Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра системного аналізу та інформаційних технологій

Бакалаврська дипломна робота на тему:

«ПЕРЕДБАЧЕННЯ СТУПЕНЯ ОЖИРІННЯ МЕТОДАМИ МАШИННОГО НАВЧАННЯ»

Виконав: студент 4 курсу, групи СА-20б
спеціальності 124 – «Системний аналіз»
_____ Андрій ПЯНКЕВИЧ

Керівник: к.т.н., доц. каф. САІТ
_____ Сергій ЖУКОВ
«03» 06 2024 р.

Рецензент: к.т.н., доц.,проф.
_____ Олег КОЛЕСНИЦЬКИЙ
«11» 06 2024 р.

Допущено до захисту
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«04» 06 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 – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 124 – «Системний аналіз»
Освітньо-професійна програма – Системний аналіз

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
"05" 03 2024 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Пянкевичу Андрію Дмитровичу

1. Тема роботи: «Передбачення ступеня ожиріння методами машинного навчання»
керівник роботи: Жуков С.О., к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «11» 03 2024 року № 80
2. Термін подання студентом роботи 31.05.
3. Вихідні дані до роботи:
Набір даних пацієнтів для передбачення стадії ожиріння.
4. Зміст текстової частини:
1) Аналіз проблеми передбачення ступеня ожиріння;
2) Ідентифікація та вибір оптимальної моделі для передбачення ступеня ожиріння;
3) Розроблення інформаційної технології передбачення ступеня ожиріння.
5. Перелік ілюстративного матеріалу:
1) Вигляд датасету;
2) Розподіл цільової змінної;
3) Розподіли ознак датасету;
4) Результати передбачення ступеня ожиріння;
5) Візуалізація важливості ознак для оптимальної моделі.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11.03

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	11.03	31.03	вик
2	Розвідувальний аналіз даних та формування ознак для передбачення ступеня ожиріння	01.04	15.04	вик
3	Тренування моделей машинного навчання	16.04	30.04	вик
4	Вибір оптимальної моделі та передбачення ступеня ожиріння	01.05	15.05	вик
5	Оформлення матеріалів до захисту БДР	16.05	31.05	вик

Студент _____ Андрій ПЯНКЕВИЧ

Керівник роботи _____ Сергій ЖУКОВ

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 78 сторінок формату А4, на яких є 53 рисунків, список використаних джерел містить 20 найменувань.

Робота присвячена передбаченню ступеню ожиріння методами машинного навчання за даними з відкритого джерела Kaggle.

Було проведено аналіз предметної області передбачення ступеню ожиріння. За даними моніторингу стану здоров’я пацієнтів проведено розвідувальний аналіз даних та створено моделі машинного навчання. Відповідно до оцінки точності передбачення моделями машинного навчання обрано оптимальну модель.

Ключові слова: розвідувальний аналіз даних, моделі передбачення, машинне навчання, ожиріння, надмірна вага, передбачення категоріальних даних.

ABSTRACT

The bachelor thesis consists of 78 pages of A4 format, on which there are 53 figures, the list of used sources contains 20 names.

The work is devoted to predicting the degree of obesity using machine learning methods based on data from the open source Kaggle.

An analysis of the subject area of predicting the degree of obesity was carried out. Exploratory data analysis was performed and machine learning models were created based on patient health monitoring data. According to the assessment of the accuracy of prediction by machine learning models, the optimal model was selected.

Keywords: exploratory data analysis, prediction models, machine learning, obesity, overweight, categorical data prediction.

ЗМІСТ

ВСТУП

Актуальність теми. Ожиріння – це хронічне захворювання, яке характеризується надмірним накопиченням жиру в організмі. Це серйозна проблема охорони здоров'я, яка пов'язана з широким спектром хронічних захворювань, включаючи серцево-судинні захворювання, діабет 2 типу, деякі види раку та артрит. За оцінками Всесвітньої організації охорони здоров'я (ВООЗ), у 2016 році 1,9 мільярда дорослих старше 18 років були надмірної ваги, з яких 650 мільйонів були з ожирінням [1].

Ожиріння є однією з найсерйозніших проблем охорони здоров'я у світі. За даними ВООЗ, воно стало п'ятою основною причиною смерті у 2016 році, випередивши діабет та хвороби дихання. Очікується, що до 2030 року ожиріння стане основною причиною смерті, що можна запобігти [2].

Ожиріння поширене у всіх країнах світу, але воно стає все більш серйозною проблемою в країнах, що розвиваються. У 2016 році 57% дорослих з надмірною вагою та 8% дорослих з ожирінням проживали в країнах, що розвиваються [3].

Виявлення ожиріння на ранніх стадіях є критично важливим для успішного управління цим станом. Прогнозування ризику розвитку ожиріння дозволяє виявити пацієнтів, які потребують ретельнішого спостереження та відповідного лікування. Чим раніше буде виявлено ожиріння, тим швидше можна розпочати необхідні терапевтичні заходи та моніторинг для запобігання ускладнень.

Прогнозування ожиріння сприяє розробці індивідуалізованих підходів до терапії та управління станом. Завчасне виявлення ризику ожиріння дає лікарям змогу впроваджувати відповідні заходи для запобігання прогресуванню захворювання та покращення прогнозу.

Сучасний розвиток технологій у сфері машинного навчання та аналізу даних відкриває нові можливості для використання цих методів у

прогнозуванні ступеня ожиріння. Це дозволяє виявляти нові кореляції та патерни у клінічних та біомаркерних даних, що можуть бути корисними для прогнозування ризику та діагностики ступеня ожиріння [4,5].

Мета і задачі дослідження. Метою дослідження є підвищення точності передбачення ступеня ожиріння методами машинного навчання.

Для досягнення поставленої мети необхідно розв’язати такі задачі:

- провести аналіз проблеми передбачення ступеня ожиріння;
- провести розвідувальний аналіз даних та вибір ознак;
- здійснити вибір оптимальної моделі машинного навчання передбачення ступеня ожиріння та навчити її на підготовлених даних;
- здійснити передбачення ступеня ожиріння;

Об’єктом дослідження є процес передбачення ступеня ожиріння методами машинного навчання.

Предметом дослідження є методи системного аналізу даних та алгоритми машинного навчання, які використовувались для передбачення ступеня ожиріння.

Публікації. За результатами даної роботи зроблено доповідь на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2023-2024 рр.) з публікацією тез [6].

1 АНАЛІЗ ПРОБЛЕМИ ПЕРЕДБАЧЕННЯ СТУПЕНЯ ОЖИРІННЯ

1.1 Аналіз предметної області

Ожиріння є однією з найбільш актуальних проблем охорони здоров'я у сучасному світі. За даними Всесвітньої організації охорони здоров'я (ВООЗ), кількість людей з надмірною вагою та ожирінням постійно зростає, що призводить до збільшення випадків захворювань, пов'язаних з надмірною вагою, таких як діабет, серцево-судинні захворювання та деякі види раку. Ожиріння є складним станом, що виникає внаслідок багатьох факторів, включаючи генетичні, поведінкові, екологічні та соціально-економічні. Тому важливо розробити ефективні методи для його передбачення та попередження [2,7].

Ожиріння є складним багатофакторним захворюванням, яке виникає в результаті взаємодії генетичних, метаболічних, поведінкових та екологічних факторів. Основною причиною ожиріння є енергетичний дисбаланс, коли споживання калорій перевищує їх витрати. Цей дисбаланс може бути спричинений неправильним харчуванням, низьким рівнем фізичної активності, генетичними факторами та порушенням метаболізму [7].

Ожиріння — це хронічне захворювання, що характеризується надмірним відкладенням жиру в організмі та високим індексом маси тіла (ІМТ), який розраховується шляхом ділення ваги (кг) на зріст у квадраті ($м^2$). Відповідно до інформаційного бюлетеня про ожиріння за 2021 рік, поширеність ожиріння постійно зростала протягом попередніх 11 років. Отже, багато захворювань, пов’язаних із ожирінням, стали проблемами здоров’я. Діагноз ожиріння важливий, оскільки він є критерієм для початку лікування [8].

ІМТ використовується для визначення ожиріння, оскільки він сильно корелює з жиром в організмі в більшості популяцій і з пов’язаними з ожирінням супутніми захворюваннями, такими як цукровий діабет, гіпертонія

та дисліпідемія. Тому для оцінки ризику ожиріння та супутніх захворювань рекомендовано щорічно вимірювати ІМТ [8,9].

Генетична схильність відіграє важливу роль у розвитку ожиріння. Дослідження показують, що близько 40-70% варіацій у вазі тіла можуть бути пояснені генетичними факторами. Багато генів пов'язані з регуляцією апетиту, метаболізму жирів та витрат енергії [7].

Ожиріння асоціюється з багатьма серйозними медичними станами, включаючи серцево-судинні захворювання, діабет другого типу, рак, гіпертонію та порушення сну. Воно також збільшує ризик смертності та знижує якість життя. Наприклад, згідно з дослідженням, проведеним Американською асоціацією серця, ожиріння збільшує ризик серцево-судинних захворювань на 20% [2].

Серцево-судинні захворювання є однією з провідних причин смертності серед людей з ожирінням. Вони включають інфаркт міокарда, інсульт та інші стани, пов'язані з порушенням кровообігу. Ожиріння сприяє розвитку атеросклерозу, що призводить до звуження артерій і збільшення ризику серцевих нападів та інсультів [2,7].

Ожиріння є загальним фактором ризику діабету 2 типу. Скринінг на цукровий діабет показаний усім пацієнтам з ожирінням. Лікування ожиріння є наріжним каменем у профілактиці та лікуванні діабету 2 типу. Зниження ваги веде до профілактики, контролю та в деяких випадках до ремісії діабету. У цій діяльності розглядається лікування пацієнтів з ожирінням і діабетом, включаючи втручання у спосіб життя, фармакологічне лікування та метаболічну та баріатричну хірургію [10].

Ожиріння має значний вплив на економіку, оскільки збільшує витрати на охорону здоров'я та знижує продуктивність праці. За даними ВООЗ, прямі медичні витрати, пов'язані з лікуванням ожиріння та супутніх захворювань, становлять значну частку витрат на охорону здоров'я у багатьох країнах. Крім

того, непрямі витрати, пов'язані з втратою продуктивності через хворобу та передчасну смертність, також є суттєвими [7].

Ожиріння часто супроводжується соціальною стигматизацією та дискримінацією. Люди з ожирінням можуть стикатися з упередженнями та негативними стереотипами, що призводить до низької самооцінки та депресії. Психологічні проблеми, такі як тривога та депресія, можуть посилюватися через соціальну ізоляцію та дискримінацію на робочому місці та в суспільстві [7].

Профілактика ожиріння включає зміни в способі життя, такі як здорове харчування та регулярна фізична активність. Медичні втручання можуть включати поведінкову терапію, медикаментозне лікування та хірургічні методи. Важливо також розробляти політики та програми, спрямовані на підвищення обізнаності про ризики ожиріння та сприяння здоровому способу життя на рівні громади та держави [7].

Ожиріння є серйозною глобальною проблемою, яка потребує комплексного підходу до її вирішення. Необхідно розробляти та впроваджувати ефективні стратегії профілактики та лікування, враховуючи генетичні, поведінкові та соціально-економічні аспекти цього захворювання. Лише спільні зусилля медичних працівників, науковців, політиків та суспільства можуть допомогти зменшити поширеність ожиріння та його негативні наслідки на здоров'я та економіку.

1.2 Аналіз методів машинного навчання для задачі передбачення ступеня ожиріння

Зростання можливостей інформаційно-комунікаційних технологій, зокрема штучного інтелекту (ШІ) та машинного навчання (МН), відкрило нові горизонти для збирання, обробки та візуалізації даних у сфері охорони здоров'я. Поєднання даних, отриманих за допомогою методів ШІ та МН, з

результатами традиційних клінічних методів дає можливість лікарям приймати більш обґрунтовані рішення щодо діагностики та прогнозування захворювань.

