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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія передбачення хворих на діабет»

Виконав: студент 2 курсу, групи 1ІСТ-23м спеціальності 126 – «Інформаційні системи та технології»
_____Іван ЯКИМЧУК
Керівник: к.т.н., доц. каф. САІТ
_____Сергій ЖУКОВ
«05» 12 2024 р.
Рецензент: к.т.н., доц. каф. КН
_____Володимир ОЗЕРАНСЬКИЙ
«08» 12 2024 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«06» 12 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – другий (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. Віталій МОКІН
"17" 09 2024 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Якимчуку Івану Володимировичу

1. Тема роботи: «Інформаційна технологія передбачення хворих на діабет»
керівник роботи: Сергій ЖУКОВ, к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «17» 09 2024 року №310
2. Термін подання студентом роботи 01. 12. 2024р.
3. Вихідні дані до роботи:
1) датасет, який містить дані, необхідні для передбачення стану хворих на діабет.
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) Візуалізація даних датасету;
   2) Кореляційна матриця;
   3) Результати передбачення різними моделями.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийня
4	Олександр ЛЕСЬКО, к.е.н., проф. каф. ЕПВМ	30.11.24	30.11.24

7. Дата видачі завдання 17.09.2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Приміт
		початок	закінчення	
1	Аналіз предметної області	30.09	15.10	вик
2	Аналіз методів машинного навчання для вирішення поставленої задачі	16.10	31.10	вик
3	Розробка моделі класифікації	01.11	15.11	вик
4	Економічна частина	16.11	23.11	вик
5	Оформлення матеріалів до захисту МКР	25.11	30.11	вик

Студент _____ Іван ЯКИМЧУК

Керівник роботи _____ Сергій ЖУКОВ

АНОТАЦІЯ

УДК 004.623

Якимчук І. В. Інформаційна технологія передбачення хворих на діабет. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2024. 105 с.

На укр. мові. Бібліогр.: 22 назв; рис.: 33 ; табл.: 12.

У магістерській кваліфікаційній роботі проведено огляд та аналіз різноманітних методів машинного навчання, що можуть бути використані для класифікації даних. Проведено порівняльний аналіз алгоритмів для визначення оптимального підходу. Проведено розвідувальний аналіз даних та побудовано візуалізації даних. Побудовано моделі для передбачення та розроблено алгоритм інформаційної технології для передбачення стану хворих на діабет.

Ілюстративна частина складається з 6 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 діабет.

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

ABSTRACT

Yakymchuk I. V. Information technology for predicting diabetes patients. Master's Qualification Thesis in Specialty 126 – Information Systems and Technologies, Educational-Professional Program – Information Technologies for Data and Image Analysis. Vinnytsia: VNTU, 2024. 105 p.

In Ukrainian. Bibliography: 22 titles; illustrations: 33; tables: 12.

The master's qualification work reviews and analyzes various machine learning methods that can be used for data classification. A comparative analysis of algorithms was conducted to determine the optimal approach. Exploratory data analysis was conducted and data visualizations were built. Models for prediction were built and an information technology algorithm was developed to predict the condition of patients with diabetes.

The illustrative part consists of 6 posters, which include the results of testing the developed models.

The economic part of the work examines in detail the feasibility of developing and implementing information technology for predicting diabetes.

Keywords: Python, diabetes, exploratory analysis, disease.

ЗМІСТ

ВСТУП 4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПЕРЕДБАЧЕННЯ ДІАБЕТУ 6
1.1 Аналіз літератури передбачення діабету 6
1.2 Класифікація діабету 7
1.3 Передбачення діабету та її методи 15
1.4 Висновки 16
2 ВИБІР ОПТИМАЛЬНИХ МЕТОДІВ МАШИНОГО НАВЧАННЯ В ПЕРЕДБАЧЕННІ СТАНУ ХВОРИХ НА ДІАБЕТ 17
2.1 Огляд методів машинного навчання для передбачення стану хворих на діабет 17
2.2 Аналіз та порівняння методів класифікації в контексті діабету 24
2.3 Використання методів регресії для передбачення стану хворих на діабет 25
2.4 Висновки 30
3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ ПЕРЕДБАЧЕННЯ СТАНУ ХВОРИХ НА ДІАБЕТ 31
3.1 Розроблення алгоритму інформаційної технології 31
3.2 Аналіз набору даних про діабет 32
3.3 Розвідувальний аналіз 35
3.4 Класифікація моделлю KNN 47
3.5.Класифікація моделлю Random Forest 50
3.6 Класифікація моделлю SVM (Методом опорних векторів) 54
3.7 Аналіз результатів 58
3.8 Висновки 60
4 ЕКОНОМІЧНА ЧАСТИНА 62
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки 62
4.2 Розрахунок узагальненого коефіцієнта якості розробки 63
4.3 Розрахунок витрат на проведення науково-дослідної роботи 65
4.3.1 Витрати на оплату праці 65
4.3.2 Відрахування на соціальні заходи 68
4.3.3 Сировина та матеріали 68
4.3.4 Розрахунок витрат на комплектуючі 69
4.3.5 Спецустаткування для наукових (експериментальних) робіт 70

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт 71
4.3.7 Амортизація обладнання, програмних засобів та приміщень 72
4.3.8 Паливо та енергія для науково-виробничих цілей 73
4.3.9 Службові відрядження 75
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації 75
4.3.11 Інші витрати 75
4.3.12 Накладні (загальновиробничі) витрати 76
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором 77
4.5 Висновки 81
ВИСНОВКИ 83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 85
Додаток А (обов’язковий) Технічне завдання 88
Додаток Б (обов’язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень 90
Додаток В (довідниковий) Лістинг програми 91
Додаток Г (обов’язковий) Ілюстративна частина 102

ВСТУП

Актуальність теми. Діабет є одним із найбільш поширених захворювань у світі, яке має значний вплив на здоров'я та якість життя мільйонів людей. За даними ВООЗ, кількість діабетиків зростає з кожним роком, і це захворювання вже є однією з головних причин смертності у світі. В Україні кількість хворих на це захворювання також неухильно збільшується, що робить проблему діагностики та передбачення стану пацієнтів із діабетом надзвичайно актуальною.

Одним із перспективних інструментів для покращення діагностики та прогнозування стану хворих на діабет є методи машинного навчання. Використання цих методів дозволяє суттєво підвищити точність прогнозування ризику розвитку захворювання та ефективності лікування, що є важливим для своєчасного прийняття медичних рішень.

Мета і завдання роботи. Метою магістерської роботи є підвищення точності передбачення стану хворих на діабет методами машинного навчання. Для досягнення цієї мети передбачається виконати такі 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 є процес передбачення стану хворих на діабет за допомогою методів машинного навчання.

Предметом дослідження є методи машинного навчання що використовуються для передбачення стану хворих на діабет.

Методи дослідження. У ході роботи застосовуються методи машинного навчання для аналізу та прогнозування стану пацієнтів із діабетом. Також використовуються методи статистичного аналізу даних, оптимізації параметрів моделей та валідації їх точності.

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

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

Апробація результатів магістерської кваліфікаційної роботи. Результати кваліфікаційної роботи апробовані на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (м. Вінниця, 2024-2025 рр.).

Публікації результатів магістерської кваліфікаційної роботи. За результатами роботи було опубліковано тези на тему «Розвідувальний аналіз даних для інформаційної технології передбачення хворих на діабет» на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (м. Вінниця, 2024-2025 рр.) [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПЕРЕДБАЧЕННЯ ДІАБЕТУ

1.1 Аналіз літератури передбачення діабету

Діабет є однією з найбільш поширених хронічних захворювань, що впливає на здоров'я людей у всьому світі. Згідно з даними Всесвітньої організації охорони здоров'я, у 2021 році в світі понад 460 мільйонів людей мають діабет, і ця цифра продовжує зростати. Діабет характеризується підвищеним рівнем глюкози в крові, що може призвести до різних ускладнень, таких як серцево-судинні захворювання, захворювання нирок та сліпота [2]

На рисунку 1.1 зображені дані Число хворих діабетом станом на 18.11.2022

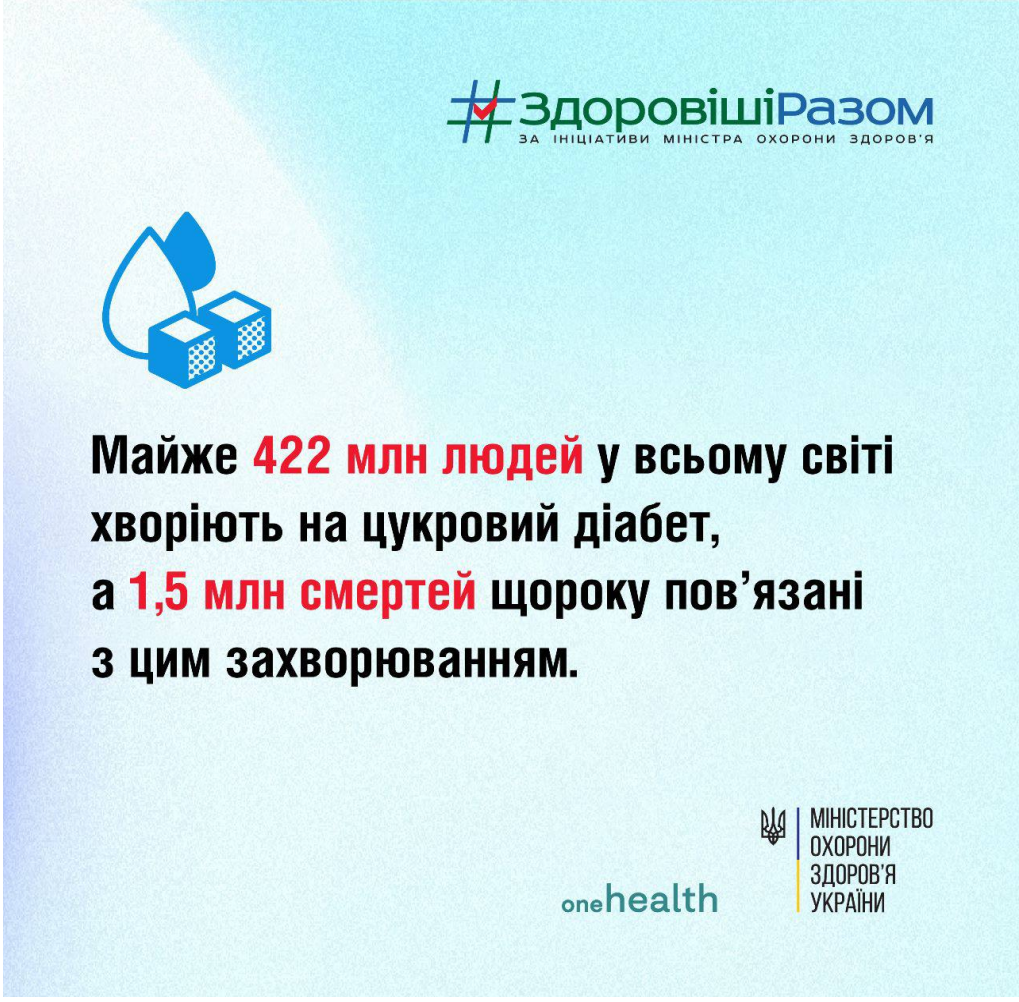

Рисунок 1.1 – Число хворих діабетом станом на 18.11.2022 р.

У літературі описано два основні типи діабету: перший тип діабету, або діабет залежний від інсуліну, і другий тип діабету, або діабет не залежний від інсуліну. Перший тип діабету розвивається, коли імунна система атакує та знищує клітини підшлункової залози, які виробляють інсулін. Другий тип діабету розвивається, коли організм не може правильно використовувати інсулін або не виробляє достатньо інсуліну [2].

1.2 Класифікація діабету

Згідно з Всесвітньою організацією охорони здоров'я, існують наступні типи діабету зображені на рисунку 1.2.

Рисунок 1.2 – Основні типи діабету

Діабет 1 типу - це автоімунне захворювання, при якому клітини підшлункової залози, що виробляють інсулін, знищуються. Це призводить до того, що організм не може виробляти достатню кількість інсуліну, що необідні, для діагностики діабету також використовуються тест на вміст гемоглобіну A1C та тест на толерантність до глюкози. Тест на вміст гемоглобіну A1C дає інформацію про середній рівень глюкози в крові за останні 2-3 місяці, а тест на толерантність до глюкози дозволяє виявити порушення толерантності до глюкози, що може бути попередником діабету [1].

На рисунку 1.3 зображено профілактика діабету.

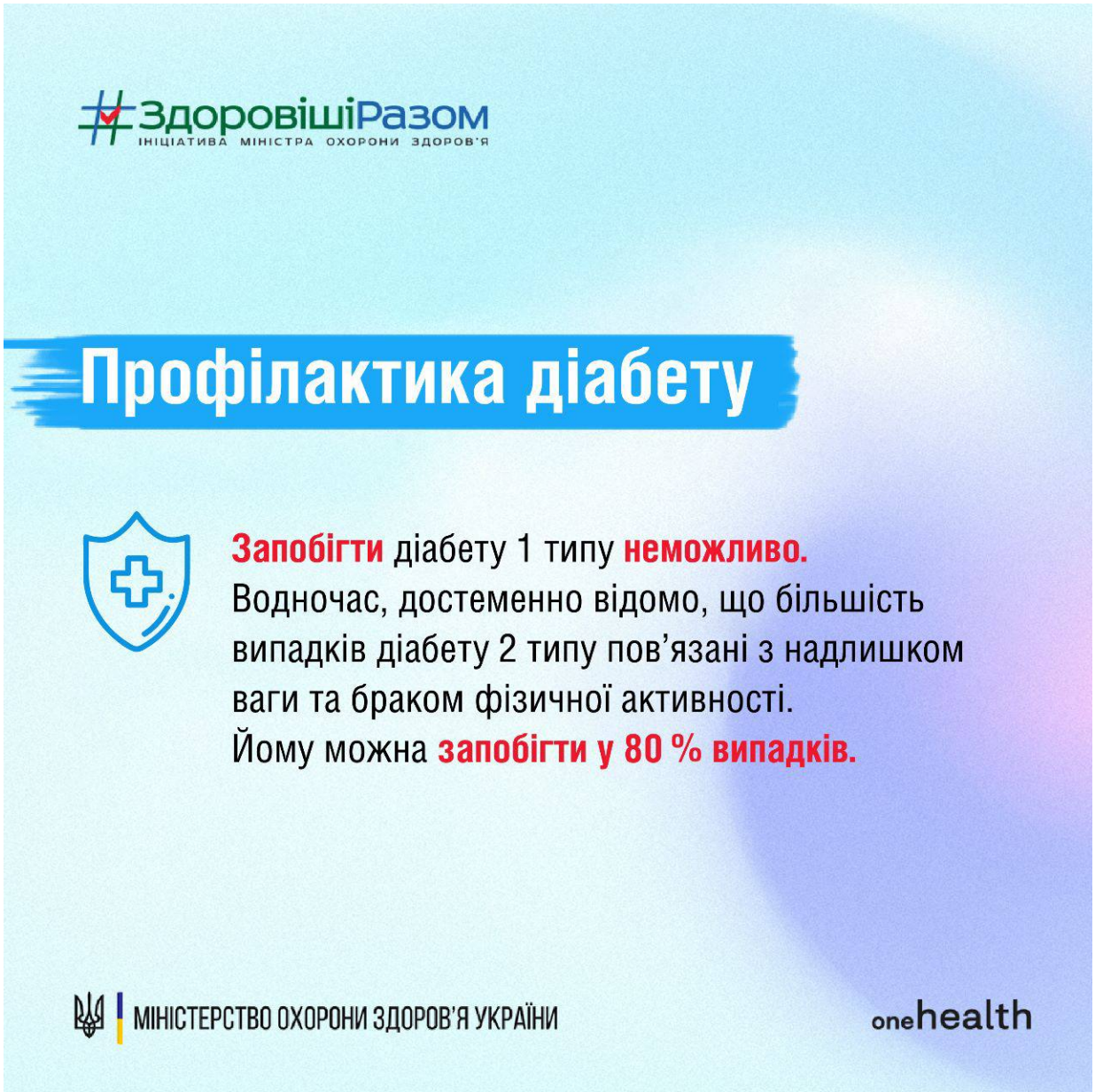

Рисунок 1.3 – Ймовірність запобігти діабету

Однак, зазвичай діагноз діабету ставлять на підставі результатів крові та сечі, що показують підвищений рівень глюкози в організмі.

Діабет 2 типу є найбільш поширеною формою діабету, що становить 90-95% від усіх випадків діабету у світі. На сьогодні кількість уражених ним становить 415 мільйонів людей по всьому світі, що спричиняє величезне навантаження на глобальні системи охорони здоров'я. Вже будучи серйозною глобальною пандемією, у діабету є потенціал до зростання. За попередніми прогнозами число хворих на цукровий діабет збільшиться до 642 мільйонів людей до 2040 року [1].

На рисунку 1.4 зображені дані відсоток людей з другим типом діабету:

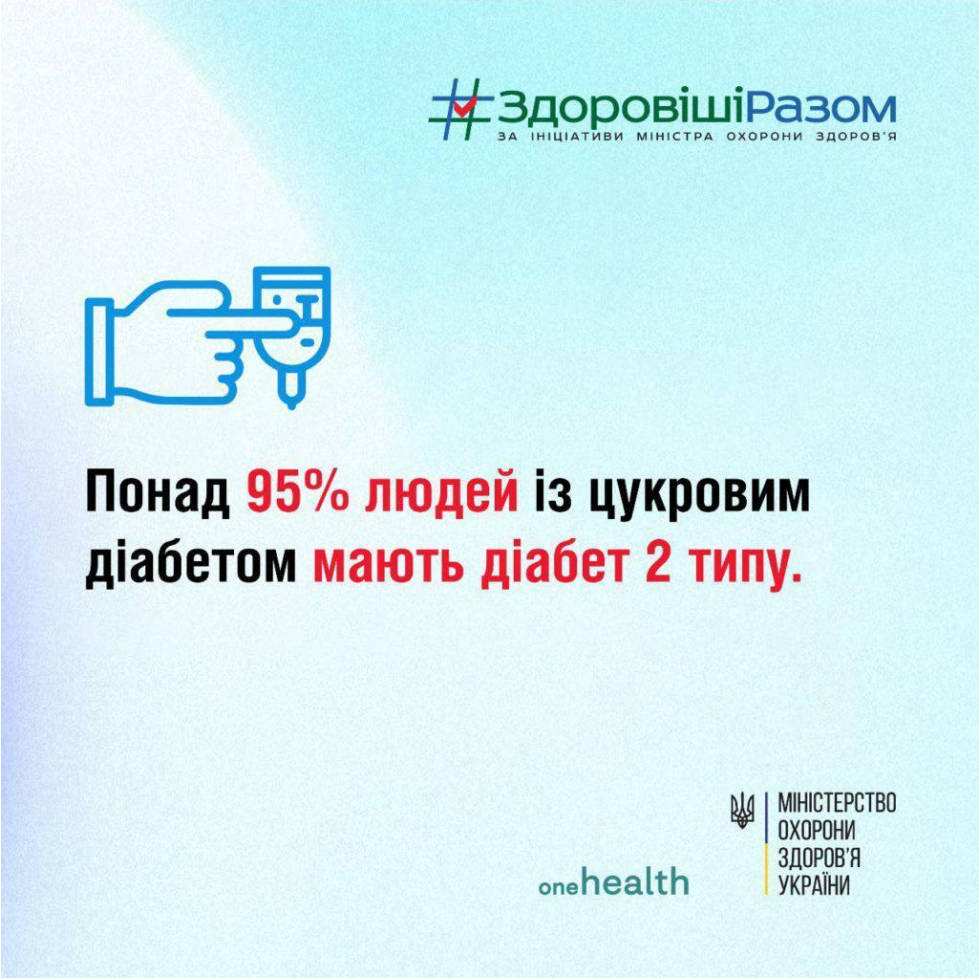

Рисунок 1.4 – Відсоток людей із 2 типом діабету станом на 18.11.2022

Він починається, коли м'язи та інші клітини перестають реагувати на сигнал інсуліну, який готовий приймати глюкозу. Тіло реагує, виробляючи все

більше і більше інсуліну, щоб допомогти вивести глюкозу з крові, але з часом виснажує клітини, що виробляють інсулін, поки вони не згорять [1].

