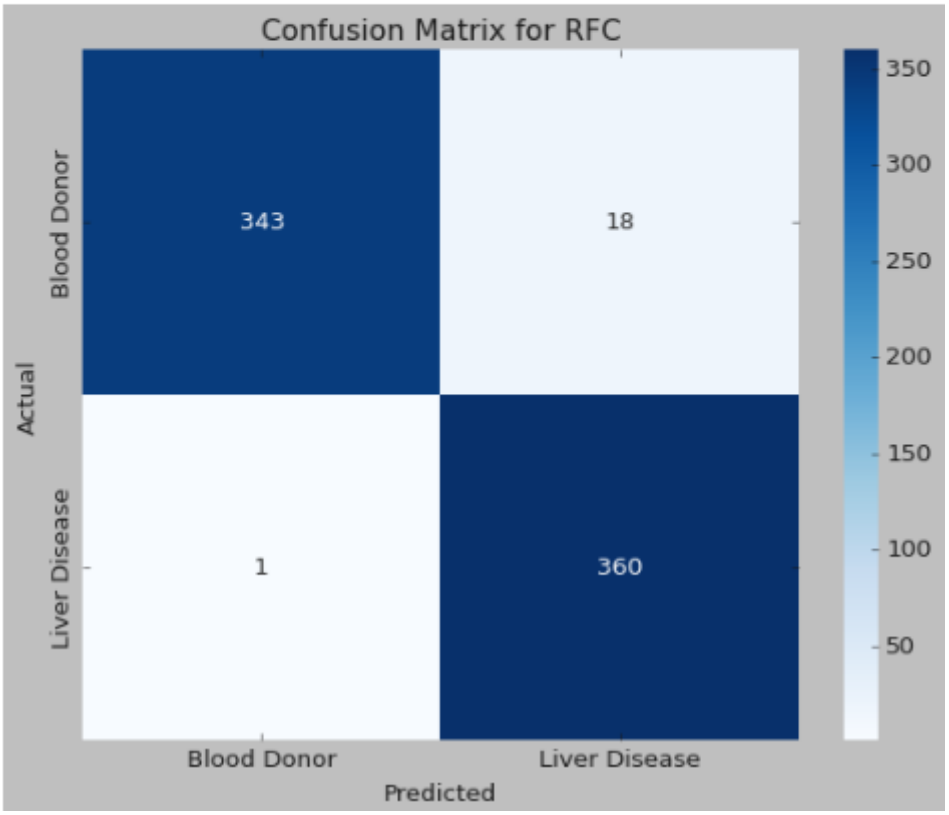


Рисунок Г.3 – Матриця плутанини моделі «Random Forest Classifier»

```
Gradient Boosting Classifier:
Accuracy: 0.9487534626038782
Classification Report:
              precision    recall  f1-score   support

         0.0       0.91      1.00      0.95       361
         1.0       1.00      0.90      0.95       361

    accuracy                           0.95       722
   macro avg       0.95      0.95      0.95       722
weighted avg       0.95      0.95      0.95       722
```