Методи МН вже продемонстрували свою значущість у ранньому прогнозуванні ускладнень цілого ряду захворювань [10-15]. До них належать:

- Цукровий діабет (прогнозування рівня глюкози на короткий термін, що розглядається як задача класифікації або регресії);
- Гіперхолестеринемія;
- Хронічна обструктивна хвороба легень;
- COVID-19;
- Інсульт;
- Хронічна хвороба нирок;
- Рак легенів;
- Розлади сну;
- Серцево-судинні захворювання.

У контексті використання методів машинного навчання (ML) для передбачення ступеня ожиріння важливо розглянути основні аспекти цієї технології. Машинне навчання є підрозділом штучного інтелекту, що займається розробкою алгоритмів, які дозволяють комп'ютерам навчатися на основі даних і приймати рішення без необхідності явного програмування. У випадку з передбаченням ожиріння, алгоритми ML можуть бути використані для аналізу великої кількості даних та виявлення патернів, які дозволяють передбачити ризик розвитку цього стану у конкретної особи.

Серед різних методів машинного навчання, які можуть бути використані для передбачення ступеня ожиріння, можна виділити такі як лінійна регресія, логістична регресія, дерева рішень, випадкові ліси, методи опорних векторів (SVM), та нейронні мережі. Кожен з цих методів має свої переваги та обмеження, і вибір конкретного методу залежить від характеру даних та специфіки задачі [10-15].

Лінійна регресія – цей метод використовується для моделювання залежності між незалежними змінними та ступенем ожиріння. Він є простим у використанні, але може бути недостатньо точним у випадку складних взаємозв'язків між змінними.

Логістична регресія: Цей метод підходить для задач класифікації, де результатом є ймовірність належності до певного класу, наприклад, наявність або відсутність ожиріння.

Дерева рішень: Цей метод дозволяє побудувати модель у вигляді дерева, де кожна вершина відповідає певному рішенню на основі значень вхідних змінних. Дерева рішень є інтуїтивно зрозумілими та легко візуалізованими, але можуть бути схильні до перенавчання.

Випадкові ліси: Цей метод є покращенням дерев рішень і включає побудову великої кількості дерев рішень на різних підмножинах даних, після чого результати цих дерев усереднюються. Випадкові ліси забезпечують високу точність і стійкість до перенавчання.

Методи опорних векторів (SVM): Цей метод використовується для класифікації та регресії і дозволяє знайти оптимальну гіперплощину, яка розділяє класи в багатовимірному просторі. SVM добре працюють з нелінійними даними і забезпечують високу точність.

Нейронні мережі: Цей метод є особливо ефективним для обробки великих обсягів даних та виявлення складних патернів. Нейронні мережі складаються з великої кількості взаємопов'язаних нейронів, що дозволяє їм навчатися на основі вхідних даних та робити високоточні передбачення.

Для ефективного передбачення ступеня ожиріння необхідно мати доступ до великої кількості даних, що включають різноманітні фактори, які можуть впливати на вагу та здоров'я людини. До таких факторів належать демографічні дані (вік, стать, рівень освіти), поведінкові фактори (харчові звички, фізична активність), медичні дані (спадковість, наявність хронічних

захворювань) та екологічні фактори (умови проживання, соціально-економічний статус).

Збір даних може здійснюватися за допомогою різних методів, включаючи опитування, медичні обстеження, носимі пристрої та мобільні додатки. Після збору дані необхідно обробити та підготувати для аналізу, включаючи нормалізацію, очищення від шуму та заповнення пропущених значень.

Виявлено, що ІМТ (зріст, вага), стать, вік, навколишнє середовище, кров'яний тиск і поведінкові ризики, такі як недостатня фізична активність, неправильне харчування, енергетичний дисбаланс тіла та звичка, є головними факторами ризику ожиріння/надмірної ваги [10].

Підбір гіперпараметрів для моделей ґрунтується на висновках, отриманих з розвідувального аналізу зібраних даних та виявлення ознак. Виявлення ознак є процесом використання знань предметної області для вилучення характеристик (ознак) з вихідних даних. Метою цього процесу є покращення якості результатів машинного навчання шляхом додавання додаткових ознак, які можуть сприяти кращому навчальному процесу, ніж необроблені дані.

Нижче наведено деякі типові методи створення корисних ознак:

– Числові перетворення (наприклад, масштабування або отримання дробів);

– Кодування категорійних даних, таких як одноразове або цільове кодування;

– Кластеризація;

– Агрегація згрупованих значень;

– Аналіз головних компонент (для числових даних);

– Створення нових ознак на основі знань предметної області, наприклад, безрозмірні числа в фізиці або аналітичні розрахунки в механіці матеріалів.

Крім того, існує альтернатива до процесу виявлення ознак – використання алгоритмів глибокого навчання. Розробка ознак може бути трудомістким та схильним до помилок процесом, що вимагає доменних знань та часто включає метод проб і помилок. Алгоритми глибокого навчання можуть працювати з великими необробленими наборами даних без необхідності створення ознак. Проте, навіть ці алгоритми потребують ретельної попередньої обробки та очищення даних, а також обрання відповідної архітектури, гіперпараметрів і алгоритмів оптимізації, що є складним і ітеративним процесом.

Основні етапи методології включають:

– Попередня обробка даних з використанням техніки передискретизації синтетичного меншості для збалансування даних, що дозволяє розробляти ефективні моделі класифікації та прогнозування.

– Оцінка важливості ознак на основі кореляції Пірсона, коефіцієнта посилення та випадкового лісу.

– Порівняльна оцінка продуктивності різних моделей машинного навчання з використанням таких показників, як точність (Accuracy) та площа під кривою (AUC).

Існує багато метрик для вимірювання продуктивності класифікаторів або якості прогнозів. Різні галузі мають свої переваги для конкретних метрик, відповідно до своїх цілей. Наприклад, в медицині часто використовуються чутливість і специфічність, тоді як у добуванні інформації більше надають перевагу влучності та повноті. Важливо відрізняти метрики за їх залежністю від поширеності категорій у популяції.

Прогнозування ступеня ожиріння передбачає вивчення характеристик та факторів, що впливають на розвиток цього стану, та застосування методів машинного навчання для побудови моделі прогнозування. Основні етапи включають:

– Збір даних: Збір достатньої кількості даних про пацієнтів з діагнозом ожиріння, включаючи клінічні ознаки, лабораторні показники та можливі фактори ризику.

– Обробка та очищення даних: Включає видалення відсутніх значень, вирівнювання аномалій та перетворення категоріальних ознак у числові.

– Визначення змінних: Вивчення характеристик, що можуть бути пов'язані з ожирінням.

– Вибір моделі: Вибір підходящої моделі машинного навчання, такої як логістична регресія, дерева рішень, випадковий ліс або градієнтний бустинг.

– Тренування моделі: Навчання моделі на тренувальних даних та оцінка її ефективності.

– Оцінка моделі: Оцінка продуктивності моделі на валідаційних даних з використанням відповідних метрик.

– Налаштування моделі: Корекція параметрів моделі для покращення її ефективності.

– Валідація моделі: Перевірка ефективності моделі на тестових даних.

– Інтерпретація результатів: Аналіз важливості ознак для отримання додаткових інсайтів.

– Валідація результатів: Порівняння прогнозів з реальними даними для оцінки практичної цінності моделі [16].

1.3 Аналіз середовища розробки та мови програмування для задачі передбачення ступеня ожиріння

Python — це динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із суворою динамічною типізацією. Розроблена в 1990

році голландським програмістом Гвідо ван Россумом, назва мови походить від популярного британського серіалу 1970-х років «Летючий цирк Монті Пайтона» [4].

Python використовується для різних цілей, включаючи розробку ігор, веб-застосунків і інструментів для різних проектів. Він також широко застосовується в наукових дослідженнях і для розв'язування прикладних завдань.

Тім Пітерс сформулював філософію Python, виділивши кілька ключових принципів:

– Практичність важливіша за бездоганність;
– Зараз краще, ніж ніколи;
– Якщо реалізацію складно пояснити – ідея погана;
– Помилки ніколи не повинні замовчуватися;
– Просте краще, ніж складне;
– Складне краще, ніж заплутане.

Популярні галузі застосування Python включають робототехніку, машинне навчання, штучний інтелект, інженерію даних, автоматизацію та програмування мікроконтролерів. Проте, Python не підходить для всіх задач, наприклад, для розробки драйверів або операційних систем через обмеження в продуктивності та відсутність безпечної системи типів.

Python здобув популярність завдяки своєму використанню в машинному навчанні та аналізі даних. Мова має багатий набір інструментів для лінійної алгебри, обробки сигналів, візуалізації, а також різноманітних математичних і статистичних підходів. Python продовжує зростати в популярності серед розробників, що видно з його високих позицій у рейтингах мов програмування [5].

Одним з недоліків Python є його швидкість виконання коду. Оскільки це не компільована мова, код спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором. Це робить програми на Python

повільнішими в порівнянні з мовами, як-от C. Проте сучасні комп'ютери мають настільки високу обчислювальну потужність, що для більшості програм важливіша швидкість розробки, а не виконання. Код на Python зазвичай пишеться швидше, і він може доповнюватися модулями на C або C++ для виконання інтенсивних обчислень.

Важливо також відзначити значення бібліотек Python для машинного навчання. Мова пропонує широкий спектр бібліотек для збирання, обробки даних, навчання моделей і їх візуалізації. Ці бібліотеки зменшують час і зусилля розробників, а підтримка багатозадачності дозволяє ефективно обробляти великі обсяги даних. Це робить Python ідеальним для задач машинного навчання, які потребують високої продуктивності.

Для нашої задачі в медичній сфері добре підходять бібліотеки Python, такі як scikit-learn, TensorFlow і PyTorch. Зокрема, scikit-learn пропонує широкий спектр алгоритмів машинного навчання, які добре підходять для медичного прогнозування, наприклад, для діагностики захворювань, прогнозування ризиків і аналізу зображень. Алгоритми класифікації, такі як Logistic Regression, Decision Trees та Random Forest, можуть прогнозувати ймовірність захворювання пацієнта на основі даних про симптоми та історію хвороби. Алгоритми прогнозування, як-от «регресія Кокса» та «регресія градієнтного підсилення», використовуються для прогнозування ризику розвитку захворювань, таких як інсульт або серцевий напад, а в нашому випадку — рак легень.

Підсумовуючи, можна сказати, що Python зі своєю простотою, гнучкістю та широким спектром бібліотек чудово підходить для вирішення задачі прогнозування раку легенів у пацієнтів.

Дослідження було вирішено проводити на платформі Kaggle, яка надає зручний редактор Jupyter Notebook для швидкого моделювання моделей та візуалізації даних. Велика перевага Kaggle — це бібліотеки наборів даних, які користувачі можуть використовувати для тренування та оцінювання моделей

машинного навчання [6]. Ми використовували один із таких наборів даних для нашого дослідження.

1.4 Висновки

В першому розділі проведено аналіз предметної області. Аналіз предметної області показав, що машинне навчання має великий потенціал для передбачення ступеня ожиріння. Використання сучасних алгоритмів ML дозволяє аналізувати великі обсяги даних та виявляти приховані патерни, що може сприяти розробці ефективних стратегій попередження та лікування ожиріння. Подальші дослідження у цій сфері можуть включати розробку більш точних моделей, інтеграцію додаткових джерел даних та створення інтелектуальних систем підтримки прийняття рішень для медичних фахівців.

Проаналізовано методи машинного навчання для передбачення ступеня ожиріння. Обрано декілька актуальних моделей машинного навчання часто використовуваних в передбаченні медичних діагнозів.