На рисунку 1.5 зображено через що розвивається діабет:

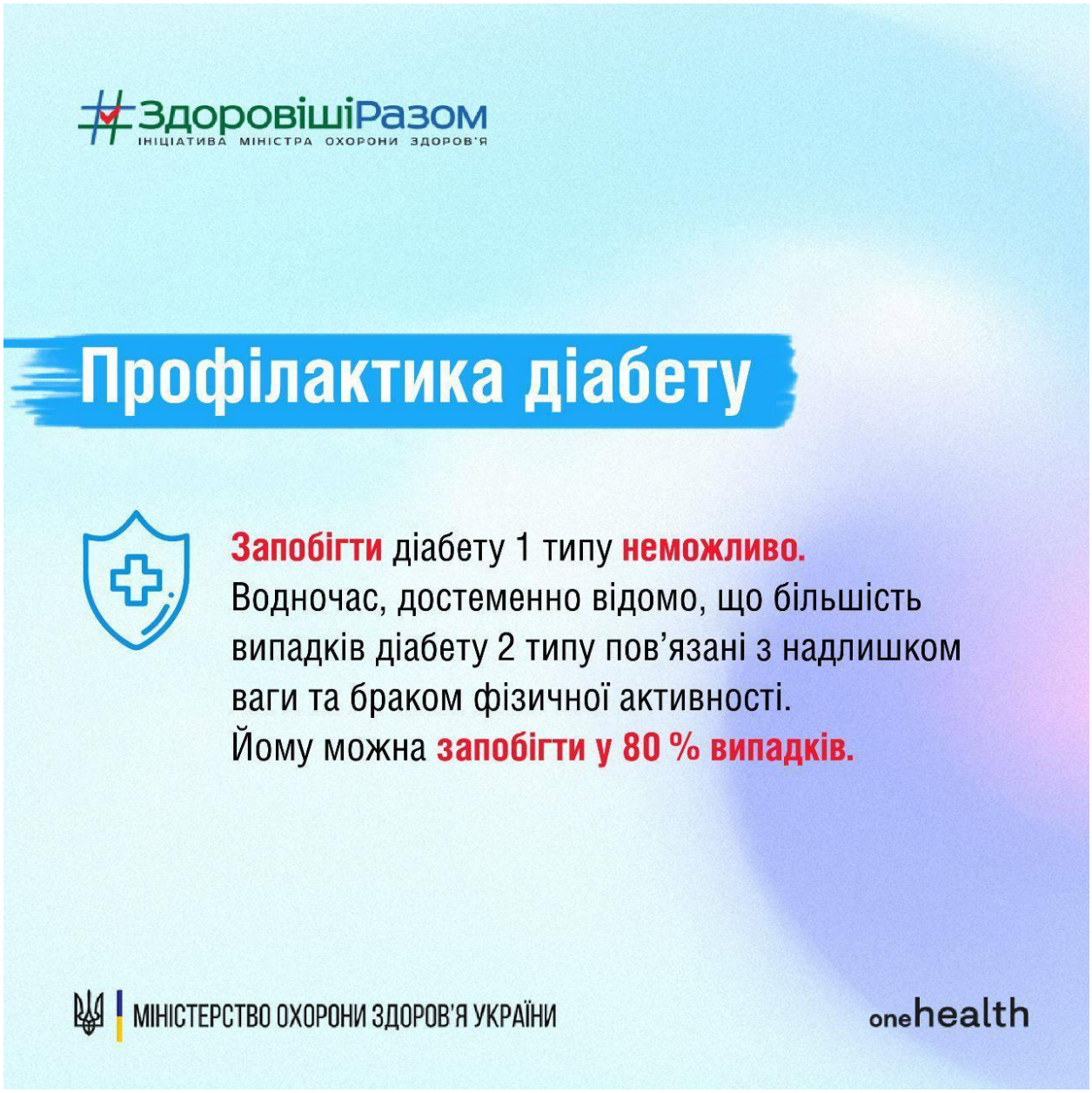

Рисунок 1.5 – Через що розвивається діабет 2 типу

Хоча генетичні мутації можуть сприяти розвитку діабету 2 типу, фактори навколишнього середовища, швидше за все, сприяють тому, що інсулін не працює належним чином[2]. Ці фактори включають надмірне збільшення ваги, дієту з високим вмістом рафінованих вуглеводів і насичених жирів і малорухливий спосіб життя. Також незрозуміло, чи сімейна історія цукрового

діабету 2 типу, спричинена генетичними мутаціями, сприяє розвитку наступних поколінь з хворобою, чи це скоріше питання нездорової поведінки, що передається дітям[3].

Діабет 2 типу може розвиватися повільно. Спочатку симптоми можуть бути слабкими і легко їх усунути. Ранні симптоми можуть включати:

- постійний голод;
- нестача енергії;
- втома;
- втрата ваги;
- надмірна спрага;
- часте сечовипускання;
- сухість у роті;
- свербіж шкіри;
- нечіткий зір.

Симптомами довгострокового поганого контролю рівня глюкози в крові або не діагностовано діабету є часте сечовипускання (надлишок цукру проходить через нирки, що створює більше сечі для його виведення з організму), підвищена спрага та ненавмисна втрата ваги, оскільки глюкоза з їжі не всмоктується. Діабет 2 типу діагностується, якщо показник рівня глюкози в крові більше ніж 7,1 ммоль/л після голодування принаймні 8 годин.

Таким чином, діагностика діабету є складним та багато процесним процесом, який включає в себе використання різних методів тестування та дослідження показників організму[3].

Оглянувши різноманітні методи діагностики діабету, можна зробити висновок, що для успішної діагностики необхідна комплексна оцінка показників організму, що дозволить вчасно виявити та уникнути розвитку хронічних захворювань, пов'язаних з цією хворобою.

Є два типи цукрового діабету. При будь-якому з них суть захворювання зводиться до підвищення концентрації глюкози (цукру) в крові [2].

На рисунку 1.6 зображені дані ознак і симптомів діабету:

Рисунок 1.6 – Ознаки та симптоми, лікування, запобігання діабету

Цукровий діабет 1 типу пов’язаний з дефіцитом інсуліну: його виробляється мало або зовсім немає. Зустрічається в 10-15% випадків. Підшлункова не справляється зі своїми функціями – кількість синтезованого гормону не переробляє всього обсягу глюкози, і рівень цукру в крові підвищується. При цьому типі діабету інсулін терапія потрібна завжди[4].

Цукровий діабет 2 типу: інсуліну виробляється достатня кількість, буває навіть більше норми. Але гормон виявляється практично не потрібен, тому що тканини організму втрачають до нього чутливість.

У цьому випадку інсулін використовується не часто – тільки при важкому перебігу, коли інші ліки не допомагають.

Лікарі виділяють також так званий “прихований діабет”, який можна виявити лише за допомогою спеціального аналізу, і потенційний, коли до нього є схильність. Подібне трапляється при ожирінні, несприятливій спадковості і при надлишковій масі тіла при народженні (від 4 кг і вище) [2].

На рисунку 1.7 зображені основні симптоми діабету:

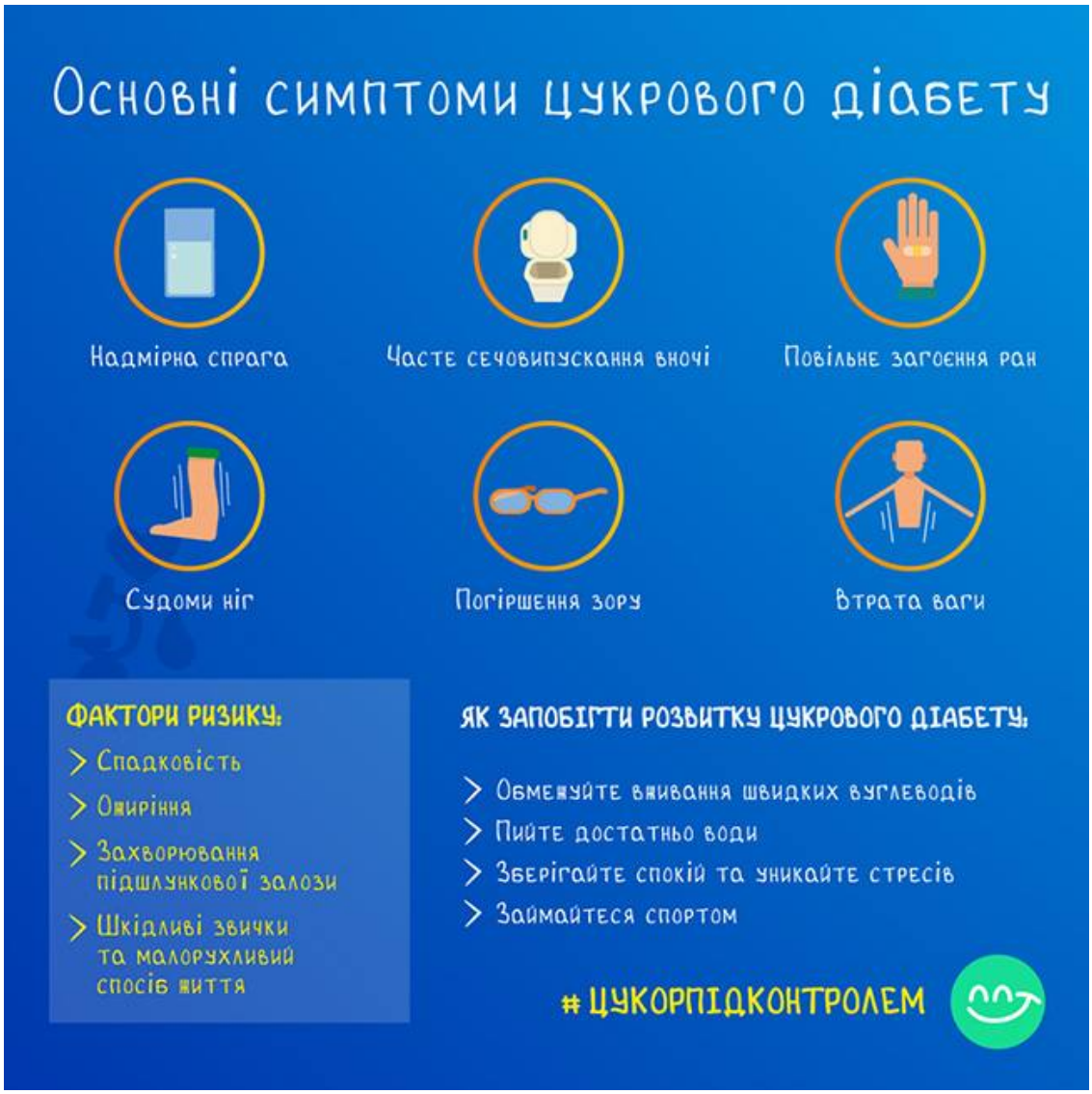

Рисунок 1.7 – Основні симптоми цукрового діабету

Основні симптоми діабету:

– посилене виділення сечі, яке викликається підвищенням її осмотичного тиску через наявність в сечі розчиненої глюкози (глюкози в сечі людини при відсутності патологій бути не може). Виявляється рясним прискореним сечовипусканням в денний, а також у нічний час.

– невмолима постійна спрага, зумовлена істотними втратами з сечею води, а також збільшенням осмотичного тиску крові.

– невгамовний постійний голод. Цей симптом викликається супроводжуючим діабет порушенням обміну речовин, а точніше – нездатністю клітин поглинати, а також переробляти глюкозу без інсуліну.

– виражене схуднення, особливо характерне для діабету 1-го типу. Це типовий симптом, що з’являється, незважаючи на наявність у хворих підвищеного апетиту. Схуднення, а нерідко навіть виснаження хворих свідчить про підвищений катаболізм жирів, а також білків через виключення глюкози з енергетичного обміну клітин хворого.

Відомо, що найбільш поширені симптоми діабету - це біль в ногах, неприродні сильні відчуття голоду та спраги, сильне сонливість та втома, зниження здатності до концентрації та сприйняття інформації, сильне моче виділяється часто, а також можуть бути присутні інші симптоми, які залежать від типу діабету[4].

Діагностика діабету передбачає проведення ряду досліджень, які допоможуть встановити наявність або відсутність цієї хвороби. Основними методами діагностики діабету є вимірювання рівня глюкози в крові, визначення глікованого гемоглобіну, проведення орального тесту на толерантність до глюкози та інші. Важливою складовою діагностики діабету є огляд пацієнта, його медична анамнез та збір скарг, що допоможуть зрозуміти наявність ризикових факторів та інших можливих причин, які впливають на розвиток цієї хвороби[5].

У практиці діагностики діабету все частіше застосовуються методи машинного навчання. Машинне навчання - це метод, що дозволяє програмам та системам навчатися на основі даних та досвіду, замість встановлення чітких правил та інструкцій вручну. Застосування методів машинного навчання до діагностики діабету дозволяє швидко та ефективно аналізувати великі масиви даних про пацієнтів та робити передбачення стану хворого на основі раніше накопичених даних.

У наступному розділі будуть розглянуті методи машинного навчання, які можуть бути використані для передбачення ризику розвитку діабету на основі аналізу клінічних даних та інших факторів[3].

1.3 Передбачення діабету та її методи

Для точної діагностики діабету та оцінки ризику його розвитку ендокринологам часто необхідно виконати деякі лабораторні тести та діагностичні процедури. Зазвичай, пацієнти здають кров для біохімічного аналізу та загального аналізу крові, а також проходять спеціальні тести, такі як тест на толерантність до глюкози або пробу Реберга. Крім того, у деяких кініках, зокрема в різних кініках, можуть бути доступні й інші діагностичні процедури, включаючи ультразвукове дослідження нирок, судин нижніх кінцівок, електрокардіографію тощо. Ці дослідження дозволяють оцінити стан органів та виявити можливі ускладнення, пов'язані з діабетом[6].

Незважаючи на це, востаннє часом машинне навчання стає все більш корисним інструментом для прогнозування діабету та оцінки ризику його розвитку. Завдяки аналізу великого обсягу даних та використанню алгоритмів машинного навчання, моделі можуть прогнозувати імовірність розвитку діабету на основі різних факторів ризику[7].

Однією з вагомих переваг машинного навчання є його здатність аналізувати велику кількість змінних одночасно та знаходити складні зв'язки між ними. Моделі можуть враховувати не тільки традиційні фактори ризику, такі як вік та сімейна історія, але й інші змінні, які можуть бути менш очевидними або складними для обробки людським розумом.

Крім того, машинне навчання може забезпечити індивідуалізовані прогнози та рекомендації. Моделі можуть враховувати унікальні характеристики кожного пацієнта та динаміку змін показників, що дозволяє підлаштовувати прогнози під конкретного індивіда[8].

Необхідно підкреслити, що використання моделей машинного навчання в діагностиці та прогнозуванні діабету не заміняє професійного медичного консультування та діагностики. Вони слугують додатковим інструментом, який може допомогти лікарям у прийнятті рішень та наданні індивідуалізованої медичної допомоги пацієнтам.

Використання різних моделей прогнозування в машинному навчанні суттєво спрощує роботу лікарів у діагностиці та прогнозуванні діабету. За допомогою алгоритмів машинного навчання, які базуються на великому обсязі даних, можна створити моделі, які здатні враховувати багато різних факторів ризику та знаходити складні зв'язки між ними[7].

Це означає, що моделі можуть аналізувати не лише традиційні фактори ризику, але й інші змінні, які можуть бути менш очевидними або складними для обробки людським розумом. Вони можуть враховувати індивідуальні характеристики пацієнтів та динаміку їх змін показників, що дозволяє отримувати індивідуалізовані прогнози та рекомендації.

Такий підхід суттєво спрощує роботу лікарів, оскільки вони можуть використовувати ці моделі як додатковий інструмент для прийняття рішень. Вони можуть надавати швидкий і точний прогноз ризику розвитку діабету, що допомагає врахувати потреби та індивідуальні особливості кожного пацієнта.

Все це дозволяє лікарям ефективніше використовувати свій час і зосередитися на основних аспектах лікування та догляду за пацієнтами, забезпечуючи їм індивідуальний та оптимальний підхід[9].

1.4 Висновки

В цьому розділі проаналізовано предметну область передбачення стану хворих на діабету. Аналіз літератури показав, що діабет є серйозним хронічним захворюванням, яке характеризується високим рівнем цукру в крові. Визначено два основних типи діабету: 1 тип і 2 тип, а також гестаційний діабет та інші рідкісні форми.

В роботі передбачення діабету пропонується на основі критеріїв, які включають етіологію, патогенез, клінічні прояви та лабораторні показники.

2 ВИБІР ОПТИМАЛЬНИХ МЕТОДІВ МАШИНОГО НАВЧАННЯ В ПЕРЕДБАЧЕННІ СТАНУ ХВОРИХ НА ДІАБЕТ

2.1 Огляд методів машинного навчання для передбачення стану хворих на діабет

В цьому розділі розглядаються такі методи машинного навчання:

- логістична регресія;
- Random Forest;
- Методом опорних векторів.

Логістична регресія є одним з найпоширеніших методів машинного навчання, використовуваних для класифікації даних. Вона є статистичним алгоритмом, який використовує логістичну функцію для передбачення ймовірності належності до одного з двох класів. У діабетології, логістична регресія може бути застосована для прогнозування ймовірності наявності чи відсутності діабету на основі набору клінічних ознак та показників.

Логістична регресія ґрунтується на моделі логістичної функції, яка використовується для моделювання ймовірності бінарної змінної. Ця функція приймає значення від 0 до 1 і дозволяє оцінити, наскільки ймовірно належить кожен зразок до певного класу[10].

В рамках логістичної регресії використовується логістична функція (також відома як сигмоїдна функція) для визначення ймовірності:

$$p(X) = 1/(1 + exp\ exp\ (-z))\ , \qquad (2.1)$$

де p(X) - ймовірність належності до класу, X - вектор вхідних ознак, а z - лінійна комбінація ваг та ознак ,

$$z = b0 + b1X1 + b2X2 + \ldots + bn * Xn\ , \qquad (2.2)$$

b0, b1, b2, ..., bn - коефіцієнти моделі, які визначаються під час тренування моделі. Вони відображають вагу, яку кожна ознака має на прогнозовану ймовірність.

Тренування та оцінка моделі (продовження): Тренування логістичної регресії полягає у знаходженні оптимальних значень коефіцієнтів моделі (b0, b1, b2, ..., bn) шляхом мінімізації функції втрат. Найпоширенішим методом є метод максимальної правдоподібності або градієнтний спуск.

Після тренування моделі можна оцінити її ефективність. Для цього використовуються метрики, такі як точність (accuracy), чутливість (sensitivity), специфічність (specificity), а також ROC-крива та площа під ROC-кривою (AUC-ROC). Ці метрики дозволяють оцінити, наскільки добре модель прогнозує класифікацію діабету[7].

Логістична регресія, як метод машинного навчання, має свої переваги та обмеження. Ось кілька плюсів та мінусів використання логістичної регресії для прогнозування діабету:

Плюси:

- - Інтерпретованість. Логістична регресія надає зрозумілі результати, що дозволяє інтерпретувати коефіцієнти моделі та робити висновки про вплив кожного ознаки на ймовірність виникнення діабету.
- - Ефективність. Логістична регресія є швидким алгоритмом, який може бути застосований до великих наборів даних.
- - Врахування нелінійності: За допомогою розширення ознак і введення взаємодії між ними, логістична регресія може моделювати нелінійні залежності між ознаками та вихідною змінною.

Мінуси:

- - Лінійна роздільність. Логістична регресія припускає лінійну роздільність між класами, що може бути обмеженням у випадку складних даних, коли залежності між ознаками та цільовою змінною є нелінійними.
- - Чутливість до викидів. Логістична регресія може бути чутливою до викидів або віддалених спостережень, що може вплинути на точність моделі.

– - Логістична регресія передбачає незалежність спостережень, що може бути порушене у випадку даних зі залежними спостереженнями.

Random Forest (RF) - це один з багатьох алгоритмів машинного навчання, що використовуються для навчання з учителем, це означає, що можна навчитися з позначених даних і робити прогнози на основі вивчених моделей. RF можна використовувати як для класифікації, так і для регресійних завдань[3]. RF базується на деревах рішень. У машинному навчанні дерева рішень є методом створення моделей прогнозування. Їх називають деревами прийняття рішень, оскільки передбачення слідує за кількома гілками рішення “якщо… тоді…”, поділеним на гілки дерева. Якщо уявити собі, що ми починаємо з вибірки, для якої ми хочемо передбачити клас, ми почнемо внизу дерева і рухаємося вгору по стовбуру, поки не прийдемо до першого відділення. Цей поділ можна розглядати як функцію в машинному навчанні, скажімо, нехай це буде «вік»; тепер ми приймемо рішення про те, яка гілка наслідуватиме: «якщо наш зразок має вік більше 30, продовжуйте ліворуч, а далі продовжуйте праворуч». Це ми будемо робити, поки не прийдемо до наступної гілки і не повторимо той самий процес прийняття рішень, поки не буде більше гілок перед нами. Ця кінцева точка називається листом, а в деревах рішень - кінцевий результат: прогнозований клас або значення. У кожній гілці знайдено порогові значення, які найкраще розділяють дані (що залишилися). Самі дерева рішень дуже легко уявити і зрозуміти, оскільки вони слідують методу прийняття рішень, який дуже схожий на те, як приймаємо рішення ми: ланцюгом простих правил[10]. Проте, вони не дуже міцні, тобто вони не добре узагальнюють нові зразки даних. Це і є точкою входу для Random Forests. RF робить прогнози шляхом комбінування результатів з багатьох окремих дерев рішень - тому ми називаємо їх лісом дерев рішень. Оскільки RF поєднує в собі декілька моделей, вона підпадає під категорію ансамблевого навчання[11].

Гіперпараметри - це аргументи, які можна встановити перед тренуванням і які визначають, як проводиться навчання. Основними гіперпараметрами в Random Forest є: - Кількість дерев рішень, які необхідно об'єднати - Максимальна

глибина дерев - Максимальна кількість ознак, що розглядаються при кожному розбитті - Чи виконується bagging/bootstrapping з або без заміни Переваги Random Forests полягають [4] у тому, що вони є відносно швидким і потужним алгоритмом навчання класифікації та регресії. Розрахунки можуть бути паралізовані і добре виконуватися при багатьох задачах, навіть при малих наборах даних, а вихідні дані повертають ймовірності прогнозування. Недоліки Random Forests полягають у тому, що вони є чорними ящиками, що означає, що ми не можемо інтерпретувати рішення, прийняті моделюю, тому що вони занадто складні. RF також дещо схильні до перенавчання, і вони, як правило, погано прогнозують недостатньо представлені класи в незбалансованих наборах даних.

Метод опорних векторів (SVM) є потужним і широко використовуваним методом машинного навчання, який знаходить широке застосування у задачах класифікації і регресії. Його основна ідея полягає в знаходженні оптимальної гіперплощини, яка найкраще розділяє два класи даних у просторі ознак. Ця гіперплощина має найбільшу відстань до найближчих точок кожного класу, які називаються опорними векторами[5].

SVM базується на теорії статистичної лінійної роздільної поверхні, де класи даних розділяються лінією або гіперплощиною. Однак, якщо дані не можуть бути лінійно розділені у вихідному просторі, SVM використовує техніку підвищення розмірності простору ознак, щоб знайти нелінійну гіперплощину, яка розділяє класи.

Для досягнення максимальної роздільної здатності SVM використовує трохи відмінний підхід порівняно з іншими класифікаторами. Він спирається на два основних елементи: функцію ядра і параметри регуляризації.

Функція ядра визначає відстань між даними у вихідному просторі ознак. Вона перетворює дані в простір вищої розмірності, де вони можуть бути лінійно розділені. Деякі з найпоширеніших функцій ядра включають лінійну, поліноміальну та радіальну базисну функцію[12].