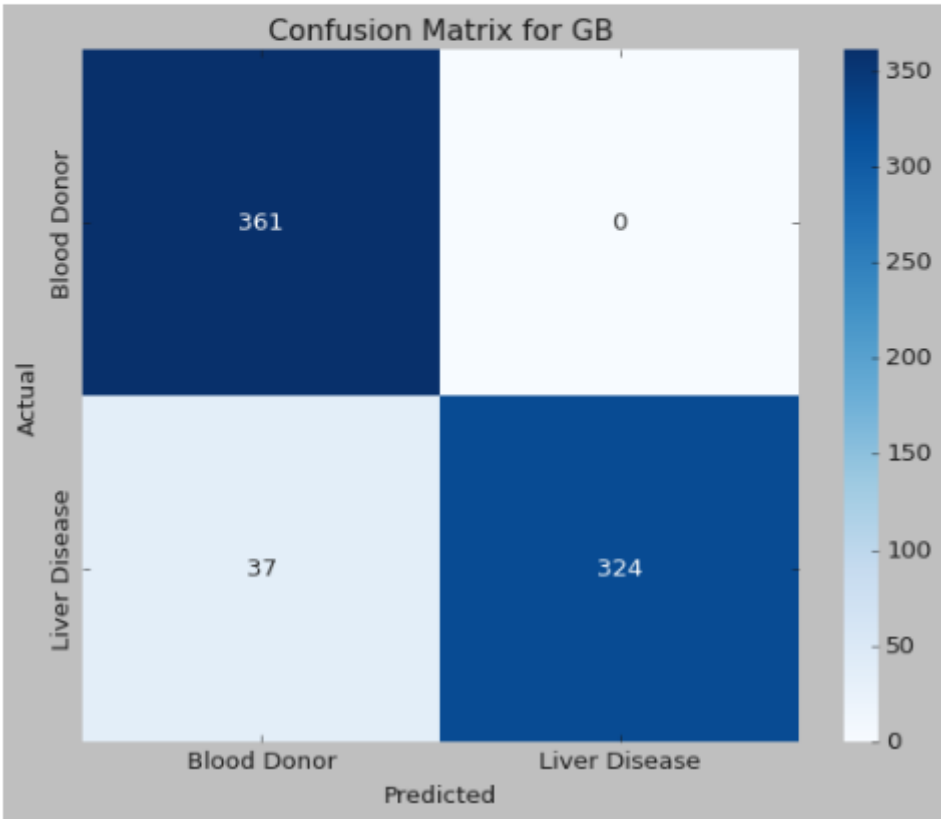


Рисунок Г.4 – Матриця плутанини моделі «Gradient Boosting»

```
Support Vector Machine Classifier:
Accuracy: 0.9238227146814404
Classification Report:
              precision    recall  f1-score   support

         0.0       0.88      0.99      0.93       361
         1.0       0.99      0.86      0.92       361

    accuracy                           0.92       722
   macro avg       0.93      0.92      0.92       722
weighted avg       0.93      0.92      0.92       722
```

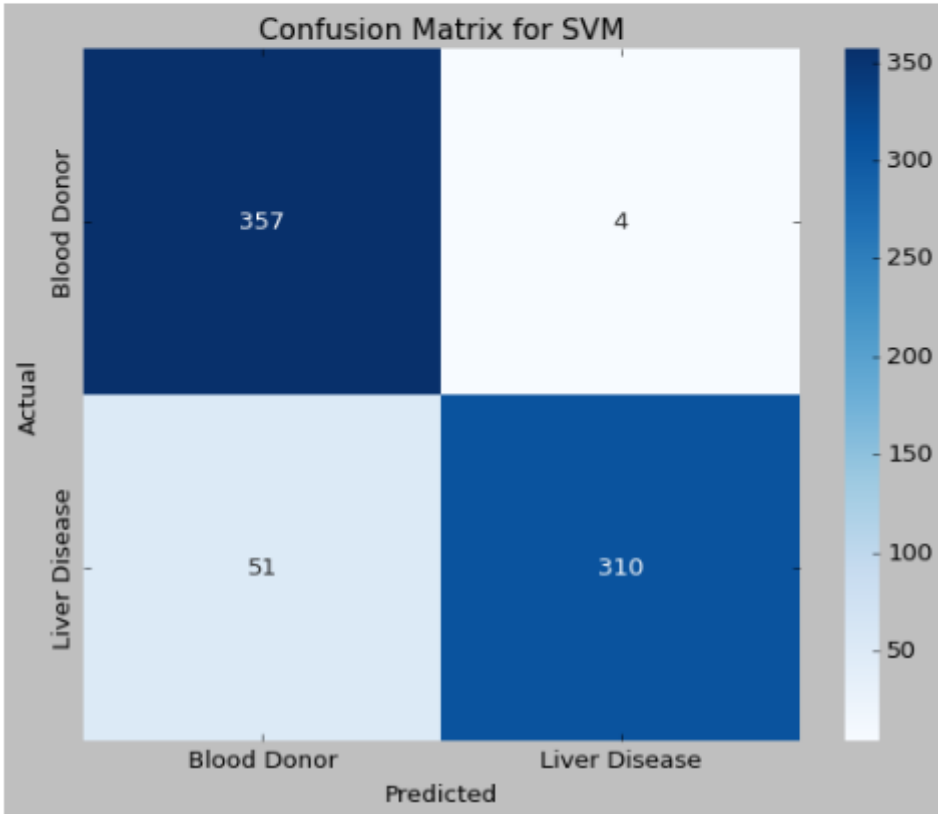


Рисунок Г.5 – Матриця плутанини моделі «SVM»

```
Decision Tree Classifier:
Accuracy: 0.8836565096952909
Classification Report:
              precision    recall  f1-score   support

         0.0       0.81      1.00      0.90       361
         1.0       1.00      0.77      0.87       361

    accuracy                           0.88       722
   macro avg       0.91      0.88      0.88       722
weighted avg       0.91      0.88      0.88       722
```