2 ІДЕНТИФІКАЦІЯ ТА ВИБІР ОПТИМАЛЬНОЇ МОДЕЛІ ДЛЯ ПЕРЕДБАЧЕННЯ СТУПЕНЯ ОЖИРІННЯ

2.1 Розвідувальний аналіз даних

Розвідувальний аналіз даних (EDA) – це метод, який використовують фахівці з даних для розуміння набору даних перед тим, як почати його моделювати. Іноді EDA називають дослідженням даних. Основна мета EDA – виявити характеристики набору даних. Проведення EDA допомагає аналітикам даних робити прогнози та висновки щодо даних. Зазвичай EDA включає візуалізацію даних, зокрема створення графіків, таких як гістограми, діаграми розсіювання та квадратні діаграми. EDA застосовується для знаходження зв'язків між змінними, коли немає (або недостатньо) попередніх уявлень про природу цих зв'язків. Зазвичай під час розвідувального аналізу враховується і порівнюється велика кількість змінних, а для пошуку закономірностей використовують різні методи. Основні методи розвідувального аналізу даних:

- Виявлення основних структур;
- Вибір найвагоміших змінних;
- Виявлення відхилень та аномалій;
- Перевірка основних гіпотез;
- Розробка початкових моделей;

Для бакалаврської дипломної роботи використано датасет платформи Kaggle «Obesity or CVD risk (Classify/Regressor/Cluster)» [17]. Дані мають 17 ознак та 2111 записів, записи позначені змінною класу NObesity (рівень ожиріння), що дозволяє класифікувати/групувати дані на такі класи:

- недостатня вага,
- нормальна вага,
- рівень надмірної ваги I,
- рівень надмірної ваги II,

- тип ожиріння I,
- тип ожиріння II
- тип ожиріння III.

Серед атрибутів можна виділити атрибути, пов’язані зі звичками харчування:

- часте споживання висококалорійної їжі (FAVC);
- частота споживання овочів (FCVC);
- кількість основних прийомів їжі (NCP);
- споживання їжі між прийомами їжі (CAEC);
- щоденне споживання води (CH20);
- споживання алкоголю (CALC).

Атрибути, пов’язані з фізичним станом:

- моніторинг споживання калорій (SCC);
- частота фізичної активності (FAF);
- час використання технологічних пристроїв (TUE);
- використовуваний транспорт (MTRANS).

Серед інших атрибутів:

- стать (Gender);
- вік (Age);
- зріст (Height);
- вага (Weight);
- член сім'ї страждав або страждає надмірною вагою (FHWO);
- Курець чи ні (Smoke);

Для початку потрібно розглянути дані, які представлені в датасеті «Obesity or CVD risk (Classify/Regressor/Cluster)» (рис. 2.1 – 2.2) [17].

```
data
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	S
0	Female	21.000000	1.620000	64.000000	yes	no	2.0	3.0	Sometimes	no	2.000000	
1	Female	21.000000	1.520000	56.000000	yes	no	3.0	3.0	Sometimes	yes	3.000000	
2	Male	23.000000	1.800000	77.000000	yes	no	2.0	3.0	Sometimes	no	2.000000	
3	Male	27.000000	1.800000	87.000000	no	no	3.0	3.0	Sometimes	no	2.000000	
4	Male	22.000000	1.780000	89.800000	no	no	2.0	1.0	Sometimes	no	2.000000	
...	...	...	...	...	...	...	...	...	...	...	...	
2106	Female	20.976842	1.710730	131.408528	yes	yes	3.0	3.0	Sometimes	no	1.728139	
2107	Female	21.982942	1.748584	133.742943	yes	yes	3.0	3.0	Sometimes	no	2.005130	
2108	Female	22.524036	1.752206	133.689352	yes	yes	3.0	3.0	Sometimes	no	2.054193	
2109	Female	24.361936	1.739450	133.346641	yes	yes	3.0	3.0	Sometimes	no	2.852339	
2110	Female	23.664709	1.738836	133.472641	yes	yes	3.0	3.0	Sometimes	no	2.863513	

2111 rows × 17 columns

Рисунок 2.1 – Перша частина датасету

```
data
```

ght	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no	Public_Transportation	Normal_Weight
yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometimes	Public_Transportation	Normal_Weight
yes	no	2.0	3.0	Sometimes	no	2.000000	no	2.000000	1.000000	Frequently	Public_Transportation	Normal_Weight
no	no	3.0	3.0	Sometimes	no	2.000000	no	2.000000	0.000000	Frequently	Walking	Overweight_Level_I
no	no	2.0	1.0	Sometimes	no	2.000000	no	0.000000	0.000000	Sometimes	Public_Transportation	Overweight_Level_II
...	...	...	...	...	...	...	...	...	...	...	...	...
yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometimes	Public_Transportation	Obesity_Type_III
yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometimes	Public_Transportation	Obesity_Type_III
yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometimes	Public_Transportation	Obesity_Type_III
yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometimes	Public_Transportation	Obesity_Type_III
yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometimes	Public_Transportation	Obesity_Type_III

Рисунок 2.2 – Друга частина датасету

Перевіримо дані на пропущені значення, щоб враховувати їх відсутність надалі та при необхідності почистити параметри, які мають занадто багато таких значень й через що не будуть представляти цінності для прогнозування (рис.2.3).

```
# Check for missing values
print(data.isnull().sum())
```

```
Gender                            0
Age                               0
Height                            0
Weight                            0
family_history_with_overweight    0
FAVC                              0
FCVC                              0
NCP                               0
CAEC                              0
SMOKE                             0
CH2O                              0
SCC                               0
FAF                               0
TUE                               0
CALC                              0
MTRANS                            0
NObeyesdad                        0
dtype: int64
```

Рисунок 2.3 – Перевірка на пропущені значення

Як видно з рисунку 2.3 датасет не містить пропущених значень. Виведемо загальну інформацію про датасет (рис.2.4).

```
# Basic statistics of the dataset
print(data.info())
print(data.describe(include='number'))
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Gender                          2111 non-null   object
 1   Age                             2111 non-null   float64
 2   Height                          2111 non-null   float64
 3   Weight                          2111 non-null   float64
 4   family_history_with_overweight  2111 non-null   object
 5   FAVC                            2111 non-null   object
 6   FCVC                            2111 non-null   float64
 7   NCP                             2111 non-null   float64
 8   CAEC                            2111 non-null   object
 9   SMOKE                           2111 non-null   object
 10  CH2O                            2111 non-null   float64
 11  SCC                             2111 non-null   object
 12  FAF                             2111 non-null   float64
 13  TUE                             2111 non-null   float64
 14  CALC                            2111 non-null   object
 15  MTRANS                          2111 non-null   object
 16  NObeyesdad                      2111 non-null   object
```

Рисунок 2.4 – Загальна інформація датасету

За результатами функції info() можна зробити такі висновки, що у даному наборі даних відсутні пусті значення, тобто усі наявні параметри мають 100% заповнення для кожного пацієнта. Також функція показала кількість параметрів, тип даних для кожного параметру та кількість параметрів з унікальними типами даних.

Виведемо базові статистики числових ознак датасету, використавши функцію describe() (рис.2.5).

	Age	Height	Weight	FCVC	NCP	\
count	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	
mean	24.312600	1.701677	86.586058	2.419043	2.685628	
std	6.345968	0.093305	26.191172	0.533927	0.778039	
min	14.000000	1.450000	39.000000	1.000000	1.000000	
25%	19.947192	1.630000	65.473343	2.000000	2.658738	
50%	22.777890	1.700499	83.000000	2.385502	3.000000	
75%	26.000000	1.768464	107.430682	3.000000	3.000000	
max	61.000000	1.980000	173.000000	3.000000	4.000000	

	CH2O	FAF	TUE
count	2111.000000	2111.000000	2111.000000
mean	2.008011	1.010298	0.657866
std	0.612953	0.850592	0.608927
min	1.000000	0.000000	0.000000
25%	1.584812	0.124505	0.000000
50%	2.000000	1.000000	0.625350
75%	2.477420	1.666678	1.000000
max	3.000000	3.000000	2.000000

Рисунок 2.5 – Базові статистики числових ознак датасету

На прикладі віку та зросту зробимо аналіз результатів. З рисунку можемо зробити висновок, що мінімальний вік пацієнта у даному наборі становить 14 років, а максимальний 61 роки, мінімальний зріст пацієнта становить 145 см, а максимальний 197 см. Аналогічно проаналізувавши решту ознак, можна зробити висновок про відсутність аномалій.

Виведемо унікальні значення цільової змінної (рис. 2.6).

```
# Перевірка унікальних значень цільової змінної
print(data['NObeyesdad'].unique())
```

```
['Overweight_Level_II' 'Normal_Weight' 'Insufficient_Weight'
 'Obesity_Type_III' 'Obesity_Type_II' 'Overweight_Level_I'
 'Obesity_Type_I']
```

Рисунок 2.6 – Унікальні значення цільової змінної

Як бачимо, у нас 7 унікальних значень цільової змінної. Побудуємо розподіл цільової змінної (рис. 2.7).