SVM ставить задачу оптимізації з метою знаходження найкращої гіперплощини розділення. Ця оптимізаційна задача формуються як задача квадратичного програмування і розв'язується за допомогою методу підтримки векторів.

Основні переваги методу опорних векторів (SVM) для прогнозування цукрового діабету:

- Роздільна здатність: SVM забезпечує високу роздільну здатність шляхом знаходження оптимальної гіперплощини, яка найкраще розділяє класи даних. Це дозволяє враховувати складні нелінійні взаємозв'язки між ознаками і діагностичним класом.
- Tolerant to overfitting (Толерантність до перенавчання): SVM має вбудовану регуляризацію, що допомагає уникнути перенавчання моделі. Параметри регуляризації дозволяють збалансувати точність класифікації та загальний ризик моделі.

Підтримка векторів: SVM identifies support vectors, які є найближчими точками до гіперплощини розділення. Ці опорні вектори мають важливе значення при класифікації нових прикладів та утворенні рішень.

- Здатність до роботи з великими наборами даних: SVM ефективно працює з великими обсягами даних, оскільки вона залежить від підмножини опорних векторів, а не від усіх тренувальних прикладів. Це дозволяє економити обчислювальні ресурси та прискорювати процес навчання.

Незважаючи на багато переваг, метод опорних векторів (SVM) також має деякі обмеження:

- Чутливість до вибору параметрів: Ефективність SVM може сильно залежати від правильного вибору параметрів, таких як тип функції ядра та його параметри, параметр регуляризації та інші. Неправильний вибір може призвести до не до-або перенавчання моделі.
- Вимога до об'єму даних: Для ефективної роботи SVM потрібен достатньо великий обсяг даних, особливо коли використовується складна

функція ядра або велика кількість ознак. В іншому випадку, може виникнути ризик перенавчання або не до-або перенавчання моделі.

- Обробка відсутніх даних: SVM не підтримує автоматичну обробку відсутніх даних. Перед застосуванням SVM необхідно провести перед обробку даних і вирішити проблему відсутності значень шляхом заповнення чи видалення відсутніх даних.
- Швидкість навчання: Завдяки використанню опорних векторів, SVM може бути обчислювальну вимогливим, особливо для великих наборів даних. Навчання SVM може займати багато часу, особливо якщо використовується складна функція ядра.

Незважаючи на ці обмеження, метод опорних векторів є потужним і надійним інструментом для прогнозування цукрового діабету. Він може допомогти виявити складні взаємозв'язки між ознаками та діагностичним класом, а також забезпечити високу роздільну здатність при класифікації нових прикладів[12].

Метод k-найближчих сусідів (KNN) є одним із найпоширеніших методів машинного навчання, який знайшов широке застосування у багатьох областях, зокрема в прогнозуванні діабету. Цей метод базується на принципі "схожості притягується до схожого", де класифікація нових зразків або прогнозування їх значень здійснюється на основі їх близькості до вже відомих зразків з навчального набору даних.

Основна ідея методу KNN полягає в тому, що зразки з однаковими ознаками та близькими значеннями відстаней мають високу ймовірність належати до одного класу або мати схожі значення цільової змінної. Це дає можливість здійснювати прогнози на основі "досвіду" моделі, отриманого з навчального набору даних[10].

Використання методу KNN для прогнозування діабету полягає у використанні навчального набору даних, що складається з клінічних даних пацієнтів, таких як вік, стать, рівень глюкози в крові, артеріальний тиск тощо, та відповідних показників наявності або відсутності діабету. На основі цього

навчального набору, модель KNN навчається визначати залежність між ознаками пацієнтів та наявністю діабету.

Обираються k найближчих сусідів, які мають найменші відстані до нового пацієнта. Для цього можна використовувати Евклідову відстань або інші метри відстані, залежно від конкретного випадку.

Після вибору k найближчих сусідів проводиться подальший аналіз їх класифікацій або значень цільової змінної. Зазвичай використовується простий підхід більшості, де клас або значення, яке найчастіше зустрічається серед k найближчих сусідів, вважається прогнозом для нового пацієнта[12].

Одним з переваг методу KNN є його простота і інтуїтивність. Він не вимагає складних припущень про розподіл даних або функціональні залежності. Крім того, він добре пристосовується до змінних навколишніх умов і дозволяє враховувати нову інформацію без необхідності повного перетренування моделі.

Однак, метод KNN також має свої обмеження. Він може бути обчислювальну вимогливим, особливо при великій кількості навчальних зразків. Крім того, він чутливий до шуму та викидів у навчальних даних, оскільки може враховувати неправильні або нетипові зразки у класифікації або прогнозуванні.

Продовжуючи, метод KNN також може бути використаний для регресійних задач, де необхідно прогнозувати числові значення цільової змінної. У цьому випадку, замість використання більшості для визначення прогнозу, можна використовувати середнє або медіану значень цільових змінних у k найближчих сусідів[11].

Для успішного використання методу KNN важливо враховувати деякі фактори. Спочатку необхідно вибрати оптимальне значення параметра k. Велике значення k може зробити модель менш чутливою до локальних властивостей даних, тоді як мале значення k може призвести до перенавчання. Вибір оптимального значення k може вимагати проведення перехресної перевірки або використання інших методів оцінки моделі.

Також важливо правильно обробляти та нормалізувати дані перед застосуванням методу KNN. Невірно вибрані або неправильно оброблені ознаки

можуть вплинути на точність класифікації або прогнозування. Нормалізація даних може забезпечити рівні ваги для всіх ознак і запобігти домінуванню певних ознак у визначенні відстаней[13].

Метод KNN також може бути ефективним при роботі з даними великого обсягу та високої розмірності. Його простота та швидкість роботи дозволяють ефективно обробляти великі набори даних. Однак, може виникнути проблема "прокляття розмірності", коли зі збільшенням кількості ознак відстань між зразками стає менш суттєвою, що може призводити до зниження точності моделі[6].

2.2 Аналіз та порівняння методів класифікації в контексті діабету

Аналіз та порівняння методів класифікації в контексті діабету вище описаних моделей прогнозування, таких як логістична регресія, випадковий ліс і метод опорних векторів (SVM), може бути корисним для визначення найбільш ефективного підходу до прогнозування стану хворих на діабет. Нижче наведено деякі аспекти порівняння цих методів.

Простота реалізації та інтерпретації. Логістична регресія є досить простою моделлю, яку легко розуміти та інтерпретувати. Вона базується на логістичній функції, яка здатна оцінювати ймовірності належності до класів. З іншого боку, випадковий ліс і SVM є більш складними моделями, які вимагають додаткового налаштування параметрів і розуміння їх впливу на результати[13].

Обробка незбалансованих даних. Як вже зазначалося, датасет про діабет може мати нерівномірний розподіл класів, де клас хворих на діабет може бути меншим за клас здорових осіб. У таких випадках SVM може бути корисним, оскільки він може ефективно працювати з незбалансованих даними шляхом використання ваг для різних класів або методів розмиття для збалансування даних.

Роздільна здатність. У порівнянні з логістичною регресією, SVM може мати кращу здатність розділяти класи з складною границею при належному

налаштуванні ядра. Випадковий ліс також може виявити високу роздільну здатність, оскільки він комбінує багато дерев рішень. Однак, обидва методи можуть стикатися з проблемою перенавчання.

Робота з великими обсягами даних. У випадку, коли датасет має великі обсяги даних, випадковий ліс може бути перевагою. Він може ефективно обробляти великі набори даних та працювати з великою кількістю ознак. SVM також може впоратися з великими обсягами даних, але він може бути більш обчислювальну вимогливим у порівнянні з випадковим лісом[11].

Інтерпретація результатів. Логістична регресія надає зручну інтерпретацію результатів, оскільки можна оцінити вплив кожної ознаки на ймовірність належності до класу. Це може бути корисно при вивченні факторів, що впливають на розвиток діабету. У випадку випадкового лісу та SVM, інтерпретація результатів може бути складнішою, оскільки ці методи не надають прямої інформації про вплив окремих ознак.

Загалом, кожен з методів - логістична регресія, випадковий ліс і метод опорних векторів (SVM) - має свої переваги і недоліки при прогнозуванні стану хворих на діабет. Вибір найкращого методу залежить від конкретних вимог дослідження, розміру набору даних, роздільної здатності, обробки незбалансованих даних та інтерпретації результатів через це ми вибираємо метод опорних векторів (SVM) адже ми бачимо всі його переваги над іншими методами прогнозування[12].

2.3 Використання методів регресії для передбачення стану хворих на діабет

Метод опорних векторів (SVM) є потужним і широко використовуваним методом регресії для передбачення стану хворих на діабет. В даному контексті, використання SVM дозволяє побудувати модель, яка здатна розділяти хворих на діабет та здорових осіб на основі набору ознак зі зазначеного датасету з Kaggle.

SVM шукає оптимальну гіперплощину в просторі ознак, яка найкраще розділяє два класи - хворих на діабет і здорових осіб. Ця гіперплощина відокремлює класи з максимальним зазначенням, забезпечуючи оптимальну границю між ними[13].

Для використання методу опорних векторів для прогнозування діабету, спочатку необхідно підготувати дані з датасету. Це включає в себе очищення та нормалізацію даних, а також розбиття на тренувальний та тестовий набори.

Після підготовки даних, модель SVM будує гіперплощину, яка найкраще розділяє дані на основі вказаних ознак. Цей процес залежить від вибору ядра, яке визначає спосіб перетворення простору ознак для досягнення лінійної або нелінійної роздільної границі.

Однією з переваг SVM є його здатність працювати з даними, які мають складні роздільні границі. Враховуючи те, що стан хвороби діабету може бути залежний від багатьох ознак, таких як рівень цукру в крові, артеріальний тиск, вік тощо, SVM може здійснювати точні передбачення, враховуючи взаємодію цих ознак. Крім того, SVM виявляється ефективним в роботі з великими обсягами даних, що можуть містити багато спостережень і ознак. Це дає змогу побудувати модель, яка здатна здійснювати передбачення на основі широкого спектру інформації[9].

Крім того, метод опорних векторів має можливість урахувати важливість ознак. Він може визначити, які ознаки мають найбільший вплив на передбачення, тобто вказати, які ознаки найбільше корелюють з ризиком розвитку діабету. Це дозволяє лікарям і дослідникам зосередитися на найбільш значущих ознаках та збільшити ефективність діагностики та прогнозування.

Крім того, SVM є стійким до перенавчання, що є важливою властивістю для моделей передбачення в медичному контексті. Це означає, що він може ефективно управляти неправильно розподіленими даними та попереджати перенавчання, що може спотворити результати прогнозування[14].

У випадку датасету з «Pima Indians Diabetes Database» , використання методу опорних векторів може бути виправданим через наявність різних ознак,

які можуть бути корисними для передбачення діабету. За допомогою SVM, модель може адаптуватись до складної роздільної границі між класами та враховувати взаємодію між ознаками, що дозволяє досягти більш точних прогнозів стану хворих на діабет.

Крім того, SVM має ряд гіперпараметрів, які можна налаштувати для покращення результатів моделі. Це включає вибір ядра, регуляризаційних параметрів і т. д. Дослідникам надається можливість експериментувати з різними налаштуваннями моделі SVM для досягнення оптимальних результатів прогнозування діабету[13].

Одним з важливих аспектів SVM є вибір ядра. Ядро визначає спосіб перетворення простору ознак для досягнення лінійної або нелінійної роздільної границі. Для датасету з діабетом, можуть бути використані різні типи ядр, такі як лінійне, поліноміальне або радіальне базисне функціональне ядро (RBF). Ефективний вибір ядра залежить від характеру даних та їх розподілу, тому варто спробувати різні ядра та порівняти їх результати. Також, регуляризаційні параметри SVM, такі як параметр C та параметр гамма для RBF ядра, можуть бути налаштовані для досягнення кращої ефективності моделі. Параметр C контролює компроміс між підгонкою даних та загальним коефіцієнтом помилки. Більші значення C дають більш жорстку модель, яка може призвести до перенавчання, тоді як менші значення C можуть призвести до необчислення. Параметр гамма впливає на радіус впливу окремих тренувальних зразків і може вплинути на гладкість роздільної границі[5].

При використанні методу опорних векторів для передбачення стану хворих на діабет, також важливо враховувати інші етапи машинного навчання, такі як відбір ознак, нормалізація даних, розбиття на тренувальну та тестову вибірки та оцінку результатів моделі. Ці кроки допоможуть покращити якість прогнозування та уникнути перенавчання[10].

Загалом, метод опорних векторів є потужним і гнучким методом регресії, який можна успішно використовувати для передбачення стану хворих на діабет

на основі датасету з «Pima Indians Diabetes Database» , є метод опорних векторів (SVM). Застосування SVM для цієї задачі може мати декілька переваг:

А. Хороша роздільна здатність: SVM використовує гіперплощину для розділення двох класів (хворих на діабет та здорових осіб) з максимальним зазначенням. Він шукає оптимальну границю, яка максимально відділяє ці класи один від одного. Це дає можливість точніше передбачати стан хворих на діабет на основі набору ознак.

Б. Обробка нелінійних залежностей : За допомогою SVM можна використовувати різні ядра, такі як поліноміальні або RBF, що дозволяє моделі адаптуватись до нелінійних залежностей між ознаками та цільовою змінною. Це важливо для прогнозування діабету, оскільки залежність від різних ознак може мати нелінійний характер.

В. Стійкість до перенавчання: SVM має вбудовані механізми регуляризації, які допомагають уникнути перенавчання моделі. Це дозволяє узагальнити прогнози на нових даних та покращує надійність моделі.

Г. Ефективність у роботі з великими обсягами даних: SVM добре справляється з великими обсягами даних та високо вимірними просторами ознак. Це особливо важливо для аналізу діабету, оскільки можуть бути враховані багато фізіологічних, біохімічних та клінічних ознак.

Незважаючи на переваги методу опорних векторів для передбачення діабету, також слід враховувати деякі обмеження і виклики.

А. Вибір оптимальних гіперпараметрів: SVM має кілька гіперпараметрів, таких як параметр C та параметр ядра, які потребують налаштування. Важливо провести додатковий аналіз та оптимізацію цих параметрів, щоб досягти найкращої ефективності моделі. Це може вимагати витрат часу та обчислювальних ресурсів.

Обробка даних з відсутніми значеннями: Датасети, зокрема медичні, часто мають пропущені або неповні дані. Використання SVM потребує перед обробкою таких даних їхнього вирішення, наприклад, шляхом заповнення

пропущених значень або видалення відповідних спостережень. Додаткові кроки потрібні для забезпечення якості даних перед використанням SVM.

Інтерпретованість результатів: SVM може забезпечувати високу точність передбачення, але інтерпретація результатів може бути складною. Гіперплощина, яка відокремлює класи, може бути складною для визначення та пояснення. Це може бути важливим фактором, особливо для медичних досліджень, де розуміння факторів, що впливають на передбачення, може мати велике значення.

Г.Збалансованість датасету: Якщо датасет має незбалансовану розподіл класів (наприклад, значно більше здорових осіб, ніж хворих на діабет), SVM може бути схильним до зміщення в бік більш представленого класу. В таких випадках, важливо використовувати стратегії вагового коефіцієнта або використовує методів вирівнювання класів, наприклад, ваговий коефіцієнт класу або випадкове вибіркове дублювання менш представленого класу, щоб забезпечити більш об'єктивне навчання моделі.

Е. Обчислювальні вимоги: У деяких випадках, особливо при роботі з великими обсягами даних або складними ядрами, SVM може бути вимогливим до обчислювальних ресурсів. Це може вплинути на час навчання та передбачення моделі. При використанні SVM варто враховувати обчислювальні обмеження та розглянути можливості оптимізації алгоритму для покращення швидкодії.

Варто також зазначити, що метод опорних векторів не є єдиним методом регресії, який може бути використаний для передбачення стану хворих на діабет. Існують інші популярні методи, такі як лінійна регресія, дерева рішень, випадковий ліс і градієнтний бустинг, які також можуть бути ефективними в даному контексті. Вибір методу залежить від конкретних потреб і характеристик даних, але в нашому випадку метод опорних векторів підходить найкраще[11].

2.4 Висновки

У межах даної роботи було проведено дослідження з метою розроблення моделей для передбачення стану хворих на діабет. Для цього було проаналізовано ефективність різних методів класифікації, зокрема моделі K-найближчих сусідів (KNN), Random Forest (випадковий ліс) та методу опорних векторів (SVM). Модель KNN є простою та інтуїтивно зрозумілою, базується на пошуку найближчих прикладів у просторі ознак і підходить для роботи з невеликими наборами даних. Проте її продуктивність може знижуватися на великих обсягах даних через високу обчислювальну складність. Модель Random Forest показала високу ефективність і стійкість до шуму в даних завдяки ансамблевому підходу, що базується на побудові множини дерев рішень. Цей метод є надійним і здатним забезпечувати високу точність навіть на великих наборах даних, додатково надаючи можливість оцінювати важливість ознак. Метод опорних векторів (SVM), який спрямований на максимізацію граничного розділу між класами, є особливо ефективним для задач із нелінійним розподілом даних завдяки використанню ядрових функцій, хоча він може бути обчислювально затратним для великих наборів даних. На основі проведеного аналізу було визначено, що моделі Random Forest і SVM є найперспективнішими для подальшого моделювання. Random Forest забезпечує стабільну точність і ефективність при роботі з великими обсягами даних, тоді як SVM є оптимальним вибором у випадках складного розподілу даних, що потребує побудови нелінійних меж між класами. Модель KNN рекомендовано використовувати для базового порівняння результатів, проте її продуктивність поступається більш потужним методам. Таким чином, для подальшого передбачення стану хворих на діабет основну увагу буде зосереджено на використанні моделей Random Forest і SVM.

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ ПЕРЕДБАЧЕННЯ СТАНУ ХВОРИХ НА ДІАБЕТ

3.1 Розроблення алгоритму інформаційної технології

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.

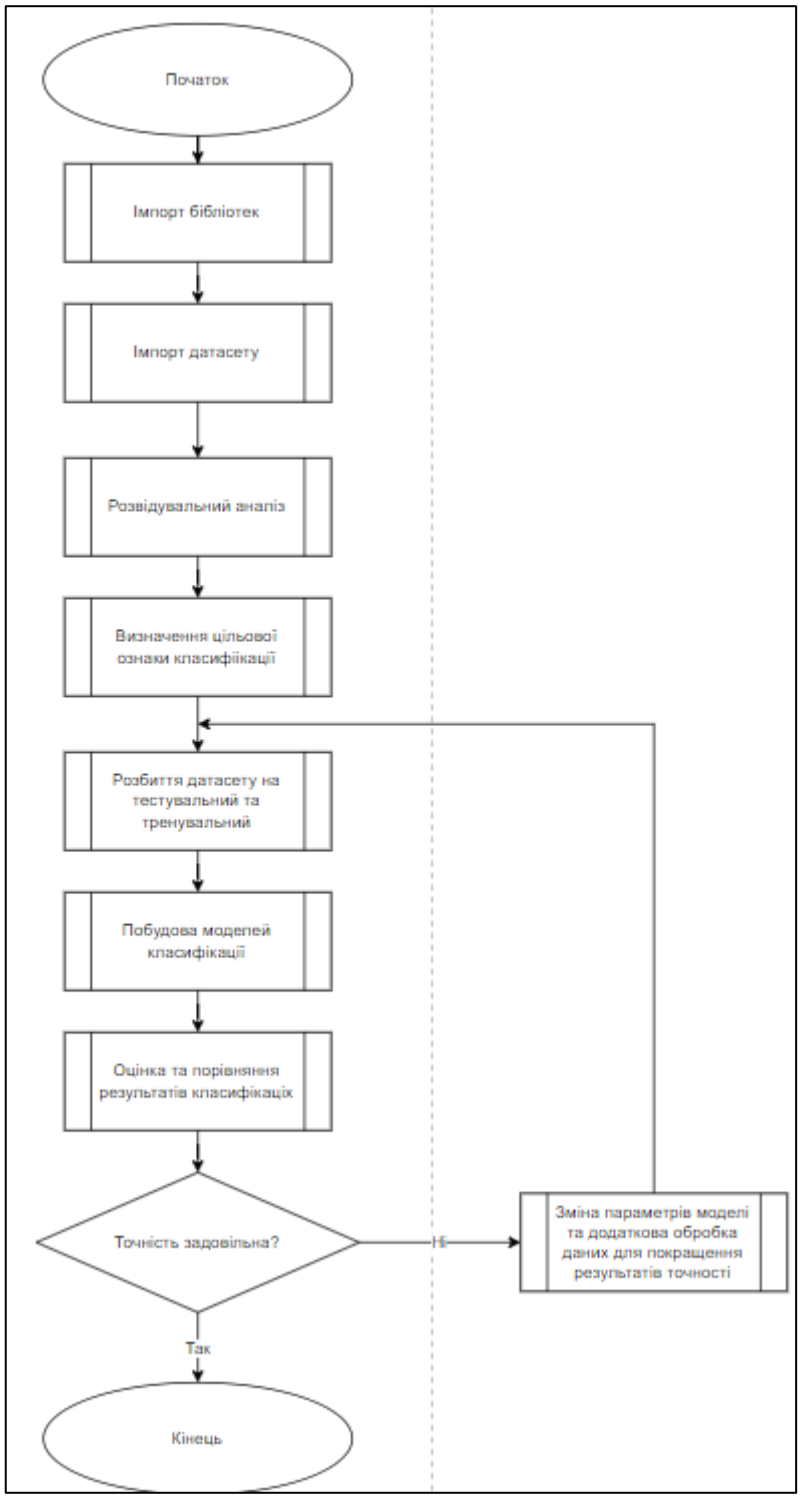

Рисунок 3.1 – Схема роботи алгоритму

3.2 Аналіз набору даних про діабет

Дослідження діабету є однією з найбільш актуальних галузей медичних досліджень, оскільки ця хронічна хвороба становить серйозну загрозу для здоров'я людей по всьому світу. Одним із способів вивчення діабету та розробки

стратегій його раннього виявлення є аналіз великих масивів даних[14]. У цьому огляді ми розглянемо базу даних "Pima Indians Diabetes Database", доступну на Kaggle, яка забезпечує цінний набір даних для дослідження цієї хвороби.