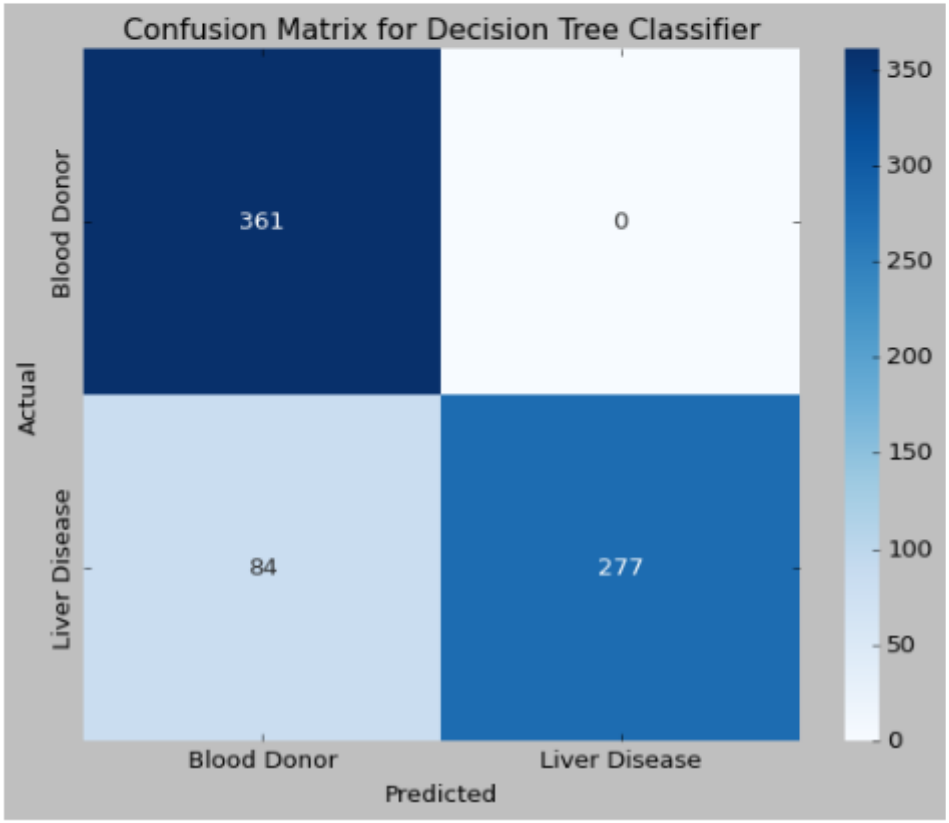


Рисунок Г.6 – Матриця плутанини моделі «DecisionTreeClassifier»