```python
# Побудова розподілу цільової змінної
plt.figure(figsize=(10, 6))
sns.countplot(x='NObeyesdad', data=data)
plt.title('Розподіл рівнів ожиріння')
plt.xlabel('Рівні ожиріння')
plt.ylabel('Кількість')
plt.xticks(rotation=45)
plt.show()
```

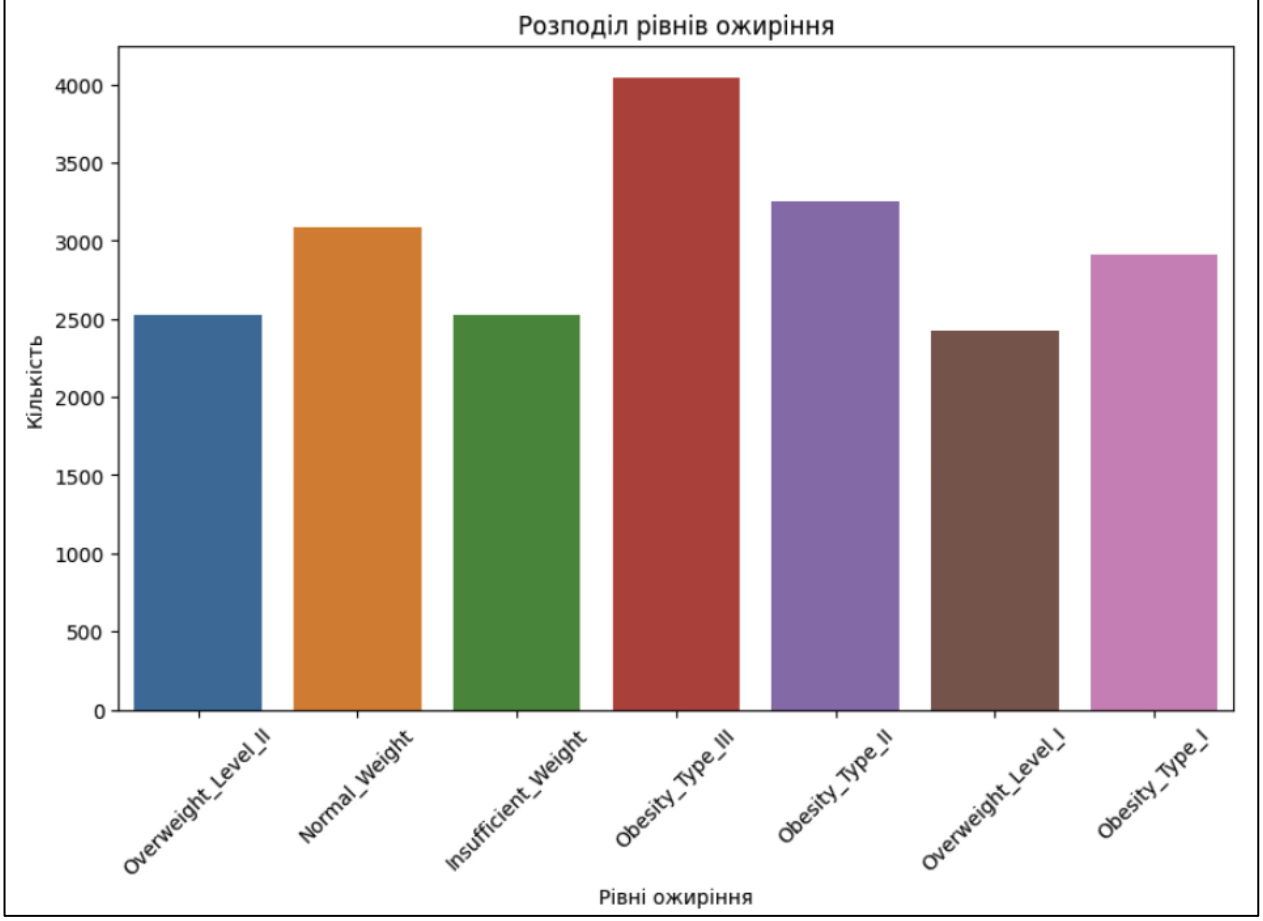

Рисунок 2.7 – Розподіл цільової змінної

Як бачимо, розподіл цільової змінної доволі рівномірний, що має позитивно вплинути на тренування моделей.

Побудуємо розподіл ознаки віку (Age) (рис. 2.8).

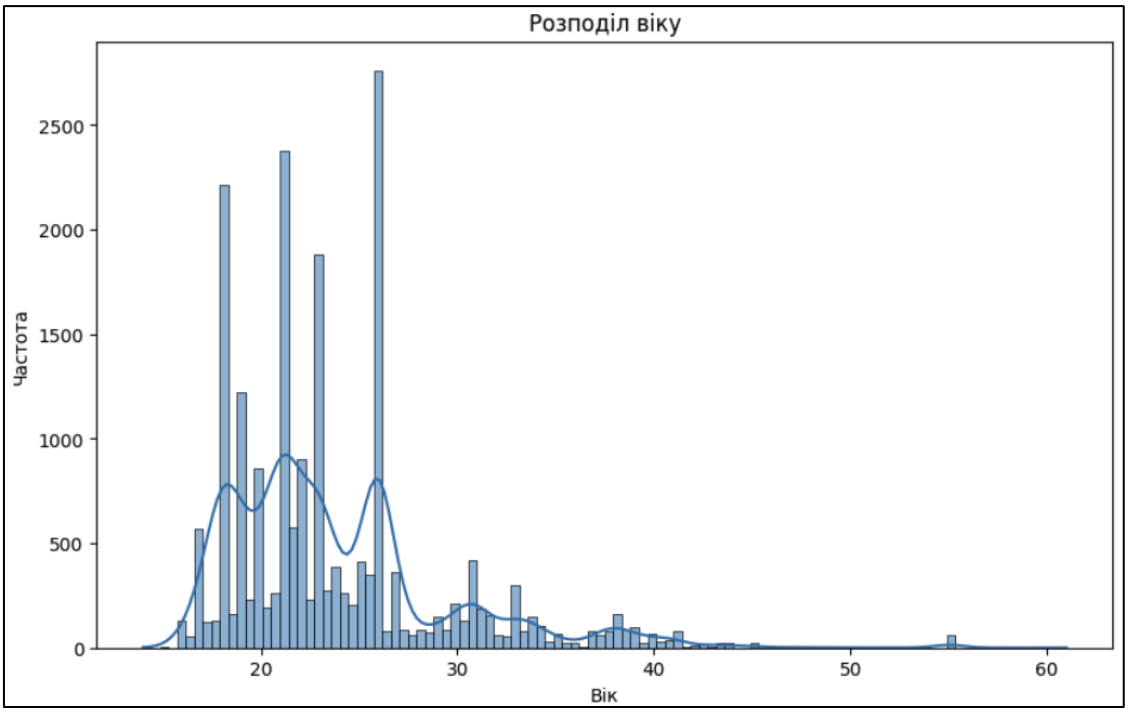

Рисунок 2.8 – Розподіл ознаки віку (Age)

Як бачимо, більшість людей мають вік між 18 і 27 роками. Також відмічаємо, що існує довгий хвіст до старшого віку, що може впливати на прогнози, якщо не врахувати належним чином.

Побудуємо розподіл ознаки стать (Gender) (рис. 2.9).

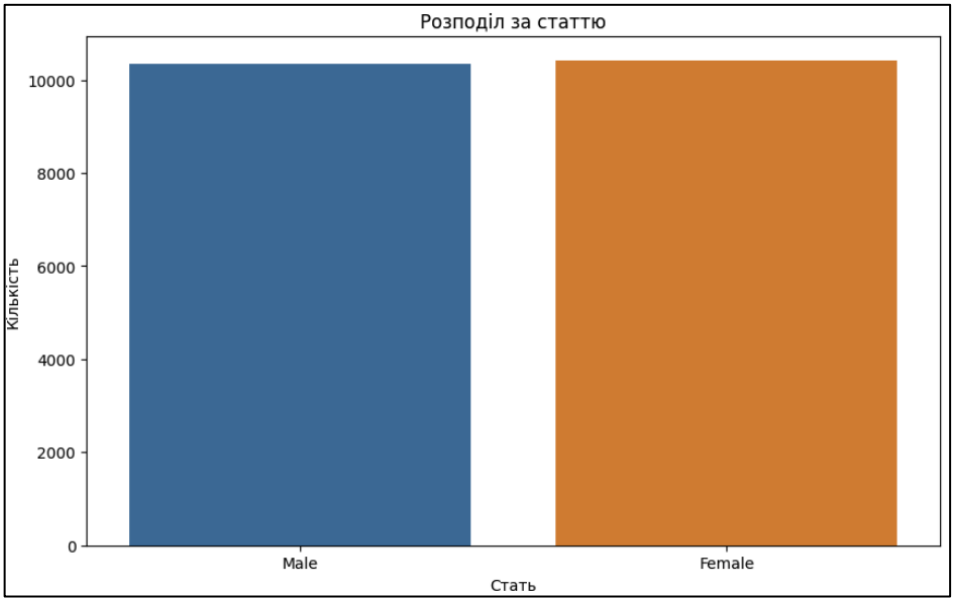

Рисунок 2.9 – Розподіл ознаки стать (Gender)

Як бачимо, стать розподілена майже порівну. Побудуємо розподіл ознаки зріст (Height) (рис. 2.10).

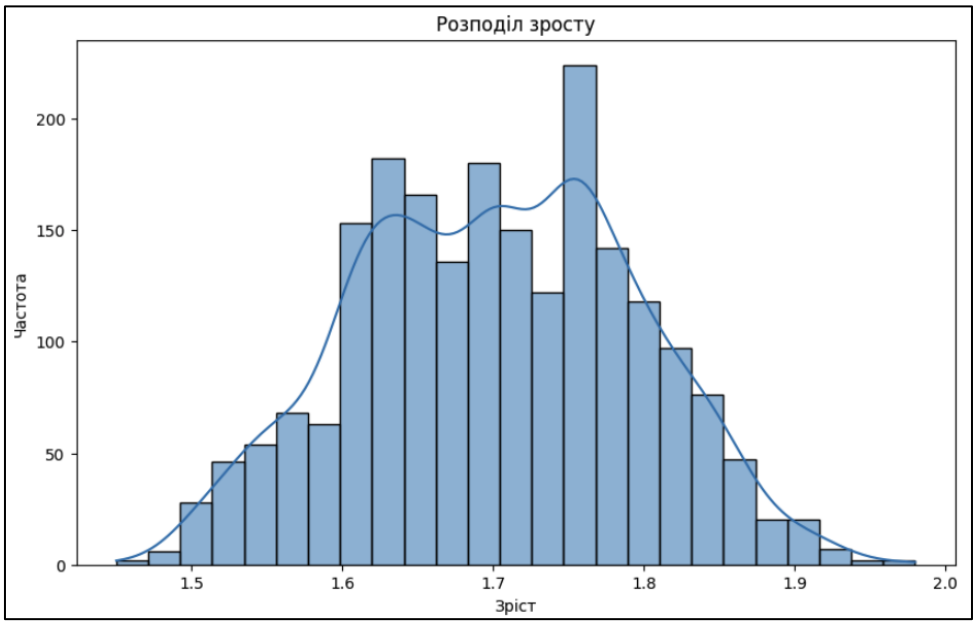

Рисунок 2.10 – Розподіл ознаки зріст (Height)

Як бачимо, більшість людей мають зріст між 160 см і 185 см і розподіл схожий на нормальний. Також можна відмітити відсутність аномальних значень. Побудуємо розподіл ознаки вага (Weight) (рис. 2.11).

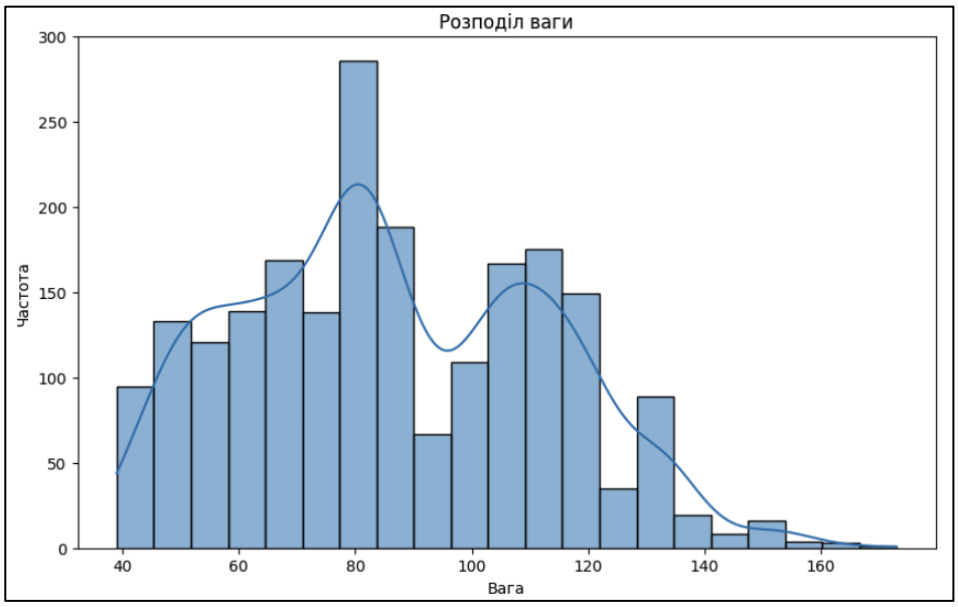

Рисунок 2.11 – Розподіл ознаки вага (Weight)

Як бачимо, вага доволі рівномірно розподілена між значеннями 40 кг і 130 кг. Також можна відмітити відсутність аномальних значень. Побудуємо розподіл ознаки «частота споживання овочів» (FCVC) (рис. 2.12).

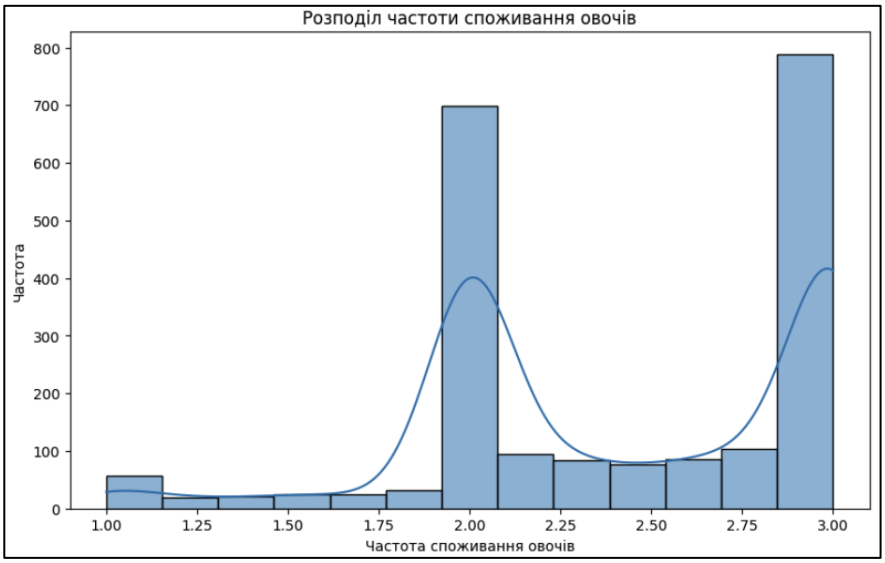

Рисунок 2.12 – Розподіл ознаки «частота споживання овочів» (FCVC)

Як бачимо, найбільш популярним є споживання овочів 2-3 рази на день, що цілком природньо. Також можна відмітити відсутність аномальних значень. Побудуємо розподіл ознаки «кількість основних прийомів їжі» (NCP) (рис. 2.13).

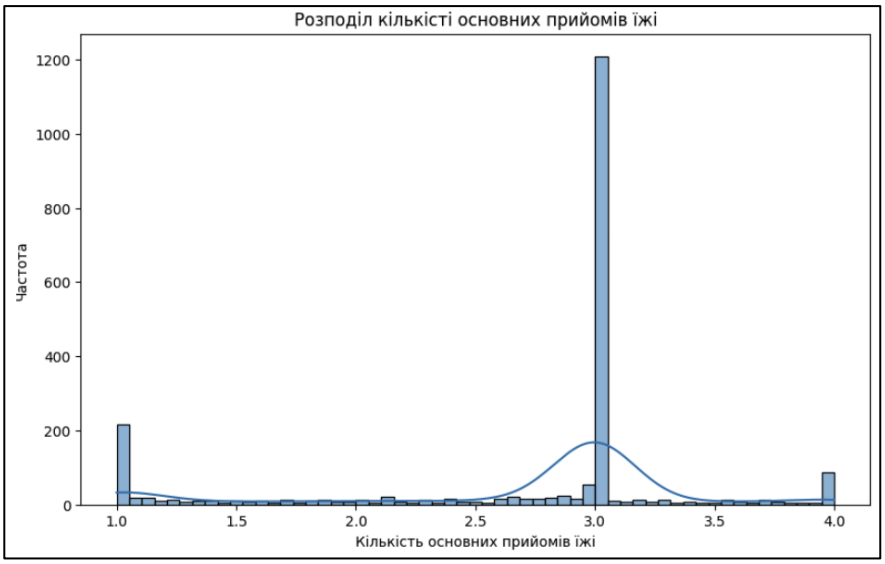