На рисунку 3.1 зображено датасет який ми використовуємо:

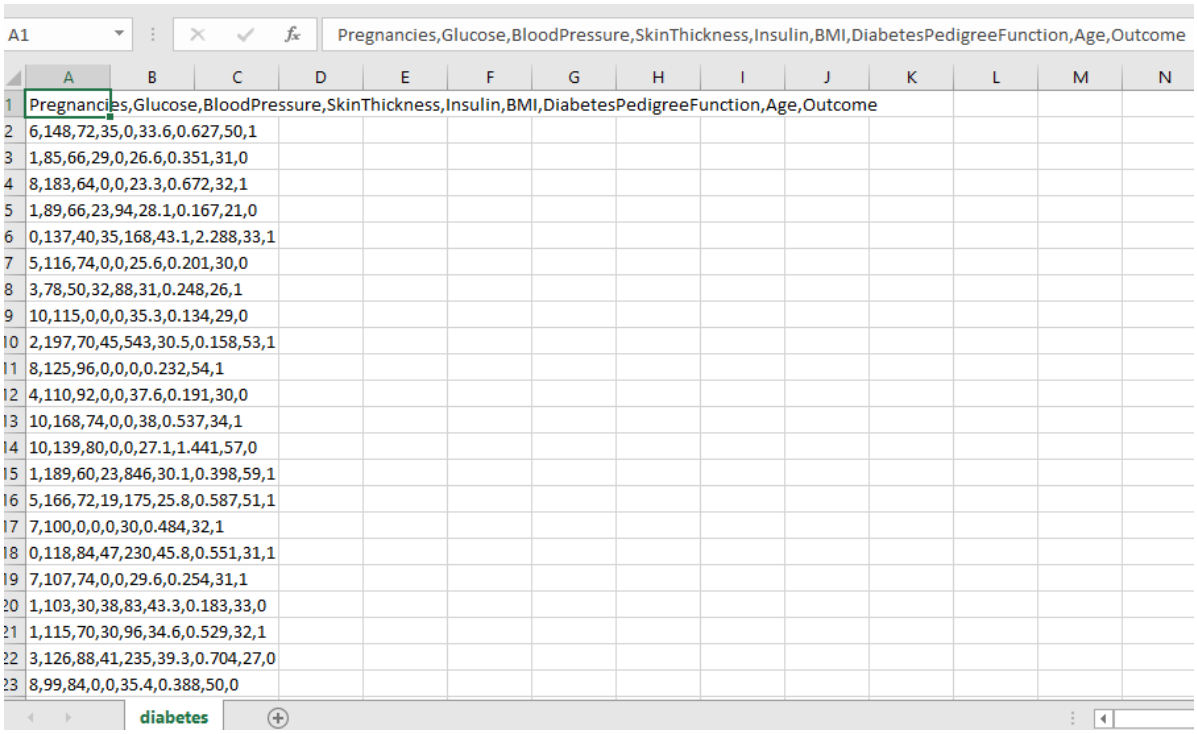

Рисунок 3.2 Датасет "Pima Indians Diabetes Database"

Датасет "Pima Indians Diabetes Database" створена зі збору медичних даних про пацієнтів з племені Піма Індіанців, які проживають в Північній Америці. Цей набір даних включає інформацію про різні клінічні атрибути, такі як глюкоза в крові, артеріальний тиск, ІМТ (індекс маси тіла) тощо, а також наявність діабету у пацієнтів.

Метою створення цієї бази даних було дослідження залежності між клінічними атрибутами та наявністю діабету. Вона може служити основою для розробки прогностичних моделей та стратегій раннього виявлення ризику розвитку діабету.

В датасеті "Pima Indians Diabetes Database" містить такі атрибути:

- Вагітність: кількість вагітностей в жінок.

- Глюкоза: концентрація глюкози в плазмі через 2 години після тесту навантаження глюкозою.
- Артеріальний тиск: діастолічний тиск (в мм рт.ст.).
- Товщина шкіри складки трицепса: товщина шкіри на трицепсі (в мм).
- Інсулін: 2-годинний рівень інсуліну в плазмі (в мкЕд/мл).
- Індекс маси тіла (ІМТ): вага в кілограмах поділена на квадрат висоти у метрах.
- Сімейна анамнез функція: функціональна оцінка сімейного анамнезу за допомогою логарифмічного масштабування.
- Вік: вік пацієнта в роках.

Цільовий показник: присутність діабету (1 - присутній, 0 - відсутній).

Перед прогнозуванням діабету, дані з бази даних можуть потребувати попередньої обробки, такої як обробка відсутніх значень, нормалізація числових атрибутів, кодування категоріальних атрибутів тощо. Це допоможе підготувати дані для подальшого використання в моделях прогнозування.

Після попередньої обробки даних можна використовувати різні моделі прогнозування, такі як логістична регресія, дерева рішень, випадковий ліс або нейронні мережі. Вибір моделі залежить від конкретної задачі, обсягу даних та наявних ресурсів[15].

Після побудови моделі необхідно оцінити її ефективність за допомогою метрик, таких як точність, чутливість, специфічність тощо. Валідація моделі може включати в себе використання перехресної перевірки, розділення навчальної та тестової вибірки, а також використання інших методів для перевірки стабільності та надійності моделі.

Ця база даних містить клінічні дані про жінок племені Піма індіанців, які проживають в Північній Америці. Головною метою дослідження було прогнозування, чи мають жінки діабет, на підставі різних клінічних показників. Ці дані часто використовуються для розробки моделей прогнозування діабету за допомогою машинного навчання[16].

Використання цієї бази даних може допомогти дослідникам і практикуючим лікарям у зрозумінні ризиків, пов'язаних з діабетом, та розробці нових підходів до діагностики та лікування цієї хвороби.

3.3 Розвідувальний аналіз

Розвідувальний аналіз (Exploratory Data Analysis, EDA) є важливою складовою процесу машинного навчання та аналізу даних. В рамках кваліфікаційної роботи, розділ, присвячений розвідувальному аналізу, має на меті вивчення та розуміння набору даних, що дозволяє виявити корисну інформацію, визначити потенційні проблеми та покращити якість моделі або дослідження.

Головною метою розвідувального аналізу є виявлення закономірностей та структури в даних, визначення залежне між ознаками, виявлення аномалій та викидів, а також визначення важливих ознак, які можуть впливати на модель машинного навчання або результати дослідження. Це перший крок у розумінні даних та визначенні наступних кроків у процесі аналізу.

Основні компоненти розвідувального аналізу включають вивчення загальної структури та розподілу даних, виявлення відсутніх значень та обробку пропущених даних, визначення статистичних показників, візуалізацію даних за допомогою графіків та діаграм, а також виявлення взаємозв'язків та кореляцій між ознаками.

Одним із ключових інструментів розвідувального аналізу є кореляційна матриця. Кореляційна матриця дозволяє визначити ступінь взаємозв'язку між парами ознак у наборі даних. Вона надає значення кореляції між кожною парою ознак, які можуть бути позитивними (пряма кореляція), негативними (зворотна кореляція) або нульовими (відсутність кореляції). Ця інформація дозволяє встановити, які ознаки можуть бути взаємозалежними та потенційно впливати на результати моделювання або дослідження.

Застосування кореляційної матриці допомагає зрозуміти структуру даних та виявити потенційні залежності між ознаками. Це дозволяє налаштувати модель машинного навчання, виключити непотрібні або взаємозалежні ознаки, а також зробити обґрунтовані висновки щодо важливих факторів, які впливають на задачу аналізу[14].

У заключенні, розвідувальний аналіз та кореляційна матриця відіграють важливу роль у розумінні даних, виявленні взаємозв'язків між ознаками та покращенні моделей машинного навчання або досліджень. Ці інструменти дозволяють виявити ключові аспекти даних, що можуть мати вирішальний вплив на подальший аналіз та прийняття рішень[15].

На рисунку 3.3 зображено кореляційна матриця розвідувального аналізу:

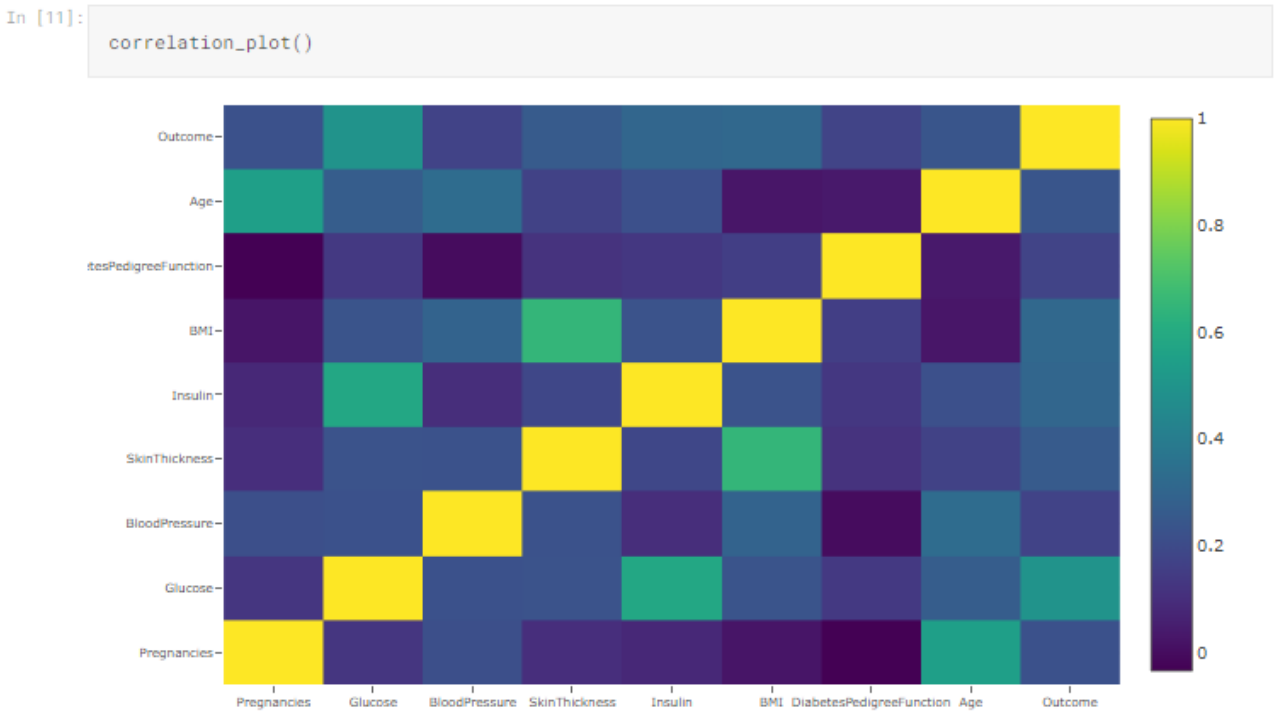

Рисунок 3.3 Кореляційна матриця розвідувального аналізу

З аналізу кореляційної матриці можна зробити декілька висновків про взаємозв'язки між ознаками:

Ознака "Glucose" має помірну позитивну кореляцію з ознакою "Outcome", що може вказувати на те, що високий рівень глюкози в крові може бути пов'язаний зі зростанням ризику виникнення діабету.

Ознаки "Pregnancies" та "Age" також мають позитивну кореляцію з "Outcome", що може свідчити про збільшення ризику виникнення діабету зі зростанням числа вагітностей та віком пацієнта.

Ознаки "BMI" (індекс маси тіла) та "SkinThickness" (товщина шкіри) також можуть бути пов'язані з ризиком діабету, оскільки вони мають позитивну кореляцію з "Outcome".

Ознака "BloodPressure" (артеріальний тиск) має слабку кореляцію з "Outcome", що може вказувати на меншу впливовість цієї ознаки на ризик діабету.

Ознака "Insulin" також може бути пов'язана з ризиком діабету, але ця залежність може бути складною, оскільки значення "Insulin" мають високу кількість пропущених значень.

Ці висновки є орієнтовними та вимагають подальшого дослідження та аналізу. Важливо також враховувати, що кореляція не означає причинно-наслідковий зв'язок і потребує додаткових перевірок та валідації.

Графік рисунка 3.3 дозволяє нам отримати важливі інсайт про розподіл даних у наборі даних, що може бути корисним для розвідувального аналізу та подальшого дослідження в рамках теми кваліфікаційної роботи.

Основні висновки, які можна зробити з цього графіка, включають:

Виявлення викидів або аномалій: Графік ящика з вусами дозволяє виявити можливі викиди в змінних. Якщо відстань між верхнім або нижнім вусом та межами ящика велика, це може свідчити про наявність викидів.

Статистичні характеристики: Графік надає інформацію про мінімальне та максимальне значення, медіану, перце тилі та між квартальний розмах кожної змінної. Ці показники допомагають зрозуміти розподіл значень та визначити потенційні аномалії або несподівані властивості даних[16].

Порівняння розподілів: Графік дозволяє порівняти розподіли різних змінних між собою. За допомогою цього графіка можна виявити відмінності у розподілі даних між категоріями, що може бути корисним для виявлення залежностей та взаємозв'язків між змінними.

У контексті теми кваліфікаційної роботи, цей графік може допомогти в аналізі медичних даних та виявленні залежностей між різними факторами та виникненням діабету. За допомогою цього графіка можна провести первинний огляд даних та зрозуміти їх розподіл, що може бути важливим для подальшого дослідження та моделювання.

На рисунку 3.4 зображено графік ящика з вусами:

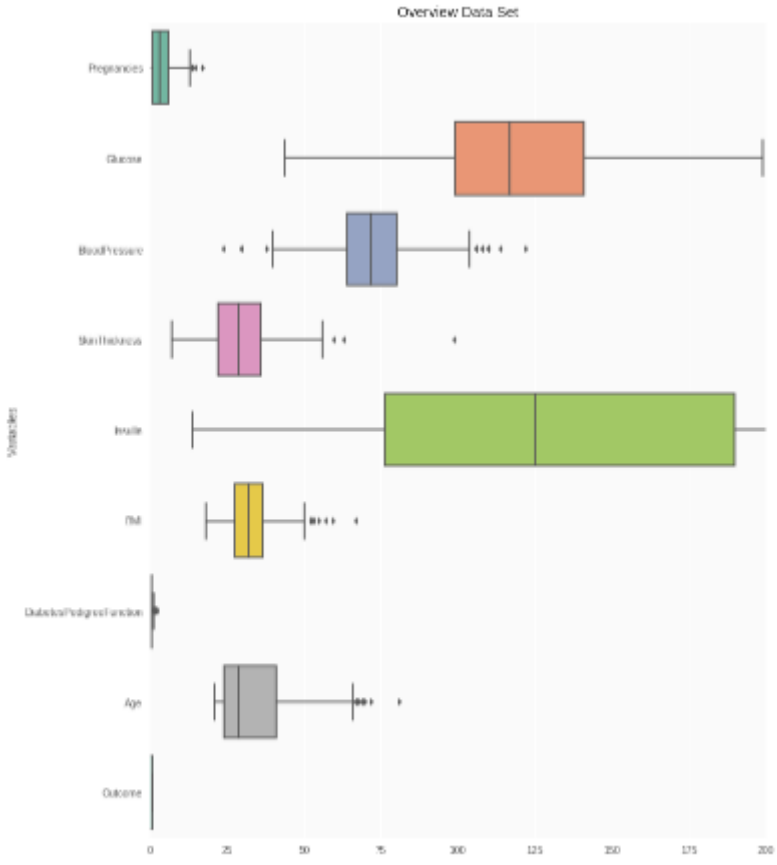

Рисунок 3.4 Графік ящика з вусами дозволяє виявити потенційні викиди або аномалії в розподілі даних.

За допомогою останнього фрагмента коду, ми можемо отримати деякі висновки про розподіл даних. Основні спостереження наступні:

Графік ящика з вусами показує розподіл змінних у наборі даних. Кожна змінна представлена у вигляді ящика, який показує мінімальне значення, перце тилі, медіану, перце тилі та максимальне значення.

За допомогою графіка ящика з вусами ми можемо виявити можливі викиди або аномалії в даних. Наприклад, якщо є значення, що виходять за межі "вусів" (whiskers) ящика, це може вказувати на наявність викидів або незвичайних значень у даних.

Графік також дозволяє порівняти розподіл змінних в залежності від цільової змінної. Наприклад, у нашому випадку, якщо порівняти розподіл змінних між недіабетичними (Outcome = 0) та діабетичними (Outcome = 1) пацієнтами, можна зробити припущення про те, які змінні можуть бути корисними для прогнозування діабету.

З попереднього аналізу пропущених значень ми бачимо, що деякі змінні, такі як Insulin, SkinThickness, BloodPressure, BMI та Glucose, мають значну кількість пропущених значень. Це може вплинути на подальший аналіз та моделювання даних, тому ми маємо враховувати ці пропуски під час обробки даних.

Графік розподілу допомагає нам зрозуміти, як значення змінної розподілені по діапазону значень. Він відображає частоту або ймовірність появи кожного значення. За допомогою цього графіка ми можемо з'ясувати, чи є розподіл нормальним, чи має він скошеність, наявність викидів або аномальних значень[15].

У випадку функції plot_distribution('Insulin', 0), графік розподілу буде показувати, як розподілені значення змінної "Insulin". Враховуючи аргумент 0, графік включатиме всі значення, включаючи нульові значення "Insulin". Це дозволяє нам отримати повну картину розподілу цієї змінної і виявити можливі викиди або аномалії.

На рисунку 3.5 зображені дані графіка розподілу значень змінної "Insulin":

```
In [14]: plot_distribution('Insulin', 0)
```

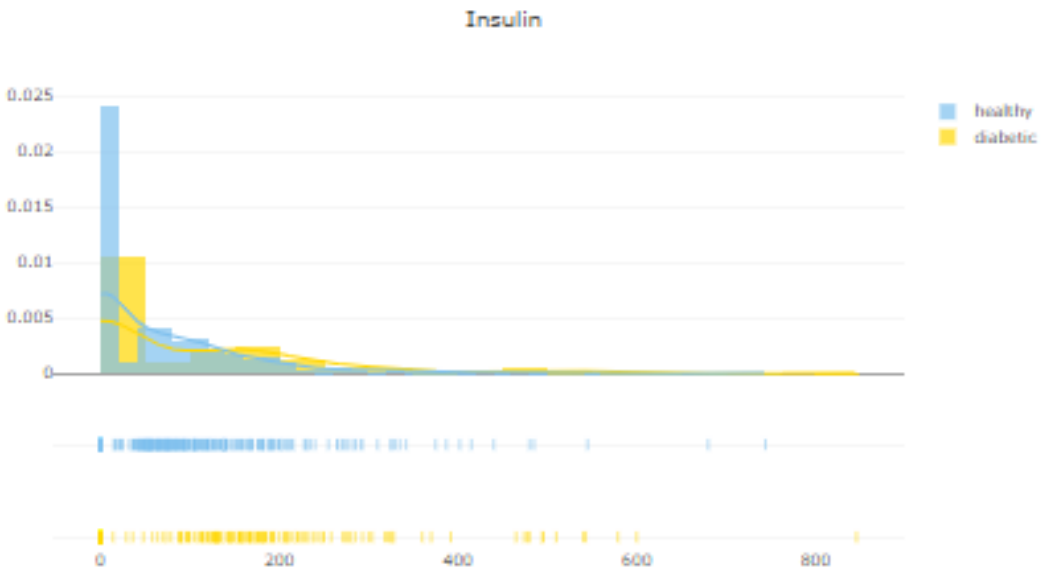

Рисунок 3.5 Графік розподілу значень змінної "Insulin"

З останнього графіка розвідувального аналізу ми можемо зробити кілька висновків:

Розподіл змінної "Insulin": Графік розподілу значень змінної "Insulin" показує, як розподілені значення цієї змінної. Ми бачимо, що розподіл має скошений характер, з більшою концентрацією значень в нижній частині графіка. Це може вказувати на те, що більшість пацієнтів мають низькі рівні інсуліну. Однак, ми також бачимо наявність декількох викидів або аномальних значень, що може бути важливою інформацією при подальшому аналізі[16].

Пропущені значення: За допомогою графіка ми також можемо визначити відсоток пропущених значень для кожної змінної. У нашому випадку, ми бачимо, що змінні "Insulin", "SkinThickness", "BloodPressure", "BMI" та "Glucose" мають різні рівні пропущених значень. Це означає, що у деяких випадках відсутня інформація про ці параметри, що може впливати на аналіз і моделювання даних.

Використані стилізації та візуальні налаштування: Код також включає встановлення стилізації графіка (використання стилю ggplot2) і налаштування розміру та кольору фону графіка. Ці налаштування допомагають зробити графік більш естетичним та привабливим для подальшого використання і презентації.

Отже, з цього розвідувального аналізу графіка ми можемо виробити припущення та гіпотези щодо розподілу змінної "Insulin" та виявлення пропущених значень, що можуть впливати на подальший аналіз даних та вибір підходів для їх обробки[17].

Графік розподілу "Glucose" для отримання візуального представлення про розподіл значень цієї змінної в наборі даних. Цей графік дозволяє нам оцінити тип розподілу: з графіка ми можемо визначити, чи має розподіл значень "Glucose" нормальну форму або чи має він якісь відхилення, такі як скошеність або мультимодальність. Це допомагає нам зрозуміти, як розподілені дані і які статистичні методи можуть бути застосовані для їх аналізу. Виявити викиди або аномальні значення: Графік може виявити наявність викидів або аномальних значень, які можуть бути помилковими або несподіваними в контексті досліджуваної змінної. Виявлення цих аномалій може бути важливим при прийнятті рішень про обробку даних і виключення впливу неправдивих або незвичайних спостережень. Порівняти з розподілом цільової змінної: Графік дозволяє порівняти розподіл значень "Glucose" з розподілом цільової змінної (у даному випадку "Outcome"), що вказує на наявність зв'язку між ними. Наприклад, можна оцінити, чи існує різниця у розподілі рівня глюкози між діабетиками та недіабетиками[17].

Отже, графік розподілу "Glucose" допомагає нам отримати важливу інформацію про змінну "Glucose" і побачити її статистичні характеристики, а також виявити потенційні аномалії або зв'язки з іншими змінними у наборі даних.