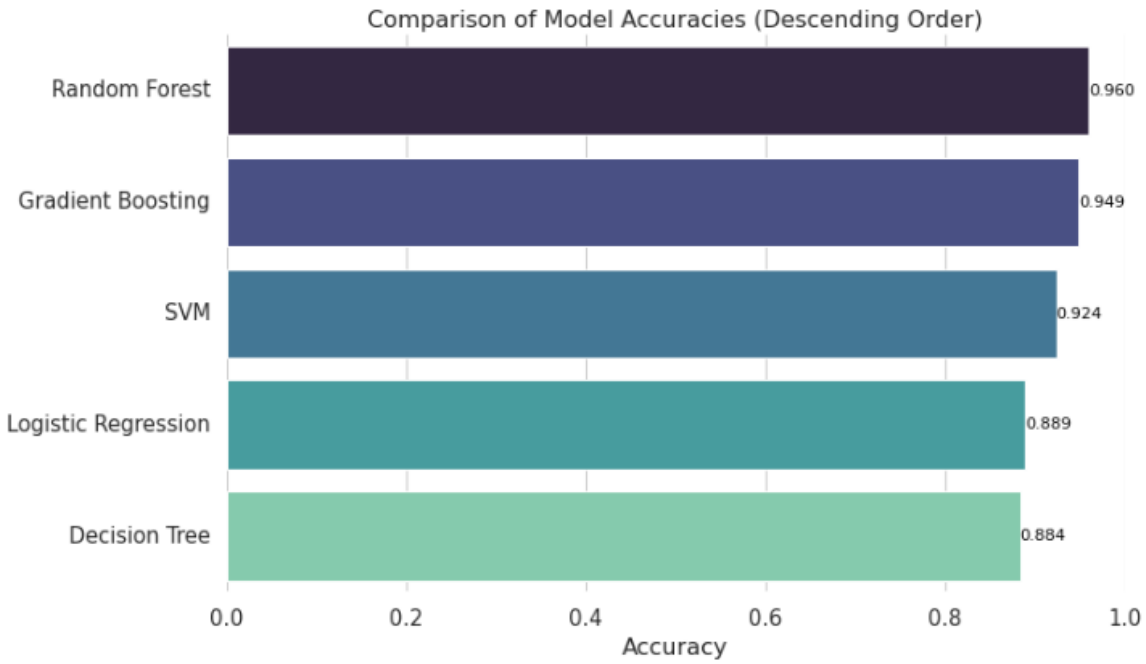


Рисунок Г.7 – Графік порівняння результатів класифікації

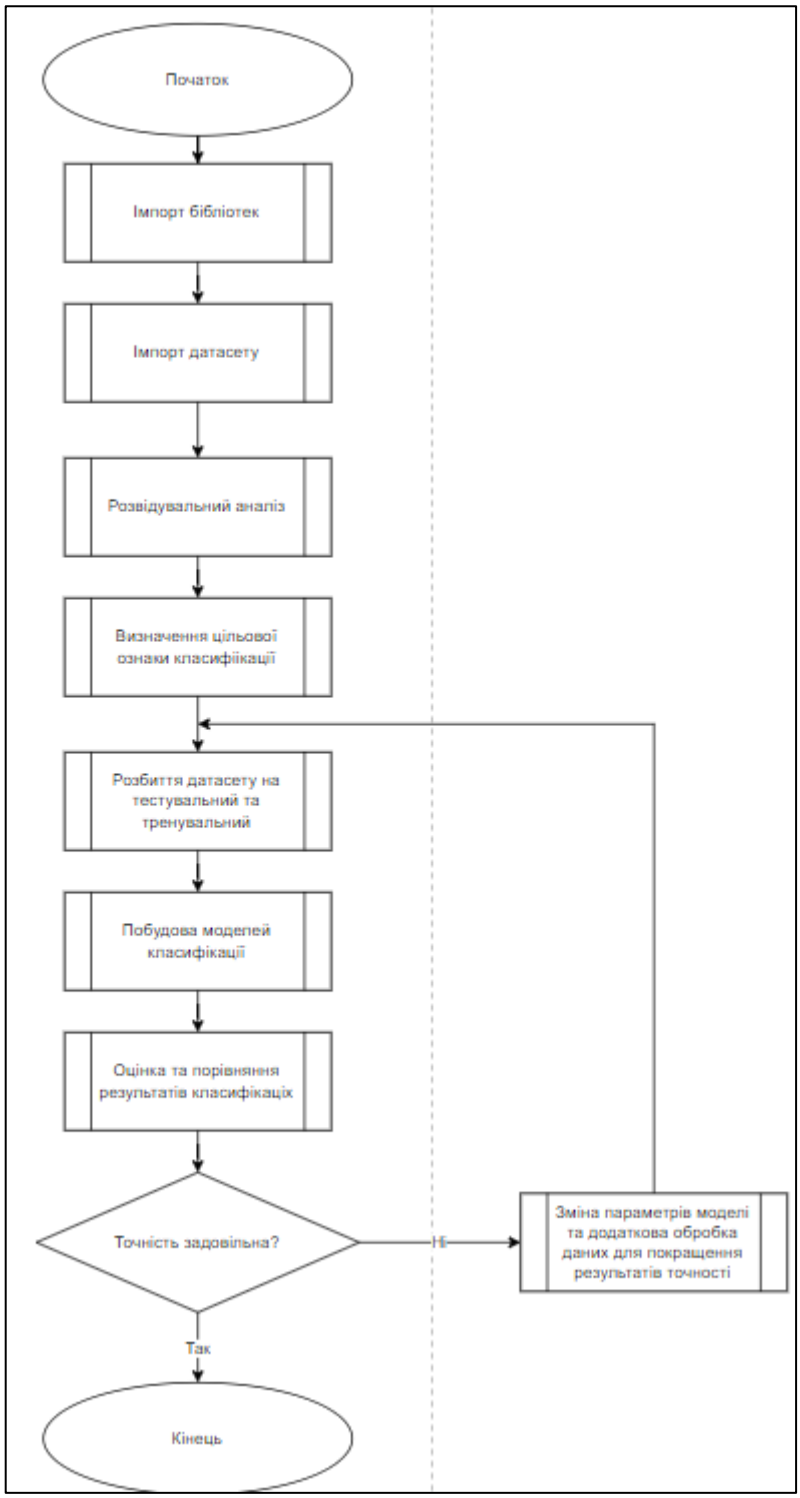


Рисунок Г.8 – Схема алгоритму.