Рисунок 2.13 – Розподіл ознаки «кількість основних прийомів їжі» (NCP)

Як бачимо, найбільш популярним значенням є 3 основних прийомів їжі на день, що цілком природньо. Також можна відмітити відсутність аномальних значень. Побудуємо розподіл ознаки «щоденне споживання води» (CH20) (рис. 2.14).

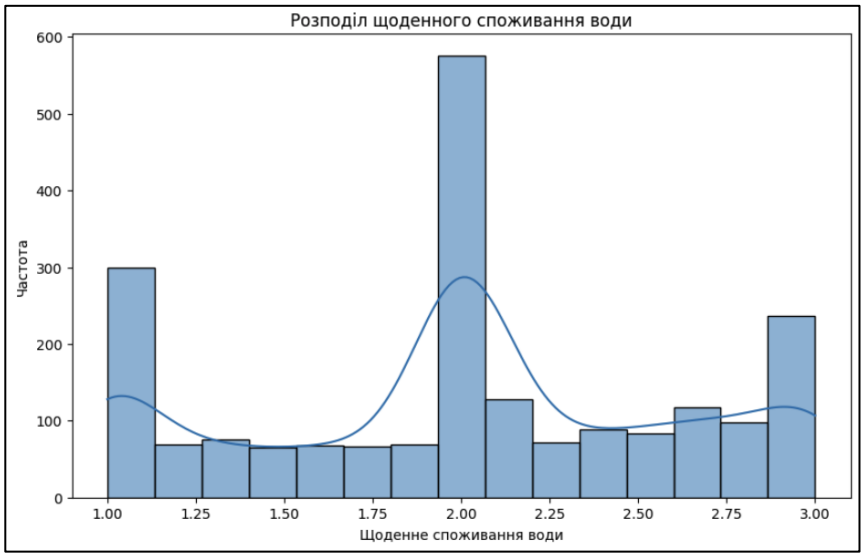

Рисунок 2.13 – Розподіл ознаки «щоденне споживання води» (CH20)

Як бачимо, найбільш популярним значенням є 2л споживання води на день, що цілком природньо. Також можна відмітити відсутність аномальних значень. Побудуємо розподіл ознаки «час використання технологічних пристроїв» (TUE) (рис. 2.14).

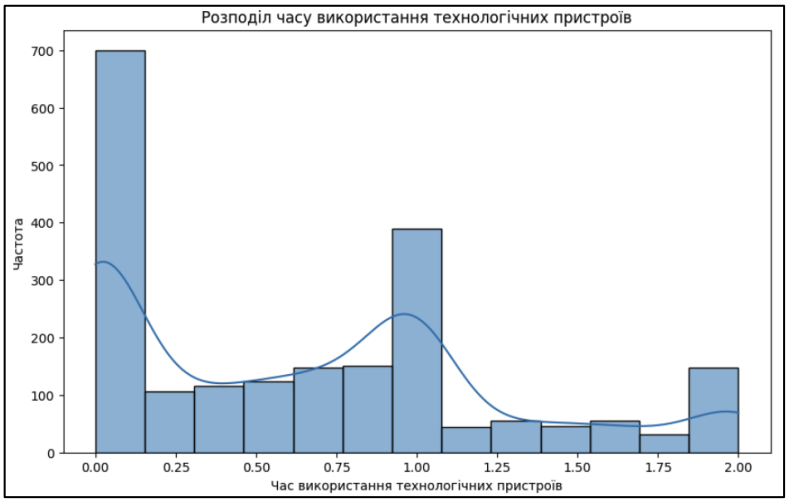

Рисунок 2.14 – Розподіл ознаки «час використання технологічних пристроїв» (TUE)

Як бачимо, найбільш популярним значенням часу використання технологічних пристроїв є 0 хв та 1 година на день, що цілком природньо. Також можна відмітити відсутність аномальних значень. Побудуємо розподіл ознаки «частота фізичної активності» (FAF) (рис. 2.15).

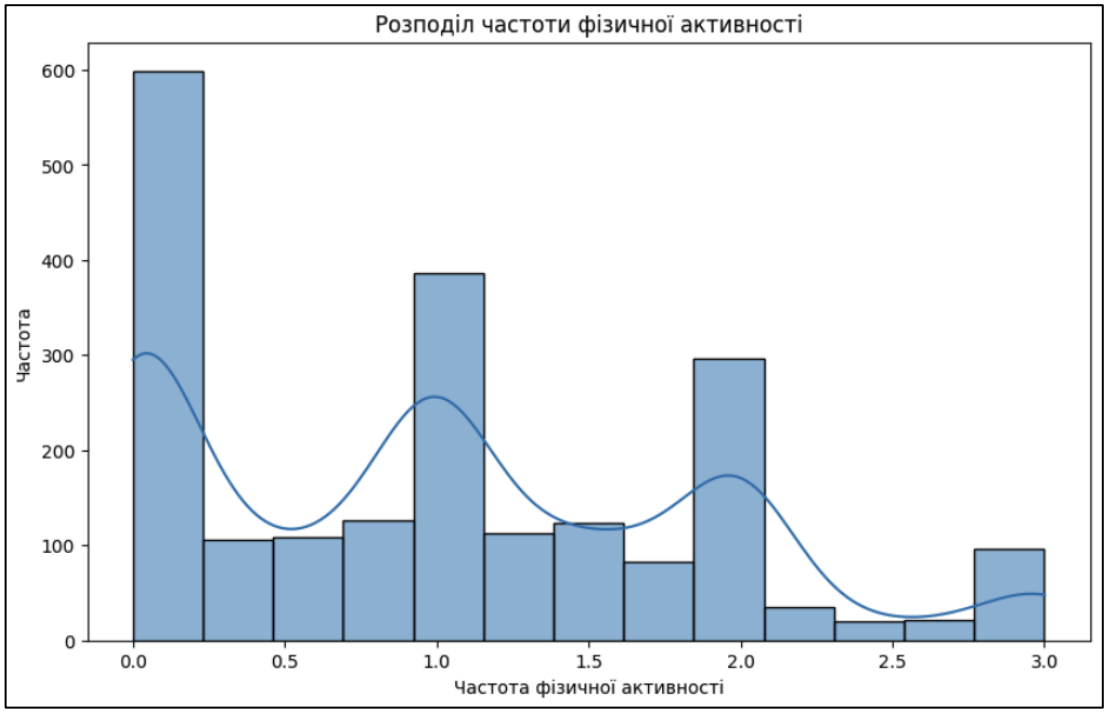

Рисунок 2.15 – Розподіл ознаки «частота фізичної активності» (FAF)

Як бачимо, найбільш популярним значенням частоти фізичної активності є 0, 1 та 2 рази на день, що цілком природньо. Також можна відмітити відсутність аномальних значень.

Побудуємо розподіли усіх не числових ознак. Побудуємо розподіл ознаки «член сім'ї страждав або страждає надмірною вагою» (FHWO) (рис. 2.16).

Рисунок 2.16 – Розподіл ознаки FHWO

Як бачимо, в більшості випадків член сім'ї страждав або страждає надмірною вагою. Побудуємо розподіл ознаки «часте споживання висококалорійної їжі» (FAVC) (рис. 2.17).

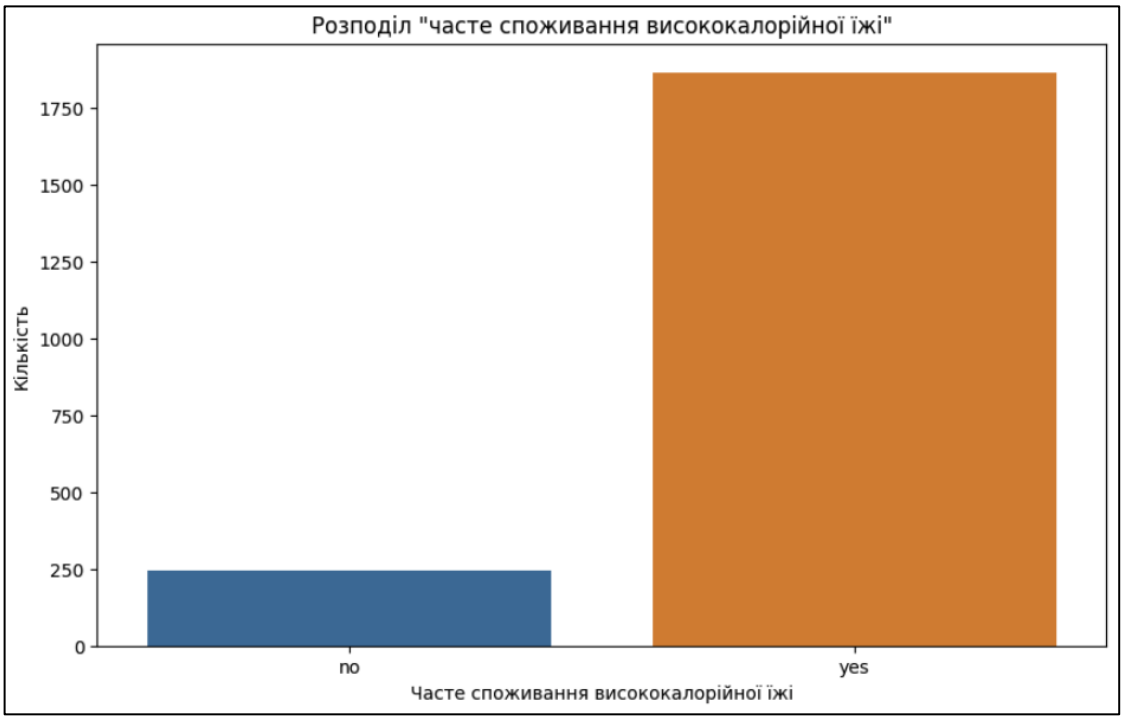

Рисунок 2.17 – Розподіл ознаки FAVC

Як бачимо, в більшості випадків присутнє часте споживання висококалорійної їжі, що і пояснює зайву вагу у опитуваних. Побудуємо розподіл ознаки «споживання їжі між прийомами їжі» (CAEC) (рис. 2.18).