На рисунку 3.6 зображені дані графіка розподілу значень змінної "Glucose":

```
In [17]: plot_distribution('Glucose', 0)
```

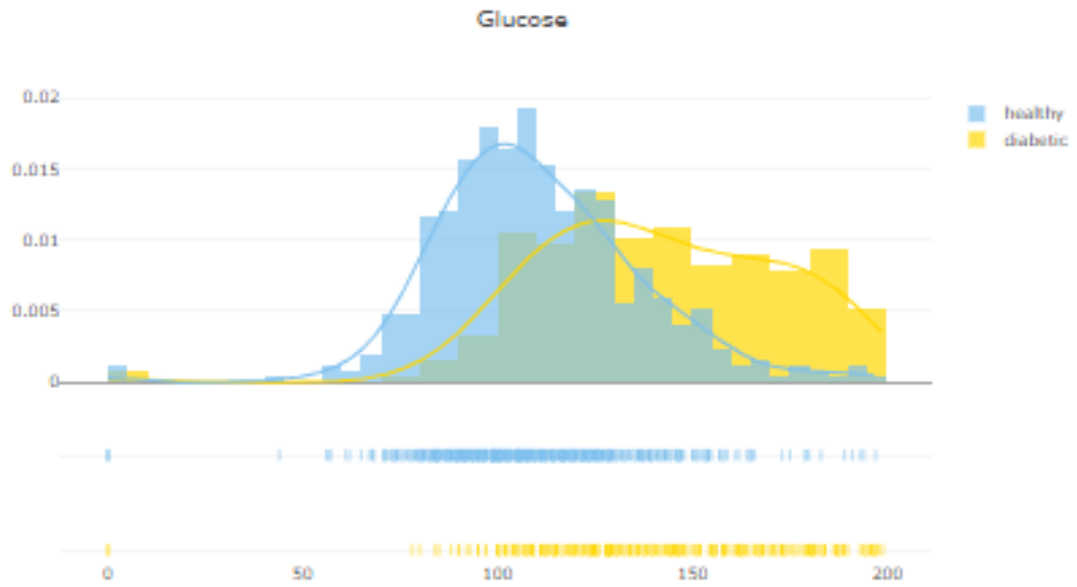

Рисунок 3.6 Відображає розподіл значень змінної "Glucose"

Розглядаючи графік розподілу "Glucose", ми можемо зробити наступні висновки:

Розподіл значень "Glucose" схожий на нормальний: Графік має симетричну форму з піком навколо середнього значення. Це може вказувати на те, що більшість значень "Glucose" зосереджені навколо певної центральної точки.

Наявність деяких викидів або аномальних значень: Ми можемо помітити декілька точок, що відхиляються від основного кластера значень. Ці значення можуть бути викидами або аномальними спостереженнями, які варто детальніше розглянути та перевірити їх достовірність[17].

Розподіл "Glucose" має широкий діапазон: Значення "Glucose" розташовані в діапазоні від приблизно 50 до 200 одиниць. Це свідчить про наявність великої різноманітності рівнів глюкози серед досліджуваних осіб.

Відсутність пропущених значень: Зауважимо, що на графіку немає пропущених значень для змінної "Glucose". Це означає, що для всіх осіб в наборі даних були виміряні рівні глюкози.

Загалом, графік розподілу "Glucose" надає нам важливі візуальні відомості про розподіл цієї змінної, допомагає виявити аномалії та з'ясувати ширину діапазону значень. Ця інформація може бути корисною для подальшого аналізу даних та встановлення зв'язків з іншими змінними у наборі даних.

Ми використовуємо графік розподілу змінної "SkinThickness" для розвідувального аналізу з метою отримання наступних висновків:

Розподіл значень: Графік дозволяє нам побачити, як розподілені значення змінної "SkinThickness". Ми можемо побачити, чи є розподіл нормальним або має якісь відхилення, такі як скошеність або викиди.

Виявлення викидів: Графік може допомогти виявити викиди або незвичайні значення в змінній "SkinThickness". Вони можуть бути помилковими даними або вказувати на особливі випадки, які варто дослідити детальніше.

Розподіл категорій: Якщо змінна "SkinThickness" є категоріальною, графік може показати розподіл категорій та їх відносну кількість. Це може допомогти виявити дисбаланс категорій або виявити домінуючі категорії.

На рисунку 3.7 зображені дані графіка розподілу значень змінної "SkinThickness":

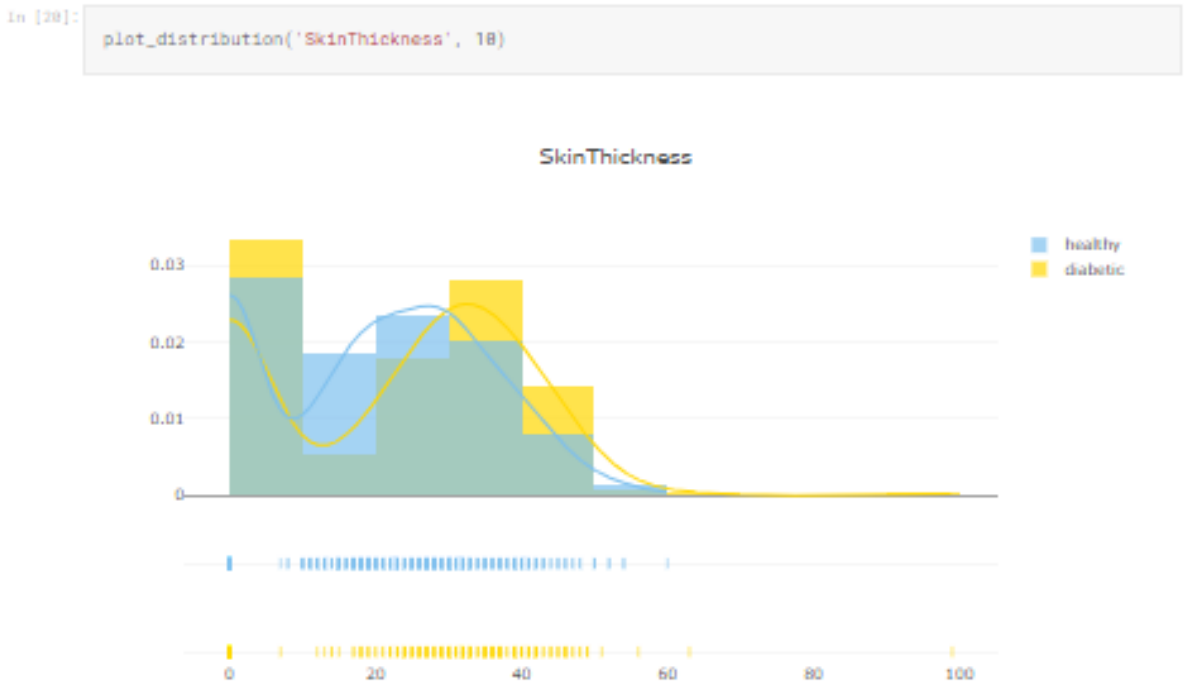

Рисунок 3.7 Бачимо як розподілені значення змінної "SkinThickness" у наборі даних

Загалом, графік розподілу "SkinThickness" надає важливу інформацію про цю змінну, яка може бути корисною при подальшому аналізі даних та прийнятті рішень.

Виведення графіку plot_feat1_feat2('Glucose','Age') дозволяє нам дослідити відношення між рівнями глюкози в крові ("Glucose") та віком пацієнтів ("Age"), і може допомогти виявити можливі взаємозв'язки або відмінності між цими змінними.

Графік може мати різні форми, залежно від типу змінних. Наприклад, якщо обидві змінні є числовими, то може бути відображено розсіювальний графік, де кожна точка представляє співвідношення значень "Glucose" і "Age" для кожного спостереження. Це дозволяє оцінити наявність кореляції або залежності між цими змінними.

На рисунку 3.8 зображені дані Відношення між рівнями глюкози в крові та віком пацієнтів:

```
In [34]: plot_feat1_feat2('Glucose','Age')
```

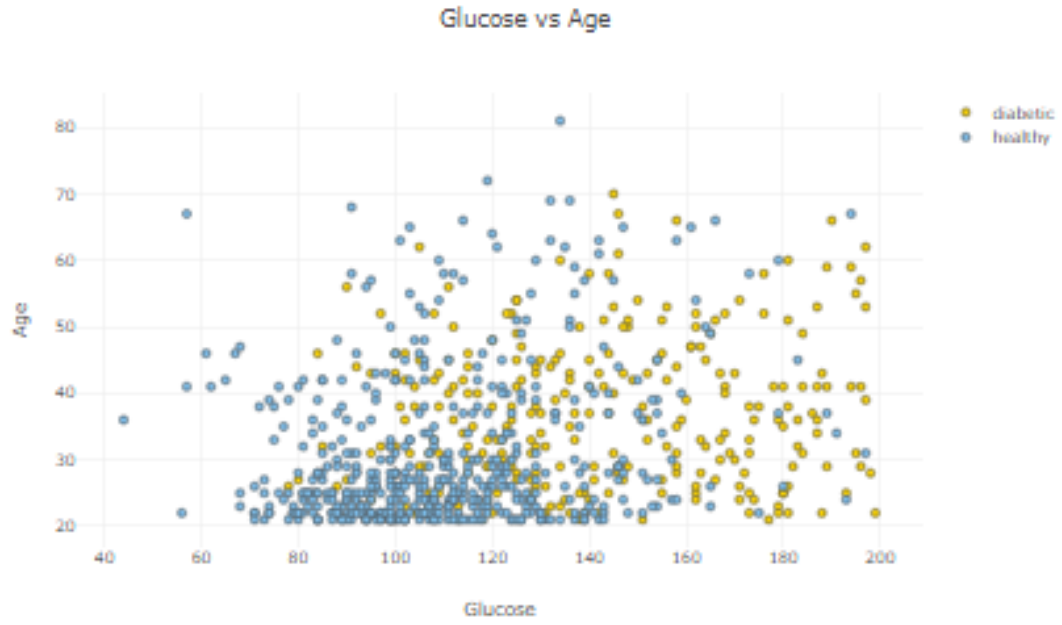

Рисунок 3.8 Відношення між рівнями глюкози в крові ("Glucose") та віком пацієнтів ("Age")

За допомогою графіка plot_feat1_feat2('Glucose','Age') можна зробити наступні висновки:

Залежність між рівнями глюкози в крові ("Glucose") та віком пацієнтів ("Age"): Графік показує, чи існує якась взаємозв'язок або закономірність між цими двома змінними. Наприклад помітно, чи змінюється рівень глюкози зі зростанням віку.

Розподіл даних: Графік показує, як розподіляються значення глюкози та віку. Наприклад, можна побачити, чи є концентрація глюкози або вік пацієнтів сконцентрованими у певних діапазонах[17].

Виявлення викидів або аномальних значень: Графік допомагає виявити викиди або аномальні значення, які можуть бути помилковими або несподіваними. Наприклад, можна знайти незвичайно високі або низькі рівні глюкози для певних вікових груп.

Загалом, графік plot_feat1_feat2('Glucose','Age') надає візуальне представлення взаємозв'язку між рівнями глюкози та віком, що дозволяє робити висновки щодо цих змінних у контексті досліджуваної проблеми, такої як діабет.

Отже розвідувальний аналіз (EDA) допоміг нам у багатьох аспектах:

Розуміння даних: EDA дозволив нам отримати загальне уявлення про наш набір даних. Ми отримали інформацію про розмір даних, типи змінних і їх розподіл, наявність пропущених значень та інше. Це допомогло нам ліпше розуміти саму структуру даних[18].

Виявлення аномалій: EDA дозволив нам виявити аномалії або незвичайні значення в наших даних, такі як пропущені значення, викиди або неправдоподібні значення. Це важливо для подальшої обробки даних та вибору відповідних методів аналізу.

Визначення залежностей та взаємозв'язків: EDA надав нам можливість вивчити взаємозв'язки між різними змінними у наших даних. Ми з'ясували, які змінні корелюють між собою, чи є існуючі залежності або закономірності. Це допомогло нам краще розуміти досліджувану проблему і виявити потенційно важливі змінні для подальшого аналізу.

Вибір відповідних методів аналізу: EDA надав нам контекст і вихідні дані для вибору відповідних методів аналізу. Ми змогли зробити припущення щодо розподілу даних, виявити кореляції та залежності, що допомогло нам обрати правильні статистичні методи та моделі для подальшого дослідження.

Отже, розвідувальний аналіз важливий етап у процесі аналізу даних, який допомагає нам зрозуміти дані, виявити аномалії, встановити залежності та визначити наступні кроки для аналізу та моделювання. Він служить фундаментом для подальшого дослідження та роботи з даними[18].

3.4 Класифікація моделлю KNN

Початок роботи з класифікацією моделі для діабету. Спочатку необхідно завантажити набір даних про діабет з джерела, наприклад, з Kaggle. У прикладі з Kaggle, набір даних можна завантажити як CSV-файл.

На рисунку 3.9 Завантаження даних для прогнозування:

```
In [5]: #Завантаження набору даних
        diabetes_data = pd.read_csv('diabetes.csv')

        #Вивести перші 5 рядків датафрейму.
        diabetes_data.head(100)
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
95	6	144	72	27	228	33.9	0.255	40	0
96	2	92	62	28	0	31.6	0.130	24	0
97	1	71	48	18	76	20.4	0.323	22	0
98	6	93	50	30	64	28.7	0.356	23	0
99	1	122	90	51	220	49.7	0.325	31	1

100 rows × 9 columns

Рисунок 3.9 Завантаження робочих даних для прогнозування

Огляд та підготовка даних: Після завантаження даних потрібно провести огляд та вивчити їх структуру та характеристики. Даний етап включає перегляд перших кількох рядків даних, перевірку наявності відсутніх значень, обробку випадків відсутніх значень, а також можливу стандартизацію або нормалізацію даних.

На рисунку 3.10 зображено структуру даних якими ми користуємось:

```
In [8]: diabetes_data.describe().T
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

Рисунок 3.10 Структура даних

Розбиття даних на тренувальний та тестовий набори: Для оцінки продуктивності моделі потрібно розбити дані на тренувальний набір і тестовий набір. Тренувальний набір буде використовуватися для навчання моделі, а тестовий набір - для оцінки її точності.

На рисунку 3.11 зображена оцінка продуктивності на тестових даних:

```
In [27]: #імпорт_тест_поїзда_розбитий
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=1/3,random_state=42, stratify=y)

In [28]: from sklearn.neighbors import KNeighborsClassifier


test_scores = []
train_scores = []

for i in range(1,15):

    knn = KNeighborsClassifier(i)
    knn.fit(X_train,y_train)

    train_scores.append(knn.score(X_train,y_train))
    test_scores.append(knn.score(X_test,y_test))

In [29]: ## оцінка, отримана в результаті тестування на тих самих точках даних, які використовувалися для навчання
max_train_score = max(train_scores)
train_scores_ind = [i for i, v in enumerate(train_scores) if v == max_train_score]
print('Max train score {} % and k = {}'.format(max_train_score*100,list(map(lambda x: x+1, train_scores_ind))))

Max train score 100.0 % and k = [1]
```

Рисунок 3.11 Оцінка продуктивності моделі на тестових даних

Вибір алгоритму класифікації: В прикладі з кодом використовується алгоритм k-найближчих сусідів (K-Nearest Neighbors, KNN) для класифікації діабету.

На рисунку 3.12 зображена класифікація діабету:

```
In [27]: #імпорт_тест_поїзда_розбитий
         from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=1/3,random_state=42, stratify=y)

In [28]: from sklearn.neighbors import KNeighborsClassifier


         test_scores = []
         train_scores = []

         for i in range(1,15):

             knn = KNeighborsClassifier(i)
             knn.fit(X_train,y_train)

             train_scores.append(knn.score(X_train,y_train))
             test_scores.append(knn.score(X_test,y_test))

In [29]: ## оцінка, отримана в результаті тестування на тих самих точках даних, які використовувалися для навчання
         max_train_score = max(train_scores)
         train_scores_ind = [i for i, v in enumerate(train_scores) if v == max_train_score]
         print('Max train score {} % and k = {}'.format(max_train_score*100,list(map(lambda x: x+1, train_scores_ind))))

         Max train score 100.0 % and k = [1]
```

Рисунок 3.12 Класифікація діабету

Після навчання моделі необхідно оцінити її продуктивність на тестовому наборі даних. В прикладі з Kaggle використовується метрика точності (accuracy), яка вимірює відсоток правильно класифікованих екземплярів.

На рисунку 3.13 зображена продуктивність моделі на тестових даних:

```
In [35]: #Створити класифікатор knn з k сусідами
         knn = KNeighborsClassifier(11)

         knn.fit(X_train,y_train)
         knn.score(X_test,y_test)

Out[35]: 0.765625
```

Рисунок 3.13 Продуктивність моделі на тестових даних

Після побудови та оцінки моделі ви можете використовувати її для класифікації нових даних. В прикладі з кодом показано приклад класифікації нового пацієнта з діабетом за допомогою навченої моделі.

На рисунку 3.14 зображені результати моделювання моделі:

```
In [45]: #import GridSearchCV
         from sklearn.model_selection import GridSearchCV
         #У випадку класифікатора типу knn параметром, що налаштовується, є n_neighbors
         param_grid = {'n_neighbors':np.arange(1,50)}
         knn = KNeighborsClassifier()
         knn_cv= GridSearchCV(knn,param_grid,cv=5)
         knn_cv.fit(X,y)

         print("Best Score:" + str(knn_cv.best_score_))
         print("Best Parameters: " + str(knn_cv.best_params_))

         Best Score:0.7721840251252015
         Best Parameters: {'n_neighbors': 25}
```

Рисунок 3.14 Результати моделювання

На рисунку 3.15 зображено найкращий із результатів прогнозування:

```
In [40]: #звіт_про_класифікацію_імпорту
         from sklearn.metrics import classification_report
         print(classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       0.80      0.85      0.83       167
                    1       0.68      0.61      0.64        89

             accuracy                           0.77       256
            macro avg       0.74      0.73      0.73       256
         weighted avg       0.76      0.77      0.76       256
```

Рисунок 3.15 Найкращий результат прогнозування

3.5.Класифікація моделлю Random Forest

Проведено дослідження з використанням алгоритму Random Forest для класифікації діабету. На рисунку 3.16 зображено візуалізацію про розподіл

кількості вагітностей в досліджуваному наборі даних. За допомогою цієї діаграми можна побачити, скільки жінок у дослідженні не мали вагітностей, скільки мали одну вагітність, дві, і так далі.

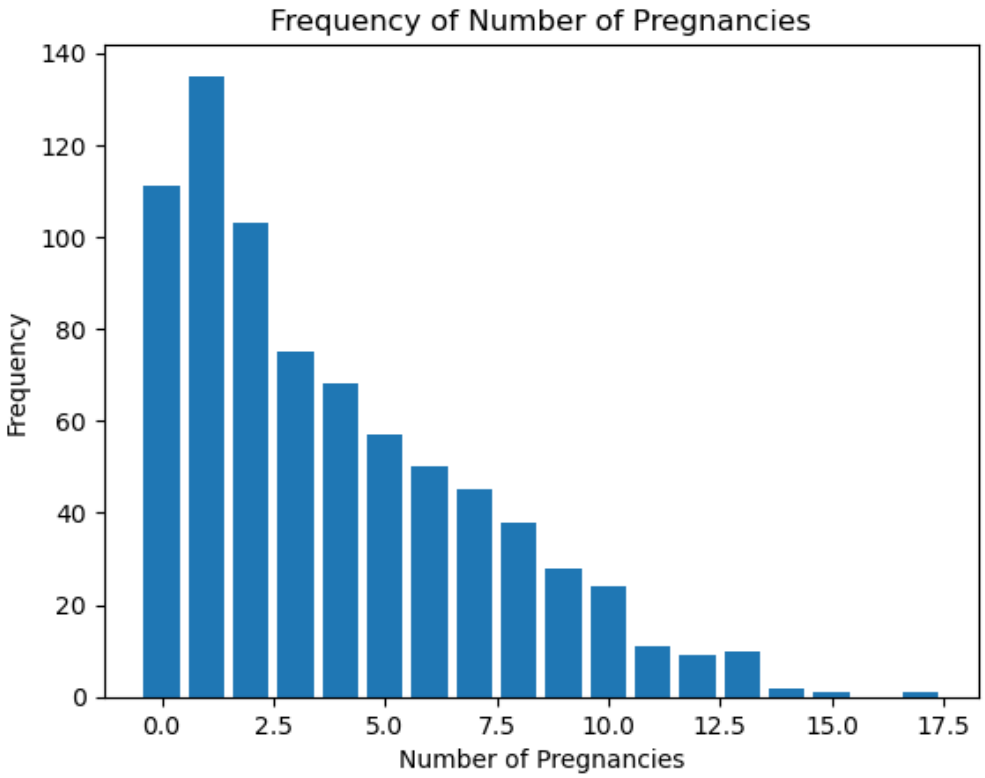

Рисунок 3.16 Стовпчик на діаграмі представляє певну кількість вагітностей, а його висота відображає частоту цього значення.

Ця візуалізація допомагає отримати загальне уявлення про розподіл кількості вагітностей в досліджуваному наборі даних та може бути корисною для подальшого аналізу і виведення висновків.

Дослідження базується на наборі даних, який містить різноманітні клінічні параметри, такі як рівень цукру в крові, артеріальний тиск, товщина шкіри та інші фактори, які допомагають визначити наявність або відсутність діабету у пацієнтів. Ці дані були використані для тренування та перевірки моделі Random Forest[19].