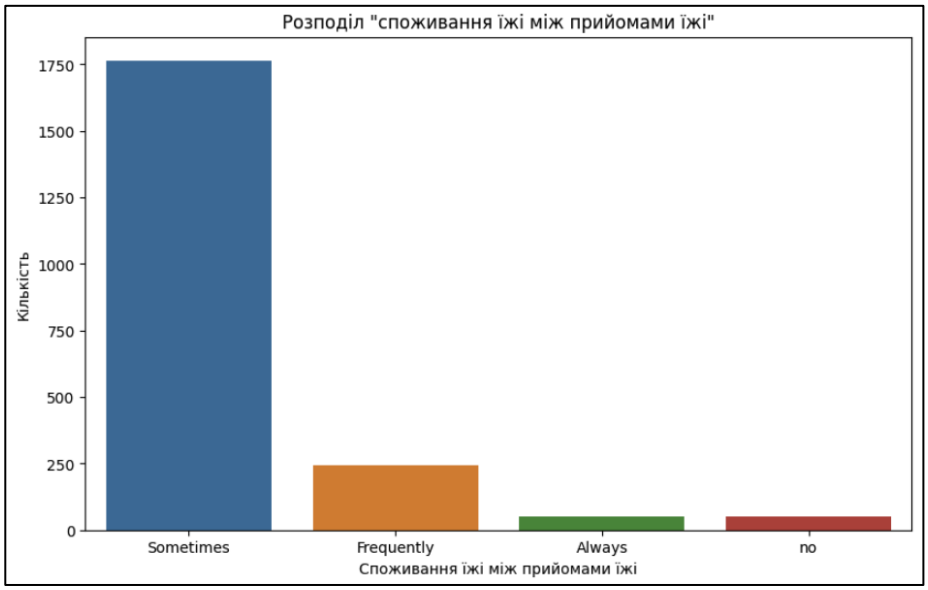

Рисунок 2.18 – Розподіл ознаки CAEC

Як бачимо, в більшості випадків іноді відбувається споживання їжі між прийомами їжі, а решта варіантів зустрічається зрідка, що теж пояснює зайву вагу у більшості опитуваних. Побудуємо розподіл ознаки «курець чи ні» (SMOKE) (рис. 2.19).

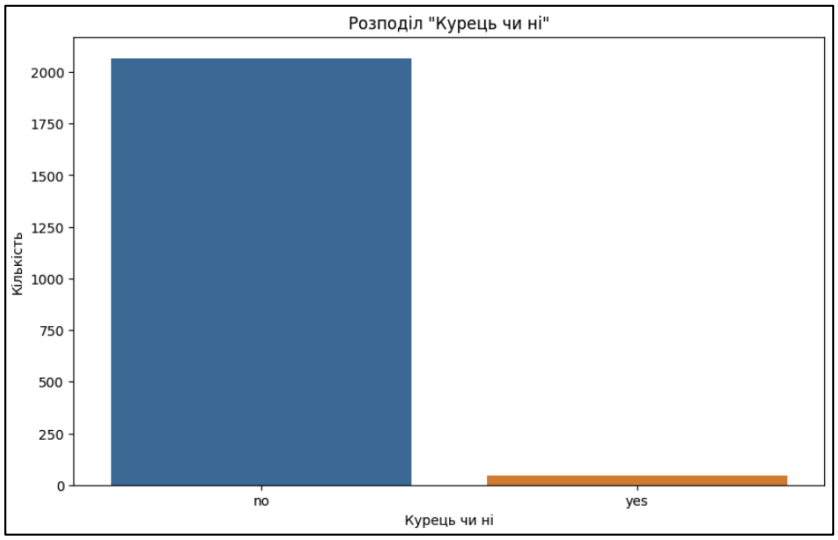

Рисунок 2.19 – Розподіл ознаки SMOKE

Як бачимо, більшість опитуваних не курять. Побудуємо розподіл ознаки «моніторинг споживання калорій» (SCC) (рис. 2.20).

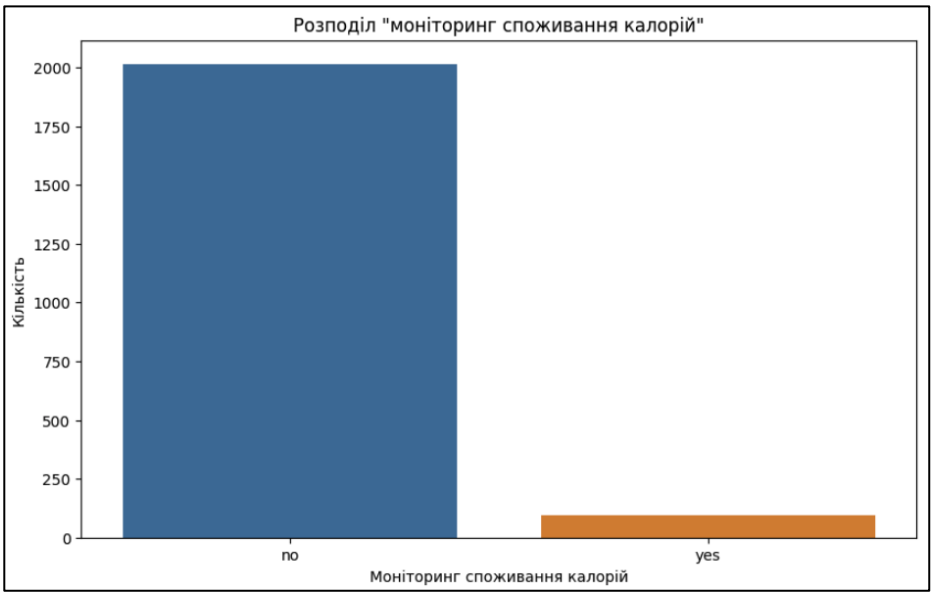

Рисунок 2.20 – Розподіл ознаки SCC

Як бачимо, в більшості випадків опитувані не слідкували за споживанням калорій, що пояснює зайву вагу у більшості опитуваних. Побудуємо розподіл ознаки «споживання алкоголю» (CALC) (рис. 2.21).

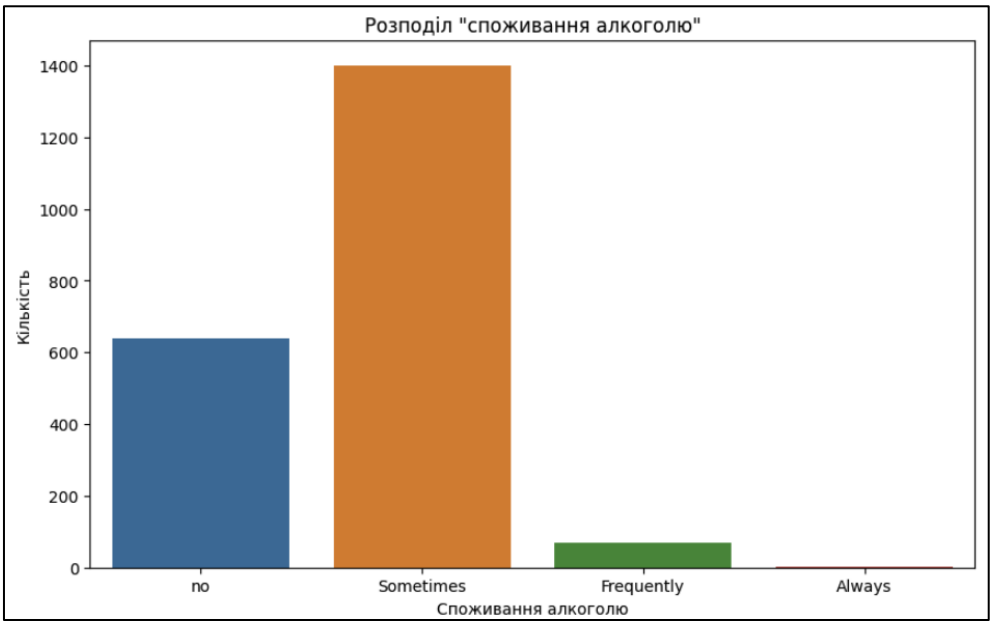

Рисунок 2.21 – Розподіл ознаки SCC

Як бачимо, в доволі велика кількість іноді або часто вживали алкоголь. Побудуємо розподіл ознаки «використовуваний транспорт» (MTRANS) (рис. 2.22).

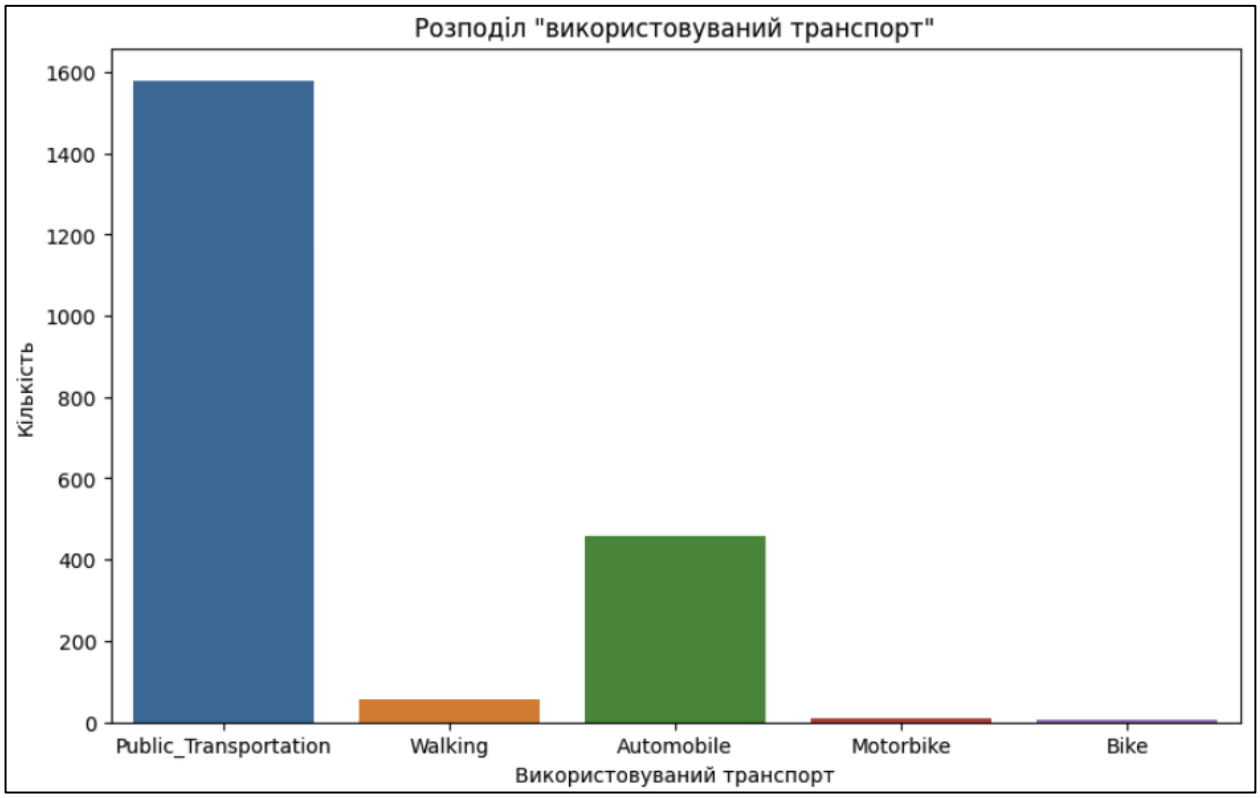

Рисунок 2.22 – Розподіл ознаки MTRANS

Як бачимо з рисунку, дуже мала кількість ходить пішки, що негативно відображається на вазі.

Побудуємо матрицю кореляцій (рис. 2.23).

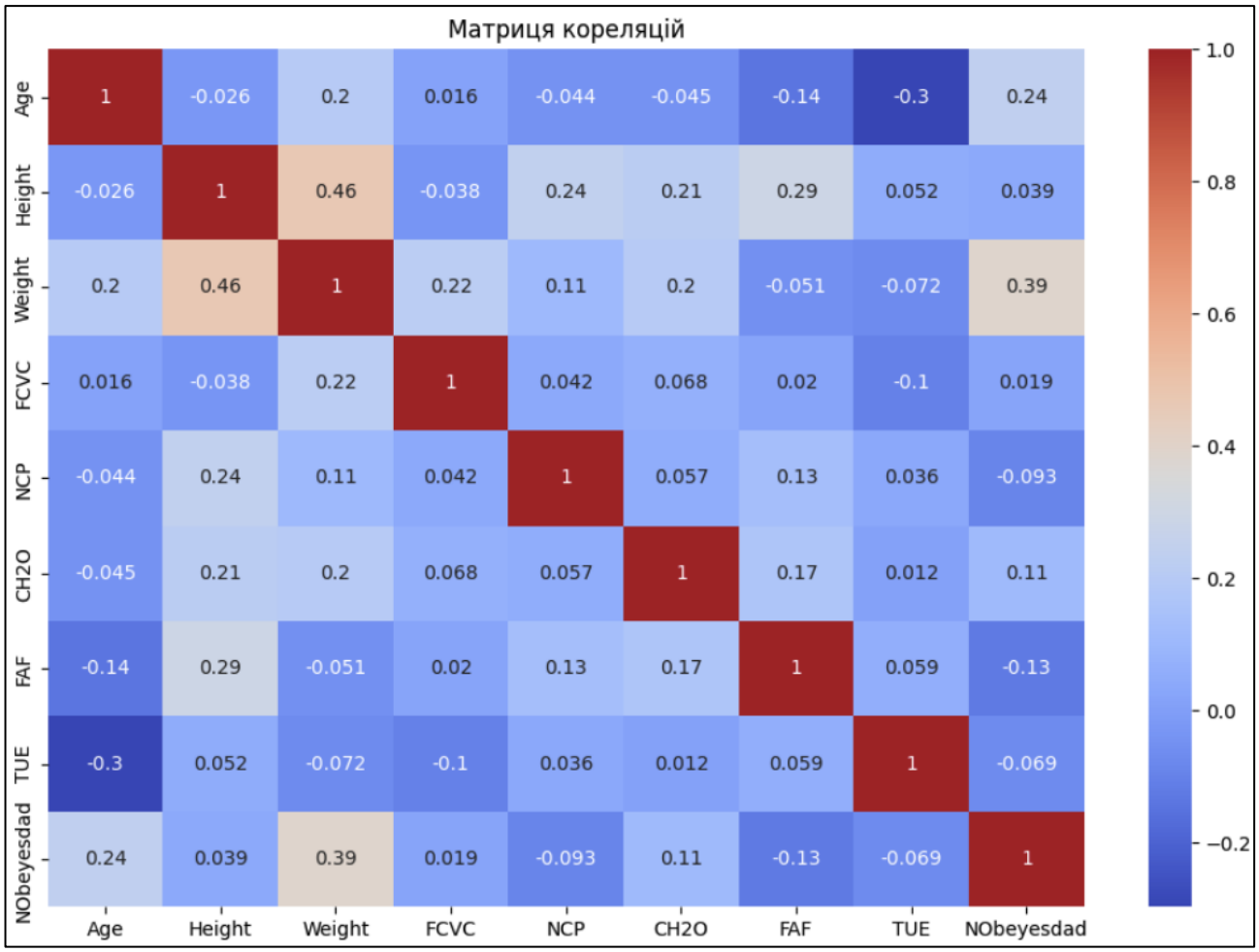

Рисунок 2.23 – Матриця кореляцій

Як можна помітити з матриці кореляцій, найбільша кореляція з цільовою ознакою мають ознаки Weight та Age.

В попередньому розділі було описано, що є висока кореляція між рівнем ожиріння і індексом маси тіла. Додамо цю ознаку в наш датасет і проведемо аналіз (рис. 2.24, 2.25).

ІМТ (Індекс маси тіла) або BMI (Body Mass Index) є показником, який використовують для оцінки відповідності ваги людини до її зросту. Він розраховується як вага (в кілограмах) поділена на квадрат зросту (в метрах). ІМТ є загальноприйнятим інструментом для класифікації недостатньої, нормальної, надлишкової ваги та ожиріння у дорослих.

```
data['BMI'] = data['Weight'] / (data['Height'] ** 2)

# Boxplot IMT проти рівня ожиріння
plt.figure(figsize=(14, 8))
sns.boxplot(x='NObeyesdad', y='BMI', data=data)
plt.title('IMT проти рівня ожиріння')
plt.xlabel('Рівень ожиріння')
plt.ylabel('IMT')
plt.xticks(rotation=45)
plt.show()
```

Рисунок 2.24 – Код додавання ознаки IMT та її візуалізації

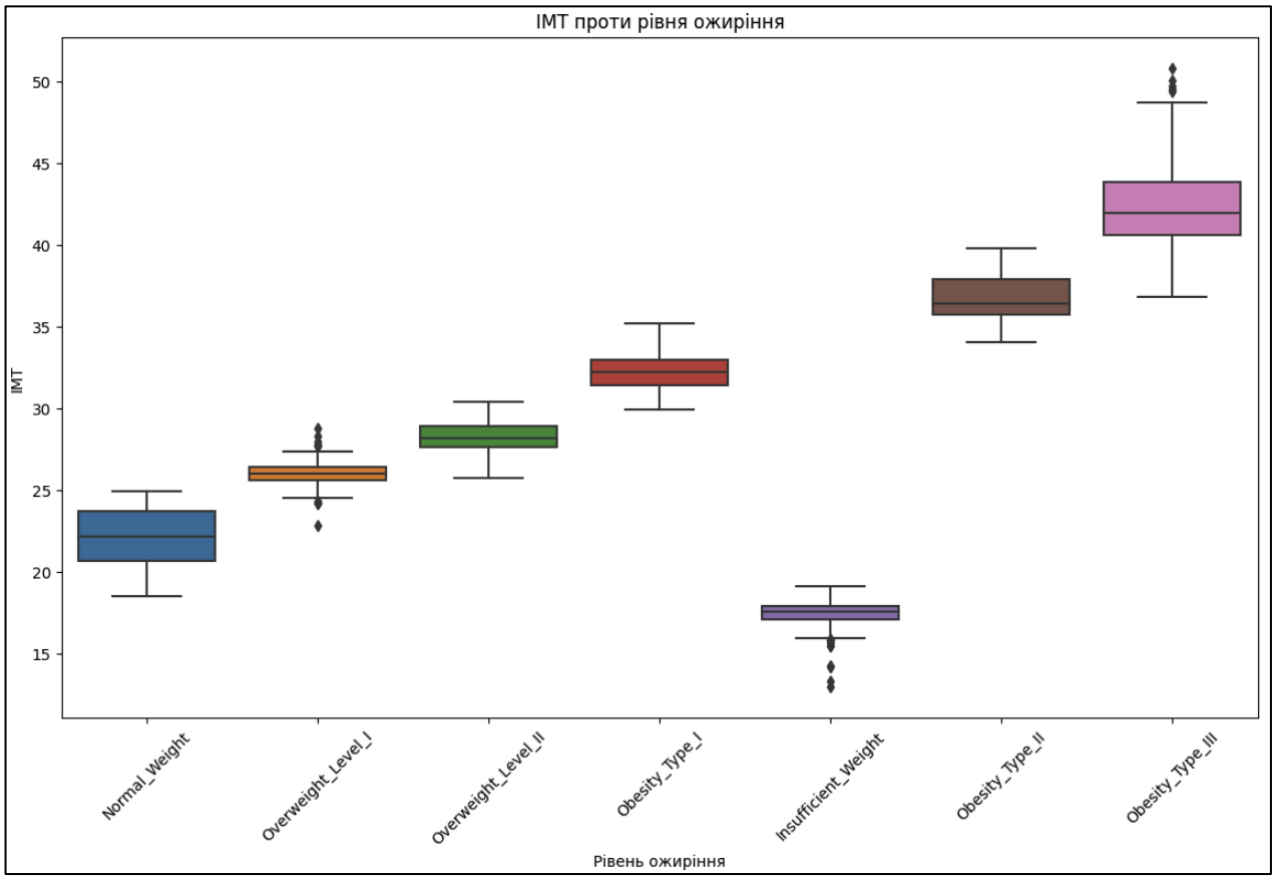

Рисунок 2.25 – Boxplot кореляції IMT та рівня ожиріння

Є чітка тенденція, що вищі значення ІМТ відповідають вищим рівням ожиріння. ІМТ є сильним предиктором рівня ожиріння.

Цей графік показує, як ІМТ (BMI) розподіляється для різних категорій ожиріння (NObeyesdad). Кожен boxplot відображає:

– Мінімальне значення ІМТ в групі
– Нижню квартиль (25-й перцентиль)
– Медіану (50-й перцентиль)
– Верхню квартиль (75-й перцентиль)
– Максимальне значення ІМТ в групі
– Аномальні значення або "викиди" (outliers), якщо вони є

Такий графік допомагає зрозуміти, як змінюється ІМТ в залежності від рівня ожиріння, та виявити потенційні аномалії.

Побудуємо знову матрицю кореляцій Пірсона та Спірмана з новою ознакою (рис. 2.26, 2.27).

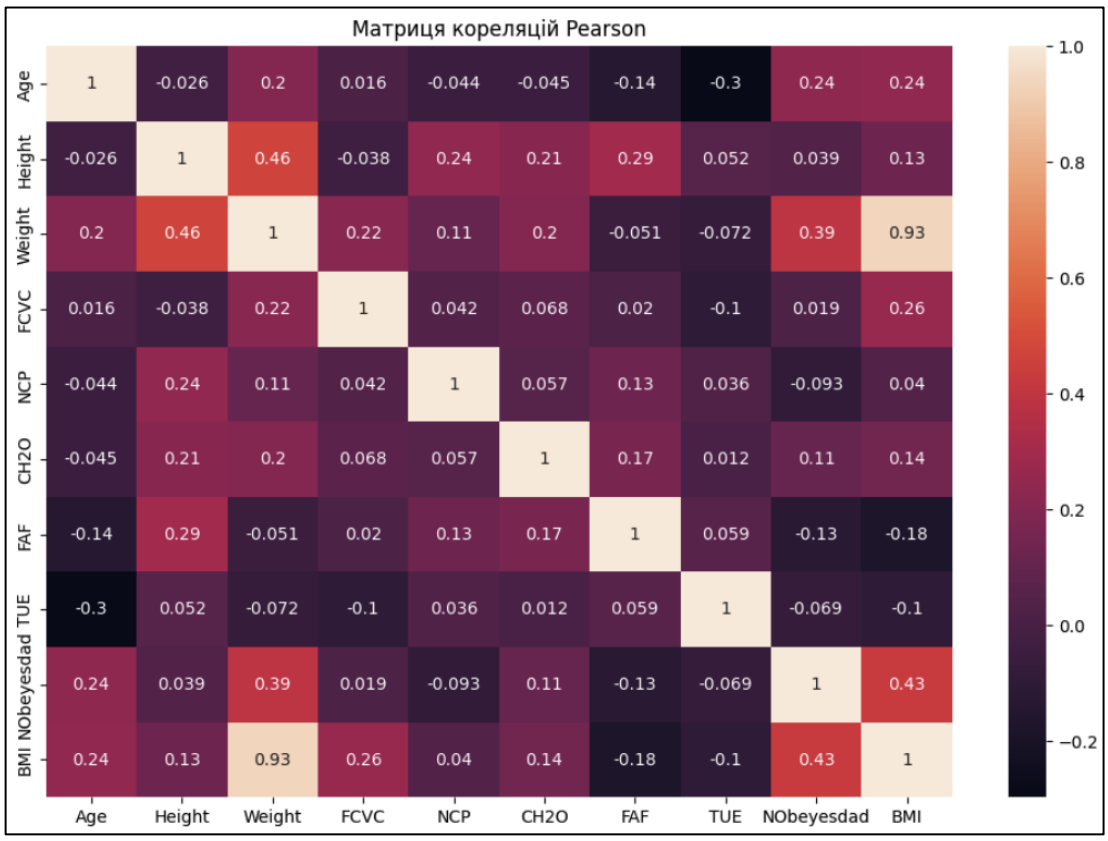

Рисунок 2.26 – Матриця кореляцій Пірсона з ознакою ІМТ(BMI)

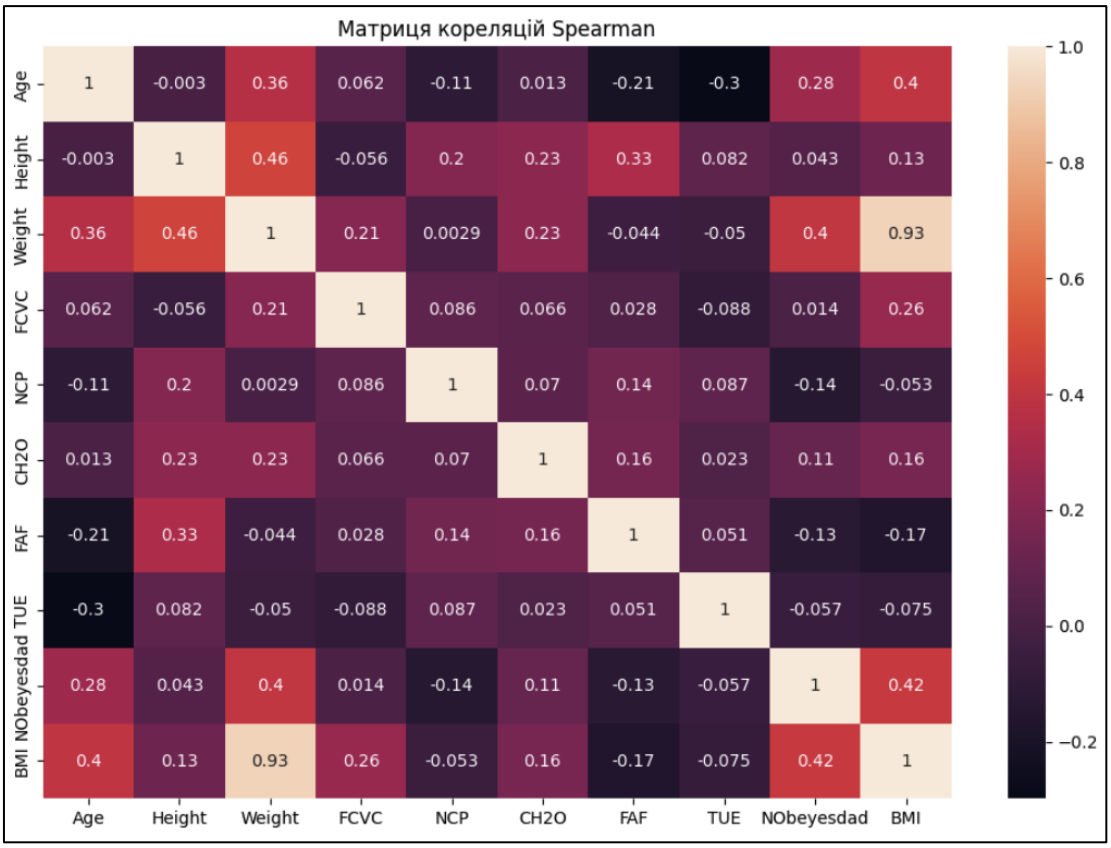

Рисунок 2.26 – Матриця кореляцій Спірмана з ознакою ІМТ(BMI)

Як бачимо, значення на різних матрицях відрізняються не суттєво і наші припущення підтвердились – ІМТ має найбільшу кореляцію з цільовою змінною.

2.2 Тренування та вибір моделей машинного навчання для передбачення ступеня ожиріння

Моделі можуть враховувати ознаки типу bool під час навчання. Однак, щоб забезпечити правильну обробку таких ознак, вони мають бути у відповідному форматі, який може використовуватись моделями. В більшості випадків, ознаки типу bool автоматично конвертуються в 0 та 1, що дозволяє моделям коректно працювати з ними.

Перевіримо, чи всі ознаки типу bool коректно представлені в нашому датасеті, та якщо потрібно, конвертуємо їх у числовий формат (рис 2.27).