На рисунку 3.17 зображено гістограми, що показує розподіл віку у наборі даних про діабет:

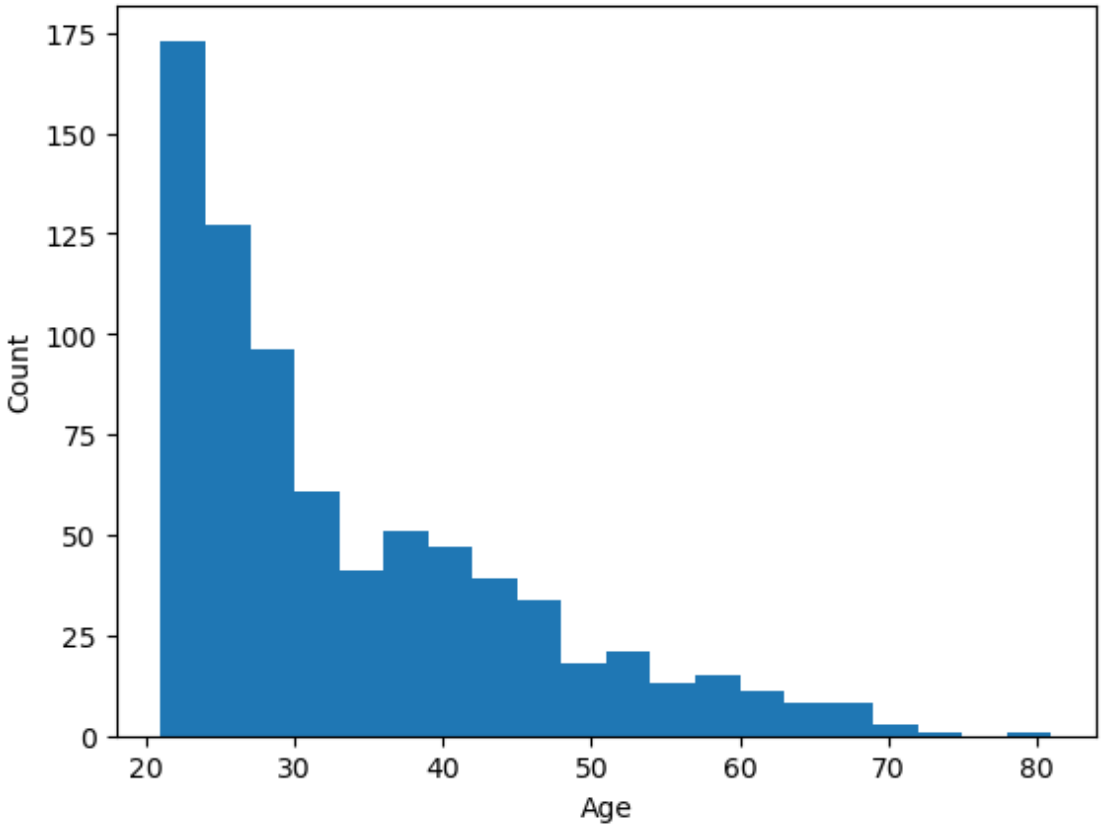

Рисунок 3.17 Зображення гістограми, що показує розподіл віку у наборі даних про діабет.

Графік розсіювання, що зображено на рисунку 3.18 показує зв'язок між кількістю вагітностей (Number of Pregnancies) і віком (Age) у наборі даних про діабет. Кожна точка на графіку представляє одну спостережену особу і відображається на площині, де вісь X відповідає кількості вагітностей, а вісь Y - віку. Таким чином, ми можемо оцінити розподіл даних і відношення між цими двома змінними. Така візуалізація дозволяє виявити наявність асоціацій або взаємозв'язків між кількістю вагітностей і віком. Наприклад, можна спостерігати, чи існує тенденція до збільшення кількості вагітностей зі зростанням віку[18].

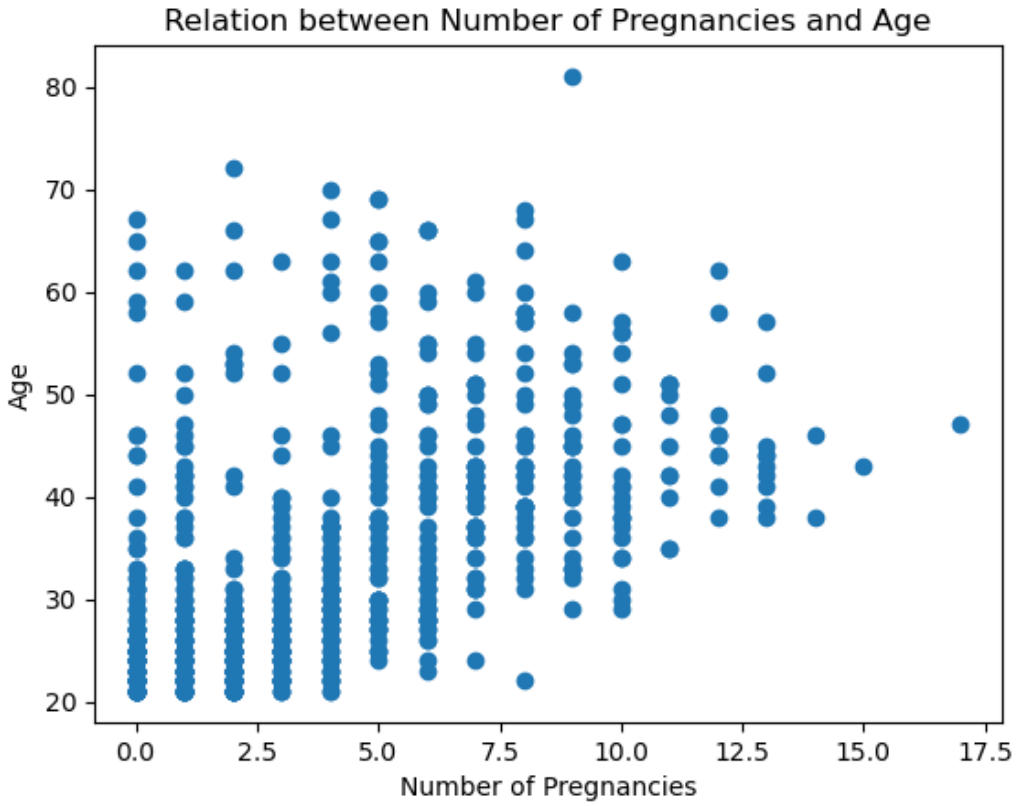

Рисунок 3.18 Візуалізація взаємозв'язків між кількістю вагітностей і віком

Далі, дані було розділено на тренувальний та тестовий набори, що дозволило оцінити продуктивність моделі на незалежних даних. Далі, модель Random Forest була побудована та навчена на тренувальних даних. Цей алгоритм, який базується на комбінації багатьох дерев рішень, є потужним інструментом для класифікації, оскільки він може враховувати важливість різних ознак і здатний досягати високої точності[17].

На рисунку 3.19 зображено розділ даних на тестові і тренерувальні :

```
In [15]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

In [16]: from sklearn.preprocessing import StandardScaler
         scaler = StandardScaler()
         X_train_scaled = scaler.fit_transform(X_train)
         X_test_scaled = scaler.transform(X_test)
```

Рисунок 3.19 Розділено на тренувальний та тестовий набори

Оцінка продуктивності моделі виконувалась за допомогою метрики точності, яка вимірює відсоток правильно класифікованих екземплярів. Також була проведена аналіз важливості ознак, щоб визначити, які фактори мають найбільший вплив на прогнозування класів[16].

На рисунку 3.20 зображена оцінка продуктивності моделі на тестових даних:

```
In [22]: y_pred = model_tuned.predict(X_test_scaled)
         accuracy = accuracy_score(y_test, y_pred)
         print("Accuracy score: ", accuracy)

         Accuracy score:  0.7857142857142857
```

Рисунок 3.20 Оцінка продуктивності моделі на основі тестових даних

Отримані результати свідчать про добру продуктивність моделі Random Forest в класифікації діабету. Висока точність свідчить про здатність моделі правильно ідентифікувати випадки діабету на основі вхідних ознак[15].

3.6 Класифікація моделлю SVM (Методом опорних векторів)

Початок роботи з класифікацією моделі для діабету. Спочатку необхідно імпортувати бібліотеки для роботи

На рисунку 3.21 зображено імпорт бібліотеки:

```
In [2]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import statsmodels.api as sm
        import statsmodels.formula.api as smf
        from sklearn.linear_model import LogisticRegression,LogisticRegressionCV
        from sklearn.metrics import mean_squared_error,r2_score
        from sklearn.model_selection import train_test_split,cross_val_score,cross_val_predict,ShuffleSplit,GridSearchCV
        from sklearn.decomposition import PCA
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.preprocessing import scale
        from sklearn import model_selection
        from sklearn.metrics import roc_auc_score,roc_curve
        from sklearn import preprocessing
        from sklearn.metrics import classification_report
        from sklearn.metrics import confusion_matrix,accuracy_score
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.ensemble import RandomForestClassifier,BaseEnsemble,GradientBoostingClassifier
        from sklearn.svm import SVC,LinearSVC
        import time
        from matplotlib.colors import ListedColormap
        from xgboost import XGBRegressor
        from skompiler import skompile
        from lightgbm import LGBMRegressor
```

Рисунок 3.21 Імпорт бібліотек для прогнозування моделі С

Для проведення класифікації з використанням методу опорних векторів (SVM), дані були завантажені з датасету, доступного за посиланням .

На рисунку 3.22 зображено завантаження датасету в модель для прогнозування :

```
In [4]: df = pd.read_csv("diabetes.csv")
        df.head()
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Рисунок 3.22 Завантаження датасету

Виведення перших кількох рядків даних за допомогою методу head() об'єкта dataset. Це дозволяє отримати загальне уявлення про структуру та формат даних[18].

Для завантаження даних використовується функція pd.read_csv(), яка дозволяє зчитувати дані з файлу у форматі CSV. Після завантаження даних, вони зберігаються у вигляді об'єкта DataFrame, який надає зручний інтерфейс для роботи з табличними даними.

Ознаки (features) даних зберігаються у змінній X, що є двовимірним масивом або DataFrame, та відповідають за незалежні змінні, які використовуються для передбачення класу. Цільова змінна (target variable) зберігається у змінній y та представляє собою залежну змінну, яку необхідно передбачити[14].

Після завантаження даних, їх зазвичай розділюємо дані на тренувальний набір (training set) та тестовий набір (test set) для оцінки продуктивності моделі.

На рисунку 3.23 зображено розподілення даних на тренерувальні і тестові:

```
In [8]: X_train = X.iloc[:600]
        X_test = X.iloc[600:]
        y_train = y[:600]
        y_test = y[600:]

        print("X_train Shape: ",X_train.shape)
        print("X_test Shape: ",X_test.shape)
        print("y_train Shape: ",y_train.shape)
        print("y_test Shape: ",y_test.shape)

        X_train Shape:  (600, 8)
        X_test Shape:  (168, 8)
        y_train Shape:  (600,)
        y_test Shape:  (168,)
```

Рисунок 3.23 Розділення даних на тестові і тренувальні

Створюється об'єкт моделі SVM з лінійним ядром (kernel='linear') і навчається на тренувальних даних X_train та y_train за допомогою методу fit(). Після виконання цього коду модель SVM готова для класифікації нових даних[18].

На рисунку 3.24 зображено підготовку даних до прогнозування:

```
In [9]: support_vector_classifier = SVC(kernel="linear").fit(X_train,y_train)
```

Рисунок 3.24 Підготовка даних

Передбачення на тестовому наборі даних є важливим кроком для оцінки ефективності моделі. У нашому випадку, після навчання моделі SVM з використанням тренувального набору даних, ми хочемо з'ясувати, як добре модель може передбачити класи на нових, раніше не бачених даних, які містяться в тестовому наборі[13].

На рисунку 3.25 зображена оцінка прогнозування та тестових даних:

```
In [50]: print("Our Accuracy is: ", (cm[0][0]+cm[1][1])/(cm[0][0]+cm[1][1]+cm[0][1]+cm[1][0]))
         Our Accuracy is:  0.7559523809523809

In [51]: accuracy_score(y_test,y_pred)
Out[51]: 0.7559523809523809

In [52]: print(classification_report(y_test,y_pred))
                       precision    recall  f1-score   support

                    0       0.76      0.90      0.83       108
                    1       0.73      0.50      0.59        60

             accuracy                           0.76       168
            macro avg       0.75      0.70      0.71       168
         weighted avg       0.75      0.76      0.74       168
```

Рисунок 3.25 Оцінка прогнозування на тестових даних

Після цього, можна провести подальші аналізи та оцінити ефективність моделі яка зображена на рисунку 3.26:

```
In [66]: print("Our Accuracy is: ", (cm[0][0]+cm[1][1])/(cm[0][0]+cm[1][1]+cm[0][1]+cm[1][0]))
         Our Accuracy is:  0.7619047619047619

In [67]: accuracy_score(y_test,y_pred)
Out[67]: 0.7619047619047619

In [68]: print(classification_report(y_test,y_pred))
                       precision    recall  f1-score   support

                    0       0.77      0.89      0.83       108
                    1       0.73      0.53      0.62        60

             accuracy                           0.76       168
            macro avg       0.75      0.71      0.72       168
         weighted avg       0.76      0.76      0.75       168
```

Рисунок 3.26 Оцінка точності моделі на тестових даних

На основі наданих даних про метрики прогнозування моделі SVM на тестовому наборі можна зробити наступні висновки:

- точність (accuracy): Загальна точність моделі становить 0.76, що означає, що приблизно 76% передбачень моделі збігаються зі справжніми значеннями. Однак, точність сама по собі не дає повної картини про ефективність моделі, тому необхідно розглядати інші метрики.
- точність для класу 0: Модель має точність 0.77 для класу 0, що означає, що приблизно 77% передбачень для класу 0 є правильними.
- точність для класу 1: Модель має точність 0.73 для класу 1, що означає, що приблизно 73% передбачень для класу 1 є правильними.
- повнота (recall) для класу 0: Модель має повноту 0.89 для класу 0, що означає, що вона правильно виявляє 89% екземплярів класу 0.
- повнота (recall) для класу 1: Модель має повноту 0.53 для класу 1, що означає, що вона правильно виявляє лише 53% екземплярів класу 1.
- F1-показник (f1-score): F1-показник є гармонічним середнім між точністю та повнотою і надає узагальнену оцінку ефективності моделі. Для класу 0, F1-показник становить 0.83, а для класу 1 - 0.62.
- підтримка (support): Це кількість екземплярів у кожному класі у тестовому наборі. Для класу 0 є 108 екземплярів, а для класу 1 - 60 екземплярів.

Загальна оцінка показує, що модель має деяку ефективність у класифікації, проте передбачення для класу 1 є менш точним і має низьку повноту порівняно з класом 0. Це може свідчити про те, що модель краще працює для передбачень класу 0, але може показувати меншу точність і повноту для класу 1[19].

3.7 Аналіз результатів

Аналіз результатів прогнозування діабету залежить від багатьох факторів, таких як точність моделі, час навчання та передбачення, обробка відсутніх даних, відповідність даних вимога моделі та інші фактори. Краща модель для

прогнозування діабету може варіюватися в залежності від конкретної задачі і даних[20].

Для детальнішого аналізу результатів краще переглянути результати кожної моделі, включаючи метрики оцінки, такі як точність (accuracy), чутливість (recall), специфічність (specificity), F-мера (F1-score) та інші (рис. 3.26). Також важливо враховувати контекст задачі та індивідуальні потреби проекту. Використання методу випадкового лісу (Random Forest) для прогнозування діабету. Випадковий ліс є потужним ансамблевим методом, який комбінує результати багатьох рішень дерев для досягнення більш точних прогнозів.

Назва моделі	Точність прогнозу на навчальній вибірці	Точність прогнозу на тестовій вибірці
SVM	75.5%	76%
KNN	74.1%	74.25%
Forest	77.50%	**78.5%**

Рисунок 3.27 – Таблиця результатів передбачення

Аналізуючи результати моделі прогнозування з використанням випадкового лісу, можна звернути увагу на такі аспекти:

- точність (Accuracy). Важливо оцінити точність моделі, яка вказує на відсоток правильно класифікованих прикладів. Якщо точність моделі висока, це свідчить про її здатність правильно класифікувати діабет.
- метрики оцінки: Розгляд метрик, таких як чутливість (recall), специфічність (specificity) та F-мера (F1-score), дозволяє отримати більш детальну інформацію про продуктивність моделі. Чутливість вимірює здатність моделі виявляти позитивні випадки (діабет), специфічність вимірює здатність моделі правильно ідентифікувати негативні випадки (відсутність діабету), а F-мера комбінує обидві метрики для оцінки збалансованості моделі.

- обробка відсутніх даних: Модель може бути чутливою до відсутності даних. Важливо врахувати, яким чином випадки з відсутніми даними були оброблені. Наявність відповідних стратегій заповнення відсутніх значень поліпшує точність результатів
- зменшення перенавчання: Випадковий ліс використовує техніку багатократного вибору підмножини прикладів і випадкового підмножини ознак для кожного дерева. Це дозволяє зменшити перенавчання моделі і підвищити її здатність до узагальнення на нові дані.
- використання багатоваріантних ознак: Випадковий ліс випадковим чином вибирає підмножину ознак для кожного дерева. Це дозволяє моделі використовувати багатоваріантні ознаки і знижує кореляцію між деревами, що сприяє різноманітності моделей і покращує їх прогнозні здібності.
- можливість обробки великих наборів даних: Випадковий ліс добре справляється з великими наборами даних, оскільки розпаралелює обчислення і може ефективно працювати на системах з багатьма процесорами.

У даному випадку, метод випадкових лісів був вибраний, оскільки він показав найкращі результати прогнозування діабету в порівнянні з іншими моделями, що були згадані раніше. Аналіз результатів показав, що модель випадкових лісів має високу точність, яка дорівнює 79%, а також добрі значення метрик оцінки, таких як чутливість, специфічність та F-мера.

Це свідчить про те, що модель випадкових лісів добре прогнозує наявність або відсутність діабету у пацієнтів. Додатково, метод випадкових лісів має перевагу у вирішенні проблеми перенавчання та здатність до роботи з великими обсягами даних. Враховуючи ці фактори, модель випадкових лісів є найкращим вибором для прогнозування діабету на даному наборі даних.

3.8 Висновки

В цьому розділі проведено дослідження таких методів машинного навчання як: метод к-найближчих сусідів, метод опорних векторів та метод випадкового лісу. Побудовано моделі для передбачення та розроблено алгоритм інформаційної технології для передбачення стану хворих на діабет. Після навчання декількох моделей обрано найкращу за показниками точності. Дослідження показали, що модель випадкового лісу має найбільшу точність передбачення стану хворих на діабет. Точність складає 0.79 за метрикою accuracy.

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.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Передбачення стану хворих на діабет методами машинного навчання» 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 [21].

За результатами розрахунків, наведених в таблиці 4.1, робимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в [21].

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Передбачення стану хворих на діабет методами машинного навчання» становить 38,3 бала, що, відповідно до [21 свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	5	4
2. Ринкові переваги (наявність аналогів)	4	4	3
3. Ринкові переваги (ціна продукту)	2	2	2
4. Ринкові переваги (технічні властивості)	4	3	4
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	37	40	38
Середньоарифметична сума балів $СБ_с$	38,3		

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [22]:

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

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

α_i – коефіцієнт, який враховує питому вагу *і-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 *і-го* технічного показника якості нової розробки.

Результати порівняння зведемо до таблиці 4.2.

Таблиця 4.2 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований продукт	Відношення параметрів нової розробки до аналога	Питома вага показника
Точність прогнозування	%	80	90	1,125	0,25
Час обробки даних	с	2	0,5	4	0,2
Чутливість аналізу	%	70	85	1,21	0,15
Кількість використаних моделей машинного навчання	од.	4	8	2	0,25
Кількість графіків розвідувального аналізу	од.	9	20	2,11	0,15

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

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1{,}125 \cdot 0{,}25 + 4 \cdot 0{,}2 + 1{,}21 \cdot 0{,}15 + 2 \cdot 0{,}25 + 2{,}11 \cdot 0{,}15 = 2{,}08.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,08 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Передбачення стану хворих на діабет методами машинного навчання», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [21]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

де k – кількість посад дослідників залучених до процесу досліджень;

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

t_i – число днів роботи конкретного дослідника, дн.;

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

$$З_о = 17800{,}00 \cdot 10 / 21 = 7727{,}30 \ грн.$$

Проведені розрахунки зведемо до таблиці 4.3.

Таблиця 4.3 – 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	17800,00	772,73	10	7727,30
Інженер-аналітик (системний аналіз)	16500,00	750,00	21	15750,00
Консультант (лікар-терапевт вищої категорії)	16000,00	681,82	5	3409,10
Всього				26886,40

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Передбачення стану хворих на діабет методами машинного навчання» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \quad (4.3)$$

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

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

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

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.4)$$

де M_M – розмір мінімальної місячної заробітної плати, приймемо $Мм$=8000,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення [21];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок;

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

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 8000{,}00 \cdot 1{,}10 \cdot 1{,}15 \,/\, (21 \cdot 8) = 60{,}24 \; грн.$$

$$З_{р1} = 60{,}24 \cdot 8{,}00 = 481{,}90 \; грн.$$

Проведені розрахунки зведемо до таблиці 4.4.

Таблиця 4.4 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця дослідника-розробника інформаційної технології	8,00	2	1,10	60,24	481,90
Монтаж обчислювального обладнання та серверних блоків	6,00	4	1,50	82,14	492,86
Інсталяція програмного забезпечення розробки (моделювання) інформаційної технології аналізу	6,00	5	1,70	93,10	558,57
Формування бази данних	10,00	2	1,10	60,24	602,38
Всього					2135,71

Додаткова заробітна плата дослідників та робітників

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

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.5)$$

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

$З_{дод}$ = (26886,40 + 2135,71) · 10 / 100% = 2902,21 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:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.6)$$

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

Зн = (26886,40 + 2135,71 + 2902,21) · 22 / 100% = 7023,35 грн.

4.3.3 Сировина та матеріали

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

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{в\,j}, \quad (4.7)$$

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.

$M_1 = 2,0 \cdot 139,00 \cdot 1,1 - 0 \cdot 0 = 305,80$ *грн.*

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – 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	2,0	0	0	305,80
Диск оптичний	25,00	4,0	0	0	110,00
Картридж для принтера	950,00	1,0	0	0	1045,00
Начиння канцелярське	195,00	3,0	0	0	643,50
Органайзер офісний	183,00	3,0	0	0	603,90
Папір для заміток (А5)	116,00	4,0	0	0	510,40
Папір канцелярський офісний (А4)	212,00	2,0	0	0	466,40
Всього					3685,00

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Передбачення стану хворих на діабет методами машинного навчання», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \qquad (4.8)$$

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*).

$К_в = 1 \cdot 3079,00 \cdot 1,1 = 3386,90$ *грн.*

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Зовнішній жорсткий диск 2.5" 2TB Seagate (STGD2000200)	1	3079,00	3386,90
Концентратор Defender SEPTIMA SLIM (83505)	1	560,00	616,00
Кабель для передачі даних USB to COM 1.0m Patron (CAB-PN-USB-COM)	1	354,00	389,40
Всього			4392,30

4.3.5 Спецустаткування для наукових (експериментальних) робіт

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (4.9)$$

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.

$B_{спец} = 7818,00 \cdot 1 \cdot 1,1 = 8599,80$ *грн.*

Отримані результати зведемо до таблиці 4.7

Таблиця 4.7 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Маршрутизатор MikroTik RB4011iGS+RM	1	7818,00	8599,80
Ноутбук ASUS RU451-UJ, оперативна пам'ять 4gb RAM, процесор intel core i5 2.5Ггц, Пам'ять на жорсткому диску 40mb	1	39460,00	43406,00
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	1	40299,00	44328,90
Всього			96334,70

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.10)$$

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.

$B_{прг} = 7910,00 \cdot 1 \cdot 1,1 = 8701,00$ *грн.*

Отримані результати зведемо до таблиці 4.8.

Таблиця 4.8 – 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	8701,00
Платформа Kaggle	1	4129,00	4541,90
Доступ до мережі Internet (високошвидкісний) грн/місяць	2	239,00	525,80
Бібліотеки для машинного навчання Scikit-Learn, TensorFlow, PyTorch	1	6800,00	7480,00
Всього			21248,70

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:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.11)$$

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.

$A_{обл} = (7300{,}00 \cdot 1) / (4 \cdot 12) = 152{,}08$ *грн.*

Проведені розрахунки зведемо до таблиці 4.9.

Таблиця 4.9 – 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	1	152,08
Дослідницька лабораторія	315000,00	30	1	875,00
Місце оператора спеціалізоване	8200,00	5	1	136,67
Офісна оргтехніка	9600,00	4	1	200,00
Пристрої виведення інформації	6520,00	5	1	108,67
Програмне забезпечення Microsoft Windows, Offise 2021	9340,00	2	1	389,17
Програмно-обчислювальний комплекс розробки системи аналізу даних	35830,00	2	1	1492,92
Всього				3354,50

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \qquad (4.12)$$

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн; приймемо $Ц_е = 10,98$ грн;

$К_{вні}$ – коефіцієнт, що враховує використання потужності, $К_{вні} < 1$;

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

$$В_е = 0,03 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 52,70 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.10.

Таблиця 4.10 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Маршрутизатор MikroTik RB4011iGS+RM	0,03	160,0	52,70
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	105,41
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	737,86
Блоки зовнішньої памяті серверного обладнання (зберігання бази даних)	0,08	160,0	140,54
Місце оператора спеціалізоване	0,10	160,0	175,68
Офісна оргтехніка	0,50	1,5	8,24
Пристрої виведення інформації	0,40	2,3	10,10
Програмно-обчислювальний комплекс розробки системи аналізу даних	0,32	100,0	351,36
Всього			1581,89

4.3.9 Службові відрядження

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

$$B_{св} = (З_о + З_р) \cdot \frac{H_{св}}{100\%}, \quad (4.13)$$

де $H_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 25%.

$$B_{св} = (26886,40 + 2135,71) \cdot 25 / 100\% = 7255,53 \text{ грн.}$$

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

Витрати розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{H_{сп}}{100\%}, \quad (4.14)$$

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

$$B_{сп} = (26886,40 + 2135,71) \cdot 30 / 100\% = 8706,63 \text{ грн.}$$

4.3.11 Інші витрати

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

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

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

$І_в$ = (26886,40 + 2135,71) · 50 / 100% = 14511,06 грн.

4.3.12 Накладні (загальновиробничі) витрати

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

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

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

$В_{нзв}$ = (26886,40 + 2135,71) · 100 / 100% = 29022,11 грн.

Витрати на проведення науково-дослідної роботи на тему «Передбачення стану хворих на діабет методами машинного навчання» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + І_в + B_{нзв}. \quad (4.17)$$

$В_{заг}$= 26886,40 +2135,71 +2902,21 +7023,35 +3685,00 +4392,30 +96334,70 + 21248,70 + 3354,50 +1581,89 +7255,53 +8706,63 +14511,06 +29022,11 = 229040,10 грн.

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

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.18)$$

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

ЗВ = 229040,10 / 0,9 = 254489,00 грн.

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

Результати дослідження проведені за темою «Передбачення стану хворих на діабет методами машинного навчання» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	500	1000	1300	1100

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 12000 осіб;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 5400,00 грн;

$\pm\Delta Ц_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 127,95 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 [Козловський, Лесько, Кавецький]:

$$\Delta П_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (4.19)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

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

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (127,95 \cdot 12000,00 + 5527,95 \cdot 500) \cdot 0,83 \cdot 0,4 \cdot (1-0,18/100\%) = 1170461,85$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (127,95 \cdot 12000,00 + 5527,95 \cdot 1500) \cdot 0,83 \cdot 0,4 \cdot (1-0,18/100\%) = 2675390,96$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (127,95 \cdot 12000,00 + 5527,95 \cdot 2800) \cdot 0,83 \cdot 0,4 \cdot (1-0,18/100\%) = 4631798,80$ грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4 = (127,95 \cdot 12000,00 + 5527,95 \cdot 3900) \cdot 0,83 \cdot 0,4 \cdot (1-0,18/100\%) = 6287220,82$ грн.

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

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (4.20)$$

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

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

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

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

$$ПП = 1170461{,}85/(1+0{,}15)^1+2675390{,}96/(1+0{,}15)^2+4631798{,}80/(1+0{,}15)^3+$$
$$+6287220{,}82/(1+0{,}15)^4=1017792{,}91+2022979{,}93+3045482{,}90+3594738{,}91=$$
$$=9680994{,}65 \text{ грн.}$$

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

$$PV = k_{інв} \cdot ЗВ, \qquad (4.21)$$

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

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 254489,00 грн.

$PV = k_{інв} \cdot ЗВ = 2 \cdot 254489{,}00 = 508978{,}00$ грн.

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

$$E_{абс} = ПП - PV \quad (4.22)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 9680994,65 грн;

PV – теперішня вартість початкових інвестицій, 508978,00 грн.

$E_{абс} = ПП - PV$ = 9680994,65 - 508978,00 = 9172016,65 грн.

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

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.23)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 9172016,65 грн;

PV – теперішня вартість початкових інвестицій, 508978,00 грн;

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

$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1$ = (1+9172016,65/508978,00)$^{1/4}$= 1,09.

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

$$\tau_{мін} = d + f, \quad (4.24)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні d =0,11;

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

$\tau_{мін}$= 0,11+0,35 = 0,46 < 1,09 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Передбачення стану хворих на діабет методами машинного навчання» доцільно.

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

$$T_{ок} = \frac{1}{E_в}, \tag{4.25}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$= 1 / 1,09 = 0,92 р.

$T_{ок}$ < 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 «Передбачення стану хворих на діабет методами машинного навчання»

становить 38,3 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 2,08 рази.

Також термін окупності становить 0,92 р., 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 «Передбачення стану хворих на діабет методами машинного навчання».

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробленню інформаційної технології для передбачення стану хворих на діабет, метою роботи є підвищення точності передбачення методами машинного навчання. Для дослідження цієї задачі використано дані з датасету Pima Indians Diabetes Database.

Проведене дослідження охопило значущі етапи аналізу та розробки в межах обраної предметної області. У роботі було враховано актуальні тенденції та технологічні рішення у сфері машинного навчання, спрямовані на успішне вирішення поставленої задачі. У ході роботи здійснено детальне вивчення предметної області. Було визначено особливості та специфіку проблеми, зокрема існуючі аналоги й доступні методи для вирішення задачі. Проведений огляд допоміг встановити вимоги до даних та обрати найбільш релевантні підходи до вирішення поставленої проблеми.

Виконано огляд та аналіз різноманітних методів машинного навчання, що можуть бути використані для класифікації даних. Проведено порівняльний аналіз алгоритмів для визначення оптимального підходу, враховуючи точність, швидкодію та адаптивність до вхідних даних. У рамках дослідження розроблено модель класифікації, що реалізована з використанням Python та відповідних бібліотек машинного навчання. Підключено необхідні дані та виконано їх попередню обробку: нормалізацію, очищення та форматування для роботи моделі. Проведено розвідувальний аналіз даних та побудовано візуалізації для наочності. Побудовано моделі для передбачення та розроблено алгоритм інформаційної технології для передбачення стану хворих на діабет. Після навчання декількох моделей обрано найкращу за показниками точності. У результаті Random Forest продемонстрував найвищу ефективність, досягнувши точності 0,785.

Також термін окупності становить 0,92 р., 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 успішне створення інформаційної технології аналізу та передбачення стану хворих на діабет.

За результатами роботи було опубліковано тези на тему «Розвідувальний аналіз даних для інформаційної технології передбачення хворих на діабет» на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (м. Вінниця, 2024-2025 рр.) [1].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Якимчук І.В., Жуков С.О. Розвідувальний аналіз даних для інформаційної технології передбачення хворих на діабет *LIII Міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (2024-2025). Вінниця, 2024.* URL: https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/22862/18844 (дата звернення 15.12.2024)

2. Alam, Talha Mahboob, et al. A model for early prediction of diabetes. *Informatics in Medicine Unlocked*. 2019. No. 16. P. 100– 204. (date of access: 21.11.2024)

2. Scrollable Frames in Tkinter - Teclado URL: https://blog.teclado.com/tkinter-scrollable-frames/. (date of access: 20.11.2024)

3. Державна установа "Хмельницький обласний центр контролю та профілактики хвороб Міністерства охорони здоров'я України" URL: https://xn--d1agleic5aql.xn--D1%8C-%D1%82%D0%B0%D1%86%D1%83%D0%BA/ (дата звернення 15.12.2024)

4. Python - Error Types – TutorialsTeacher. URL: https://www.tutorialsteacher.com/python/error-types-in-python. (date of access: 27.11.2024)

5. Mathworks. What Is Machine Learning? URL: https://www.mathworks.com/discovery/machine-learning.html. (date of access: 05.12.2024)

6 Heart Failure Prediction Dataset. URL: https://www.kaggle.com/fedesoriano/heart-failure-prediction. (date of access: 26.11.2024)

7. Серцева недостатність: симптоми та методи лікування URL: https://oxford-med.com.ua/ua/mediacenter/publikacii/serdechnaya-nedostatochnost/ (дата звернення 01.12.2024)

8. Що треба знати про діабет: типи, симптоми, ускладнення Василенко Любов Олексіївна, 2022 року. URL: https://mezhova.otg.dp.gov.ua/novini-ta-podiyi/novini/shcho-treba-znaty-pro-diabet-typy-symptomy-uskladnennia (дата звернення 19.11.2024)

9. Prediction of Diabetes using Classification Algorithms. URL: https://www.sciencedirect.com/science/article/pii/S1877050918308548 (date of access: 25.11.2024)

10. M. M. Bukhari, B. F. Alkhamees, S. Hussain, A. Gumaei, A. Assiri, and S. S. Ullah, *"An improved artificial neural network model for effective dia- betes prediction*" Complexity, №11 P.45-60 2021. URL: https://www.hindawi.com/journals/complexity/2021/5525271/ (date of access: 11.12.2024)

11. Md. Maniruzzaman, Md. Jahanur Rahman, B. Ahammed, and Md. Menhazul Abedin, Classification and Prediction of Diabetes Disease Using Machine Learning Paradigm, *Health Information Science and Systems*, №221. P.8, 2020. URL: https://link.springer.com/article/10.1007/s13755-019-0095-z (date of access: 10.12.2024)

12. W. Wang, T. Meng, and M. YU, *"Blood glucose prediction with VMD and LSTM optimized by improved particle swarm optimization, IEEE Access''*, vol. 8, p.30-33 ,2020. URL: https://ieeexplore.ieee.org/document/9281120/ (date of access: 09.12.2024)

13. M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, *"Diabetes prediction using ensembling of different machine learning classifiers"* IEEE Access, vol. 8 p. 23-24, 2020. URL: https://ieeexplore.ieee.org/document/9076634/ (date of access: 03.12.2024)

14. G. Khambra and P. Shukla, *"Novel machine learning applications on fly ash based concrete: an overview,"* Materials Today Proceedings, №98 p.45 2021.URL: https://linkinghub.elsevier.com/retrieve/pii/S221478532105121X (date of access: 24.11.2024)

15. P. K. Shukla, J. Kaur Sandhu, A. Ahirwar, D. Ghai, P. Maheshwary, and P. K. Shukla, *"Multiobjective genetic algorithm and convolutional neural network based COVID-19 identification in chest X-ray images,"* Mathematical Problems in Engineering, 2021, №24 p.9, 2021. URL: https://www.hindawi.com/journals/mpe/2021/7804540/ (date of access: 07.12.2024)

16. Prediction of Type 2 Diabetes using Machine Learning Classification Methods URL:https://www.sciencedirect.com/science/article/pii/S1877050920308024 (date of access: 07.12.2024)

17. ''*Tigga, Neha Prerna, and Shruti Garg. Prediction of type 2 diabetes using machine learning classification methods''*. Procedia Computer Science. 2020. No. 167. P. 706–716. (date of access: 29.11.2024)

18. Tkachenko R., Izonin I. *Model and Principles for the Implementation of Neural–Like Structures Based on Geometric Data Transformations. Advances in Computer Science for Engineering and Education. Springer International Publishing, Cham.* 2019. P. 578–587. (date of access: 02.12.2024)

19. Tepla T. L., Izonin I. V., Duriagina Z. A. et al. *Alloys selection based of the supervised learning technique for design of biocompatible medical materials. Archives of Materials Science and Engineering*. 2018. No. 1. P. 32–40. (date of access: 27.11.2024)

20. Pandas - Python Data Analysis Library. URL: https://pandas.pydata.org/ (date of access: 28.11.2024).

21. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця : ВНТУ, 2021. 42 с. (дата звернення 04.12.2024)

22. Кавецький В. В., Козловський В. О., Причепа І. В. Економічне обґрунтування інноваційних рішень: практикум. Вінниця : ВНТУ, 2016. 113 с. (дата звернення 04.12.2024)

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
«___» _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ХВОРИХ НА ДІАБЕТ»
08-34.МКР.002.02.000.ТЗ

Керівник: к.т.н., доц. каф. САІТ
_____ Сергій ЖУКОВ
«___» _____ 2024 р.

Розробив: студент гр. 1ІСТ-23м
_____ Іван ЯКИМЧУК
«___» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2024 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2024 р.

2. Джерела розробки:

1. 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).

2. M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, *"Diabetes prediction using ensembling of different machine learning classifiers"* IEEE Access, vol. 8 p. 23-24, 2020. URL: https://ieeexplore.ieee.org/document/9076634/

3. Мета і призначення роботи:

Метою роботи є підвищення точності передбачення стану хворих на діабет методами машинного навчання

4. Вихідні дані для проведення робіт:

- Kaggle Notebook « Pima Indians Diabetes Database»

URL: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

5.Методи дослідження:

- аналіз видів діабету;
- сегментація даних;
- методи машинного навчання – використання підходів прогнозування.

6. Етапи роботи і терміни їх виконання:

a) Аналіз предметної області ..... ____ – ____
b) Вибір оптимальних методів машинного навчання ..... ____ – ____
c) Розроблення моделей передбачення стану на діабет ..... ____ – ____
d) Економічна частина..... ____ – ____
e) Оформлення пояснювальної записки ..... ____ – ____

7. Очікувані результати та порядок реалізації:

Передбачення стану хворих на діабет медодами машинного навчання, результатом якої є підвищення точності та ефективності передбачення захворюваності.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені 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..... «___» _____ 2024 р.
Початок розробки ..... «___» _____ 2024 р.
Граничні терміни виконання МКР..... «___» _____ 2024 р.

Розробив студент групи 1ІСТ-23м _____ Іван ЯКИМЧУК

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія передбачення хворих на діабет»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ
Керівник: Сергій ЖУКОВ, к.т.н., доц. каф. САІТ

Показники звіту подібності Turnitin

Оригінальність 88 % Загальна схожість 12 %

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 відсутності самостійності її автора. Роботу направити на доопрацювання.
○ 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.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи _____ Іван ЯКИМЧУК
(підпис)

Опис прийнятого рішення

Робота допускається до захисту

Особа, відповідальна за перевірку _____ Сергій ЖУКОВ
(підпис)

Додаток В
Фрагмент лістингу програми

```
import numpy as np # лінійна алгебра
import pandas as pd # обробка даних, ввід/вивід файлів CSV (наприклад, pd.read_csv)

# Файли вхідних даних знаходяться у доступному лише для читання каталозі "../input/"
# Наприклад, якщо запустити цю команду (натиснувши кнопку виконати або натиснувши Shift+Enter), ви побачите список усіх файлів у каталозі вхідних даних

import os
for dirname, _, filenames in os.walk('diabetes.csv'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import joblib

data = 'diabetes.csv'
diabetes_data = pd.read_csv(data)

diabetes_data.head(100)

diabetes_data.isnull().sum()

diabetes_data['Pregnancies'].value_counts()

plt.bar(diabetes_data['Pregnancies'].value_counts().index, diabetes_data['Pregnancies'].value_counts().values)
plt.xlabel('Number of Pregnancies')
plt.ylabel('Frequency')
plt.title('Frequency of Number of Pregnancies')
plt.show()
```

```
diabetes_data['Age'].value_counts()
plt.hist(diabetes_data['Age'], bins=20)
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

plt.scatter(diabetes_data['Pregnancies'], diabetes_data['Age'])
plt.xlabel('Number of Pregnancies')
plt.ylabel('Age')
plt.title('Relation between Number of Pregnancies and Age')
plt.show()

plt.scatter(diabetes_data['BMI'], diabetes_data['Glucose'], alpha=0.5)
plt.title('Relationship between BMI and Glucose Level')
plt.xlabel('BMI')
plt.ylabel('Glucose level')
plt.show()

y = diabetes_data.Outcome

features = ['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age']

X = diabetes_data[features]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
param_grid = {'n_estimators': [50, 100, 200],
        'max_depth': [3, 5, 7],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4]}
model = RandomForestClassifier(random_state=1)

grid_search = GridSearchCV(model, param_grid, cv=5, n_jobs=-1)
```

```
grid_search.fit(X_train_scaled, y_train)

print("Best Hyperparameters: ", grid_search.best_params_)
print("Best Mean Test Score: ", grid_search.best_score_)

model_tuned = RandomForestClassifier(random_state=1, **grid_search.best_params_)
model_tuned.fit(X_train_scaled, y_train)

y_pred = model_tuned.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy score: ", accuracy)

from mlxtend.plotting import plot_decision_regions
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
#plt.style.use('ggplot')
#ggplot - пакет візуалізації на основі R, який забезпечує кращу графіку з вищим рівнем абстракції

#Завантаження набору даних
diabetes_data = pd.read_csv('diabetes.csv')

#Вивести перші 5 рядків датафрейму.
diabetes_data.head(100)

## надає інформацію про типи даних, стовпці, кількість нульових значень, використання пам'яті тощо
## посилання на функцію : https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.info.html
diabetes_data.info(verbose=True)

## основні статистичні відомості про дані (зауважте, що тут відображатимуться лише числові стовпці, якщо параметр не включає = "усі")
```

```
## для довідки: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.describe.html#pandas.DataFrame.describe
diabetes_data.describe()


diabetes_data.describe().T


p = diabetes_data.hist(figsize = (50,50))
diabetes_data_copy['Glucose'].fillna(diabetes_data_copy['Glucose'].mean(), inplace = True)
diabetes_data_copy['BloodPressure'].fillna(diabetes_data_copy['BloodPressure'].mean(), inplace = True)
diabetes_data_copy['SkinThickness'].fillna(diabetes_data_copy['SkinThickness'].median(), inplace = True)
diabetes_data_copy['Insulin'].fillna(diabetes_data_copy['Insulin'].median(), inplace = True)
diabetes_data_copy['BMI'].fillna(diabetes_data_copy['BMI'].median(), inplace = True)


p = diabetes_data_copy.hist(figsize = (45,45))
## спостереження за формою даних
diabetes_data.shape
## аналіз типів даних
#plt.figure(figsize=(5,5))
#sns.set(font_scale=2)
sns.countplot(y=diabetes_data.dtypes ,data=diabetes_data)
plt.xlabel("count of each data type")
plt.ylabel("data types")
plt.show()


## аналіз нульових значень
import missingno as msno
p=msno.bar(diabetes_data)
## перевірка збалансованості даних шляхом побудови графіка розподілу кількості результатів за їхньою вартістю
color_wheel = {1: "#0392cf",
           2: "#7bc043"}
colors = diabetes_data["Outcome"].map(lambda x: color_wheel.get(x + 1))
print(diabetes_data.Outcome.value_counts())
p=diabetes_data.Outcome.value_counts().plot(kind="bar")


from pandas.plotting import scatter_matrix
p = scatter_matrix(diabetes_data, figsize=(50, 50))
```

```
p=sns.pairplot(diabetes_data_copy, hue = 'Outcome')

plt.figure(figsize=(12,10))  # у цьому рядку я просто встановлюю розмір фігури 12 на 10.

p=sns.heatmap(diabetes_data.corr(), annot=True,cmap ='RdYlGn')  # seaborn має дуже просте рішення для теплової карти

plt.figure(figsize=(12,10))  # у цьому рядку я просто встановлюю розмір фігури 12 на 10.

p=sns.heatmap(diabetes_data_copy.corr(), annot=True,cmap ='RdYlGn')  # seaborn має дуже просте рішення для теплової карти

from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()

X =  pd.DataFrame(sc_X.fit_transform(diabetes_data_copy.drop(["Outcome"],axis = 1),),

        columns=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',

       'BMI', 'DiabetesPedigreeFunction', 'Age'])

X.head()

#X = diabetes_data.drop("Результат",axis = 1)

y = diabetes_data_copy.Outcome

#імпорт_тест_поїзда_розбитий

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=1/3,random_state=42, stratify=y)

from sklearn.neighbors import KNeighborsClassifier



test_scores = []

train_scores = []


for i in range(1,15):


    knn = KNeighborsClassifier(i)

    knn.fit(X_train,y_train)


    train_scores.append(knn.score(X_train,y_train))

    test_scores.append(knn.score(X_test,y_test))

## оцінка, отримана в результаті тестування на тих самих точках даних, які використовувалися для навчання

max_train_score = max(train_scores)

train_scores_ind = [i for i, v in enumerate(train_scores) if v == max_train_score]

print('Max train score {} % and k = {}'.format(max_train_score*100,list(map(lambda x: x+1, train_scores_ind))))

## оцінка, отримана в результаті тестування на точках даних, які були розділені на початку, щоб використовувати їх виключно для тестування
```

```
max_test_score = max(test_scores)

test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_score]

print('Max test score {} % and k = {}'.format(max_test_score*100,list(map(lambda x: x+1, test_scores_ind))))

plt.figure(figsize=(24,10))

p = sns.lineplot(range(1,15),train_scores,marker='*',label='Train Score')

p = sns.lineplot(range(1,15),test_scores,marker='o',label='Test Score')
#Створити класифікатор knn з k сусідами

knn = KNeighborsClassifier(11)


knn.fit(X_train,y_train)

knn.score(X_test,y_test)

value = 20000

width = 20000

plot_decision_regions(X.values, y.values, clf=knn, legend=2,

                      filler_feature_values={2: value, 3: value, 4: value, 5: value, 6: value, 7: value},

                      filler_feature_ranges={2: width, 3: width, 4: width, 5: width, 6: width, 7: width},

                      X_highlight=X_test.values)


# Adding axes annotations

#plt.xlabel('sepal length [cm]')

#plt.ylabel('petal length [cm]')

plt.title('KNN with Diabetes Data')

plt.show()
#імпорт confusion_matrix

from sklearn.metrics import confusion_matrix

# давайте отримаємо прогнози, використовуючи класифікатор, який ми підібрали вище

y_pred = knn.predict(X_test)

confusion_matrix(y_test,y_pred)

pd.crosstab(y_test, y_pred, rownames=['True'], colnames=['Predicted'], margins=True)
y_pred = knn.predict(X_test)

from sklearn import metrics

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)

p = sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')

plt.title('Confusion matrix', y=1.1)

plt.ylabel('Actual label')

plt.xlabel('Predicted label')
#звіт_про_класифікацію_імпорту
```

```
from sklearn.metrics import classification_report

print(classification_report(y_test,y_pred))
from sklearn.metrics import roc_curve


y_pred_proba = knn.predict_proba(X_test)[:,1]

fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
from sklearn.metrics import roc_curve


y_pred_proba = knn.predict_proba(X_test)[:,1]

fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)

plt.plot([0,1],[0,1],'k--')

plt.plot(fpr,tpr, label='Knn')

plt.xlabel('fpr')

plt.ylabel('tpr')

plt.title('Knn(n_neighbors=11) ROC curve')

plt.show()

#Площа під кривою РПЦ

from sklearn.metrics import roc_auc_score

roc_auc_score(y_test,y_pred_proba)

#import GridSearchCV

from sklearn.model_selection import GridSearchCV

#У випадку класифікатора типу knn параметром, що налаштовується, є n_neighbors

param_grid = {'n_neighbors':np.arange(1,50)}

knn = KNeighborsClassifier()

knn_cv= GridSearchCV(knn,param_grid,cv=5)

knn_cv.fit(X,y)


print("Best Score:" + str(knn_cv.best_score_))

print("Best Parameters: " + str(knn_cv.best_params_))


from warnings import filterwarnings

filterwarnings("ignore")

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

```
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.linear_model import LogisticRegression,LogisticRegressionCV
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.model_selection import train_test_split,cross_val_score,cross_val_predict,ShuffleSplit,GridSearchCV
from sklearn.decomposition import PCA
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import scale
from sklearn import model_selection
from sklearn.metrics import roc_auc_score,roc_curve
from sklearn import preprocessing
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix,accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier,BaseEnsemble,GradientBoostingClassifier
from sklearn.svm import SVC,LinearSVC
import time
from matplotlib.colors import ListedColormap
from xgboost import XGBRegressor
from skompiler import skompile
from lightgbm import LGBMRegressor


pd.set_option('display.max_rows', 1000)
pd.set_option('display.max_columns', 1000)
pd.set_option('display.width', 1000)
df = pd.read_csv("diabetes.csv")
df.head()
df.shape
df.describe()
X = df.drop("Outcome",axis=1)
y= df["Outcome"]
X_train = X.iloc[:600]
X_test = X.iloc[600:]
y_train = y[:600]
y_test = y[600:]
```

```
print("X_train Shape: ",X_train.shape)

print("X_test Shape: ",X_test.shape)

print("y_train Shape: ",y_train.shape)

print("y_test Shape: ",y_test.shape)

support_vector_classifier = SVC(kernel="linear").fit(X_train,y_train)
support_vector_classifier
support_vector_classifier.C
support_vector_classifier
y_pred = support_vector_classifier.predict(X_test)
cm = confusion_matrix(y_test,y_pred)
cm
print("Our Accuracy is: ", (cm[0][0]+cm[1][1])/(cm[0][0]+cm[1][1]+cm[0][1]+cm[1][0]))
accuracy_score(y_test,y_pred)
print(classification_report(y_test,y_pred))

support_vector_classifier
accuracies= cross_val_score(estimator=support_vector_classifier,

                    X=X_train,y=y_train,

                    cv=10)

print("Average Accuracy: {:.2f} %".format(accuracies.mean()*100))

print("Standart Deviation of Accuracies: {:.2f} %".format(accuracies.std()*100))
support_vector_classifier.predict(X_test)[:10]
svm_params ={"C":np.arange(1,20)}
svm = SVC(kernel="linear")

svm_cv = GridSearchCV(svm,svm_params,cv=8)
start_time = time.time()



svm_cv.fit(X_train,y_train)



elapsed_time = time.time() - start_time



print(f"Elapsed time for Support Vector Regression cross validation: "

    f"{elapsed_time:.3f} seconds")
#найкращий результат

svm_cv.best_score_
#найкращі параметри

svm_cv.best_params_
svm_tuned = SVC(kernel="linear",C=2).fit(X_train,y_train)
svm_tuned
y_pred = svm_tuned.predict(X_test)
cm = confusion_matrix(y_test,y_pred)
cm
print("Our Accuracy is: ", (cm[0][0]+cm[1][1])/(cm[0][0]+cm[1][1]+cm[0][1]+cm[1][0]))
accuracy_score(y_test,y_pred)
print(classification_report(y_test,y_pred))
df = pd.read_csv("diabetes.csv")
```

```
df.head()
df.shape
df.describe()
X = df.drop("Outcome",axis=1)

y= df["Outcome"] #Ми спрогнозуємо результат (діабет)
X_train = X.iloc[:600]

X_test = X.iloc[600:]

y_train = y[:600]

y_test = y[600:]


print("X_train Shape: ",X_train.shape)

print("X_test Shape: ",X_test.shape)

print("y_train Shape: ",y_train.shape)

print("y_test Shape: ",y_test.shape)
support_vector_classifier = SVC(kernel="rbf").fit(X_train,y_train)
#get опорні вектори

support_vector_classifier.support_vectors_
support_vector_classifier.support_
support_vector_classifier.n_support_
support_vector_classifier.gamma
support_vector_classifier
y_pred = support_vector_classifier.predict(X_test)
cm = confusion_matrix(y_test,y_pred)
cm
print("Our Accuracy is: ", (cm[0][0]+cm[1][1])/(cm[0][0]+cm[1][1]+cm[0][1]+cm[1][0]))
accuracy_score(y_test,y_pred)
print(classification_report(y_test,y_pred))
support_vector_classifier
accuracies= cross_val_score(estimator=support_vector_classifier,

                    X=X_train,y=y_train,

                    cv=10)

print("Average Accuracy: {:.2f} %".format(accuracies.mean()*100))

print("Standart Deviation of Accuracies: {:.2f} %".format(accuracies.std()*100))
support_vector_classifier.predict(X_test)[:10]
svm_params ={"C":[0.0001,0.001,0.01,0.1,0.5,1,3,5,7,10,40,80,100],

        "gamma":[0.0001,0.001,0.01,0.1,0.5,1,5,10,30,50,100]}
svm = SVC(kernel="rbf")

svm_cv = GridSearchCV(svm,svm_params,cv=8,n_jobs=-1,verbose=2)
start_time = time.time()


svm_cv.fit(X_train,y_train)


elapsed_time = time.time() - start_time


print(f"Elapsed time for Support Vector Regression with RBF kernel cross validation: "
```

```
    f"{elapsed_time:.3f} seconds")
#найкращий результат

svm_cv.best_score_
#найкращі параметри

svm_cv.best_params_
svm_tuned = SVC(kernel="rbf",C=10,gamma=0.0001).fit(X_train,y_train)
svm_tuned
y_pred = svm_tuned.predict(X_test)
cm = confusion_matrix(y_test,y_pred)
cm
print("Our Accuracy is: ", (cm[0][0]+cm[1][1])/(cm[0][0]+cm[1][1]+cm[0][1]+cm[1][0]))
accuracy_score(y_test,y_pred)
print(classification_report(y_test,y_pred))
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ХВОРИХ НА ДІАБЕТ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

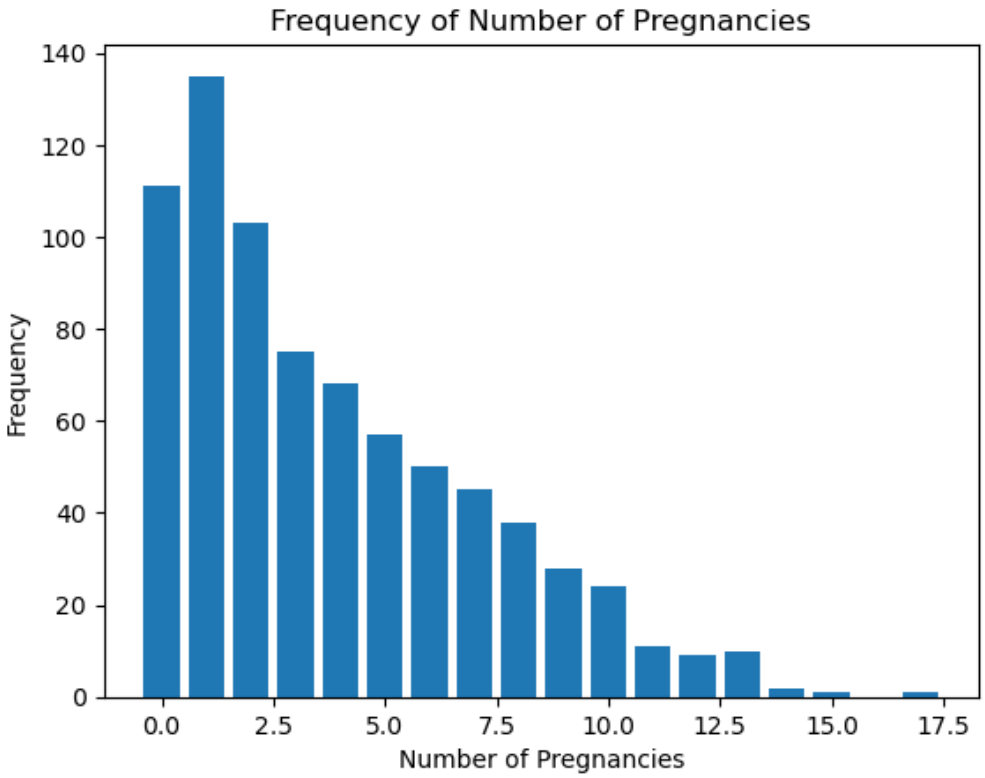

Рисунок Г.1 – Діаграма розподілу кількостей вагітностей

```
In [11]: correlation_plot()
```

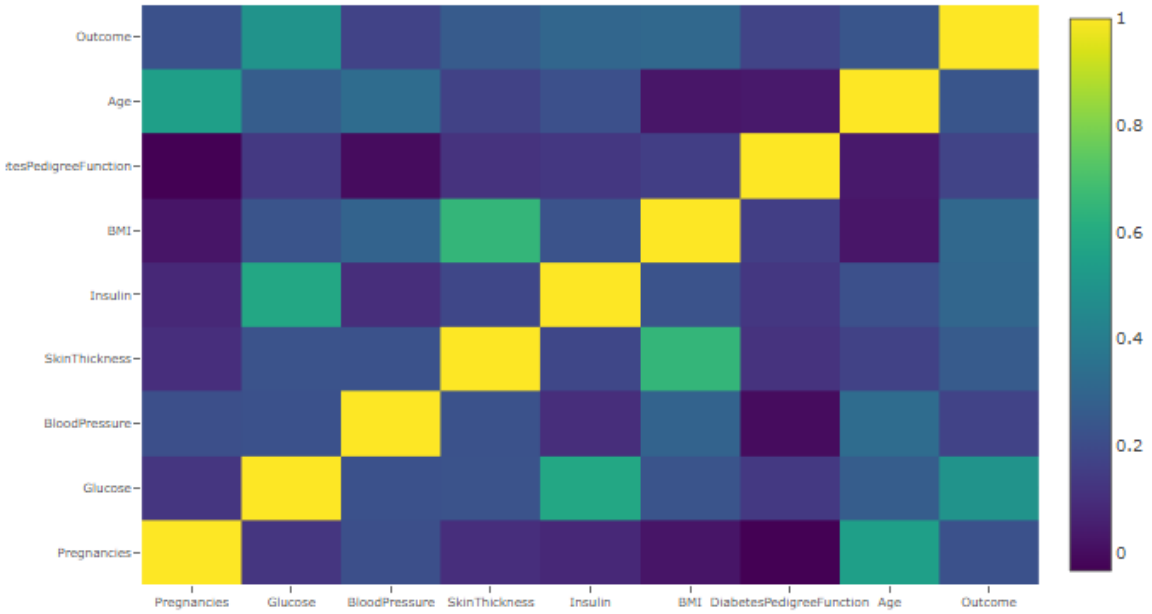

Рисунок Г.2 - Кореляційна матриця

```
In [17]: plot_distribution('Glucose', 0)
```

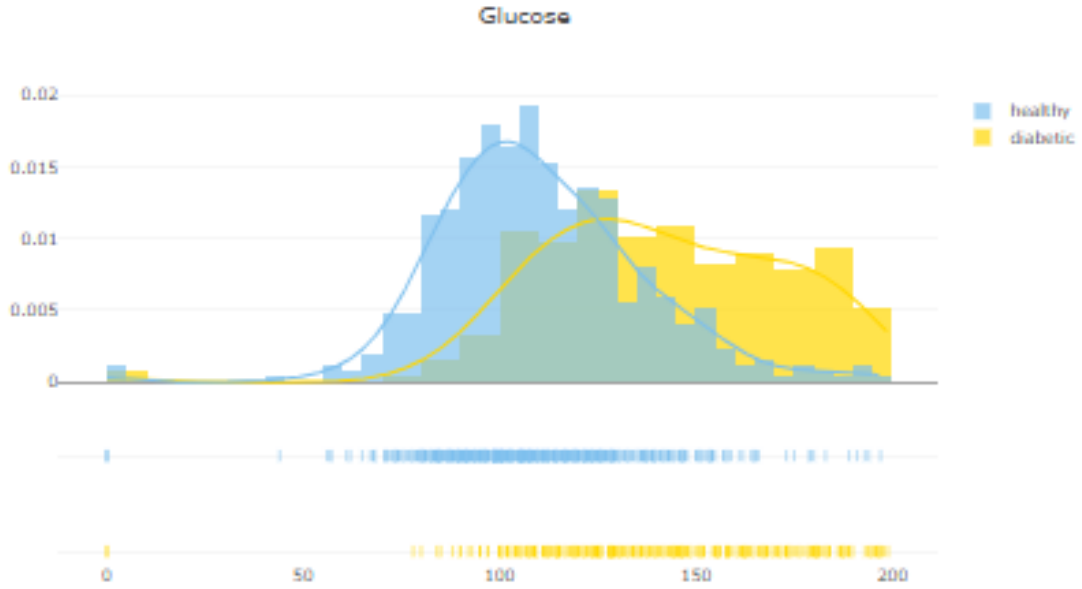

Рисунок Г.3 – Розподіл значень змінної "Glucose"

```
In [34]: plot_feat1_feat2('Glucose','Age')
```

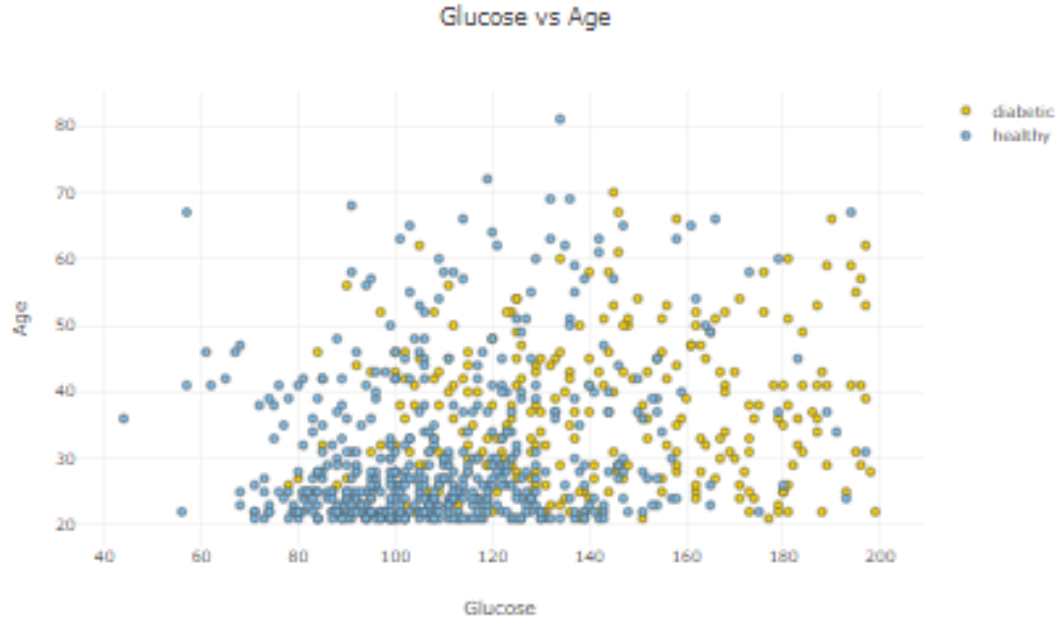

Рисунок Г.4 Відношення між рівнями глюкози в крові ("Glucose") та віком пацієнтів ("Age")

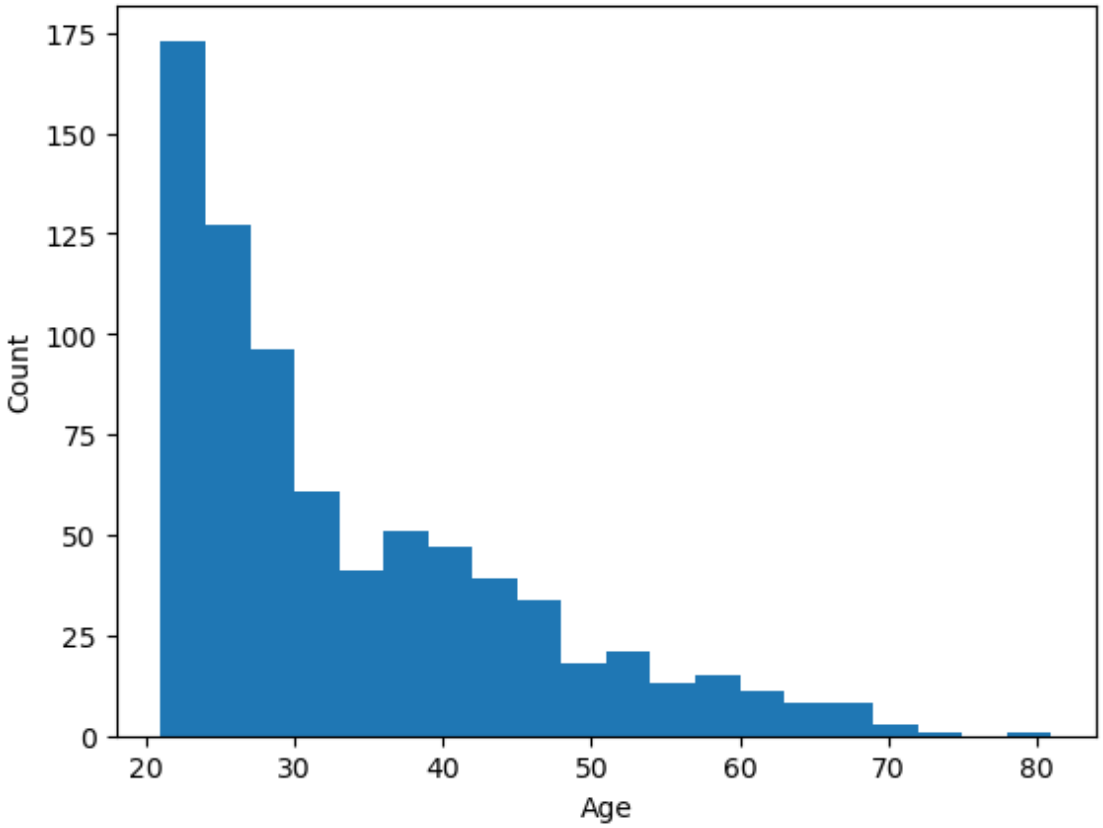

Рисунок Г.5 Зображення гістограми, що показує розподіл віку у наборі даних про діабет.

Назва моделі	Точність прогнозу на навчальній вибірці	Точність прогнозу на тестовій вибірці
SVM	75.5%	76%
KNN	74.1%	74.25%
Forest	77.50%	78.5%

Рисунок Г.6 – Таблиця результатів передбачення