```
# Перевірка наявності ознак типу bool
bool_columns = data.select_dtypes(include=['bool']).columns
print(f"Boolean columns: {bool_columns}")

# Конвертація bool ознак у числовий формат (0 та 1)
data[bool_columns] = data[bool_columns].astype(int)
```

Рисунок 2.27 – Перетворення bool даних на числові

За допомогою train_test_split розіб'ємо датасет на навчальну та тестову вибірки з розмірами 80% та 20% від початкової вибірки відповідно (рис 2.28).

```
# Розбиття даних на тренувальний та тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Рисунок 2.28 – Створення навчальної та тестової вибірок

Виконаємо масштабування даних (рис. 2.29).

```
# Масштабування даних
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Рисунок 2.29 – Масштабування даних

Масштабування даних є важливим кроком в процесі підготовки даних для машинного навчання. Основні причини для масштабування даних включають:

– Забезпечення порівнянності: деякі алгоритми машинного навчання (наприклад, SVM, KNN, логістична регресія) є чутливими до масштабу

вхідних даних. Відмінності в масштабах ознак можуть призводити до неправильного обчислення відстаней або градієнтів.

– Прискорення процесу навчання: масштабування даних може прискорити збіжність алгоритмів оптимізації, що використовуються в моделях машинного навчання.

– Покращення точності: масштабовані дані можуть покращити точність і стабільність роботи моделей.

Будемо використовувати такі моделі як: логістична регресія (Logistic Regression), дерева рішень (Decision Trees), випадковий ліс (Random Forest), метод опорних векторів (Support Vector Machine, SVM).

Кожна з яких має свої переваги та недоліки залежно від характеру даних. Логістична регресія проста в реалізації та інтерпретації, добре працює на лінійно роздільних даних. Але може погано працювати на складних, нелінійних даних.

Дерева рішень легко візуалізувати та інтерпретувати, не потребує нормалізації даних. Але може призводити до перенавчення, а також ця модель чутлива до змін у даних.

Випадковий ліс зменшує ризик перенавчення, стійкий до шуму в даних. Але може бути важко інтерпретувати, а також вимагає багато ресурсів для обчислень.

Метод опорних векторів ефективний на високовимірних просторах, добре працює для лінійно розділених класів. Але може бути повільним на великих наборах даних, важко інтерпретувати.

Код навчання моделі наведено на рисунку 2.30.

```
# Моделі
logistic_regression = LogisticRegression(max_iter=1000)
decision_tree = DecisionTreeClassifier(max_depth=5)
random_forest = RandomForestClassifier(n_estimators=100, max_depth=5)
svm = SVC()

# Навчання моделей
logistic_regression.fit(X_train, y_train)
decision_tree.fit(X_train, y_train)
random_forest.fit(X_train, y_train)
svm.fit(X_train, y_train)
```

Рисунок 2.30 – Навчання моделей

Далі отримуємо прогнози для кожної моделі на тестових і тренувальних даних (рис.2.31).

```
# Отримання прогнозів для кожної моделі на тренувальних даних
logistic_regression_pred_train = logistic_regression.predict(X_train)
decision_tree_pred_train = decision_tree.predict(X_train)
random_forest_pred_train = random_forest.predict(X_train)
svm_pred_train = svm.predict(X_train)
```

Рисунок 2.31 – Передбачення моделей на тренувальних даних

Отримані результати роботи моделей покажемо у вигляді метрик та у вигляді конфузійних матриць для кожної моделі для тестових і тренувальних даних.

Результати передбачення логістичної регресії на тренувальних даних і відповідна конфузійна матриця забражені на рисунках 2.32 та 2.33.

```
Model: Logistic Regression (Train Data)
Accuracy: 0.9277251184834123
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.99      0.96       216
           1       0.92      0.84      0.88       225
           2       0.96      0.95      0.95       273
           3       0.97      0.99      0.98       239
           4       1.00      1.00      1.00       261
           5       0.84      0.84      0.84       234
           6       0.87      0.88      0.88       240

    accuracy                           0.93      1688
   macro avg       0.93      0.93      0.93      1688
weighted avg       0.93      0.93      0.93      1688
```

Рисунок 2.32 – Результат передбачення логістичної регресії на тренувальних даних

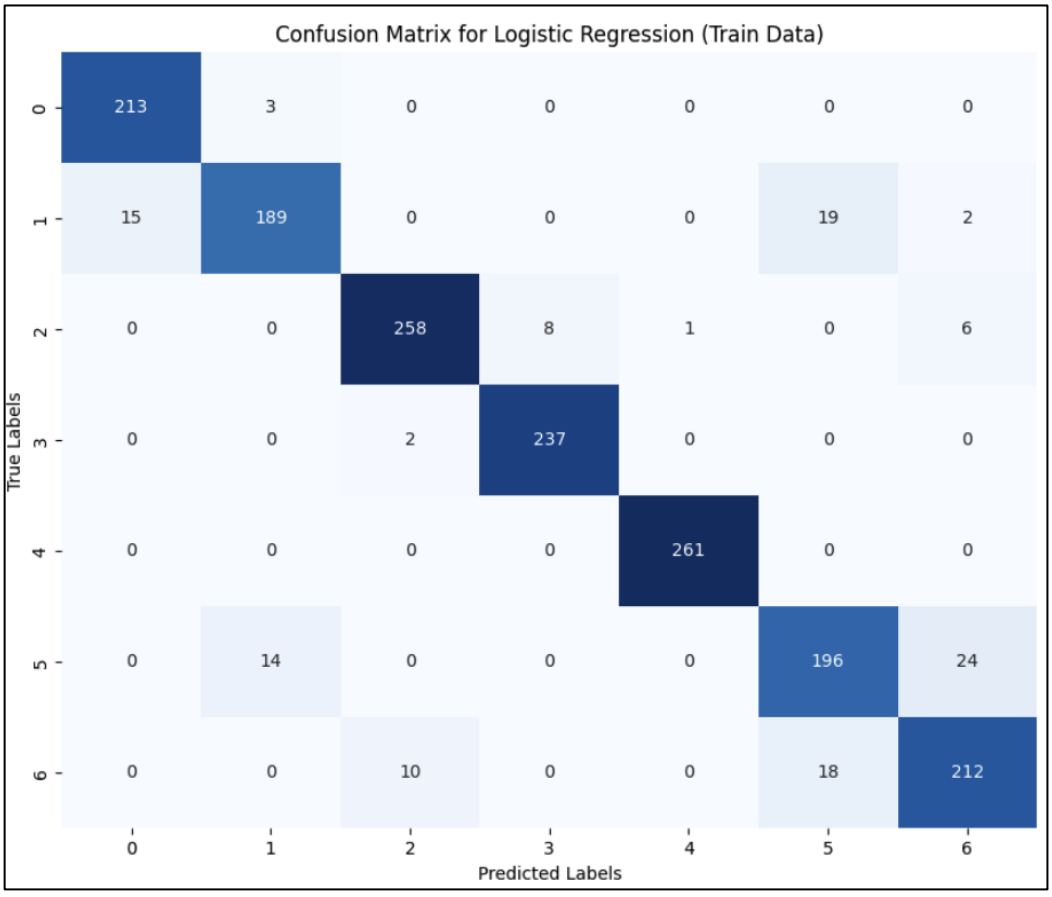

Рисунок 2.33 – Конфузійна матриця логістичної регресії на тренувальних даних

Результати передбачення дерева рішень на тренувальних даних і відповідна конфузійна матриця забражені на рисунках 2.34 та 2.35.

```
Model: Decision Tree (Train Data)
Accuracy: 0.9869668246445498
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       216
           1       0.98      1.00      0.99       225
           2       0.99      0.99      0.99       273
           3       0.99      1.00      0.99       239
           4       1.00      1.00      1.00       261
           5       0.99      0.94      0.97       234
           6       0.96      0.98      0.97       240

    accuracy                           0.99      1688
   macro avg       0.99      0.99      0.99      1688
weighted avg       0.99      0.99      0.99      1688
```

Рисунок 2.34 – Результат передбачення дерева рішень на тренувальних даних

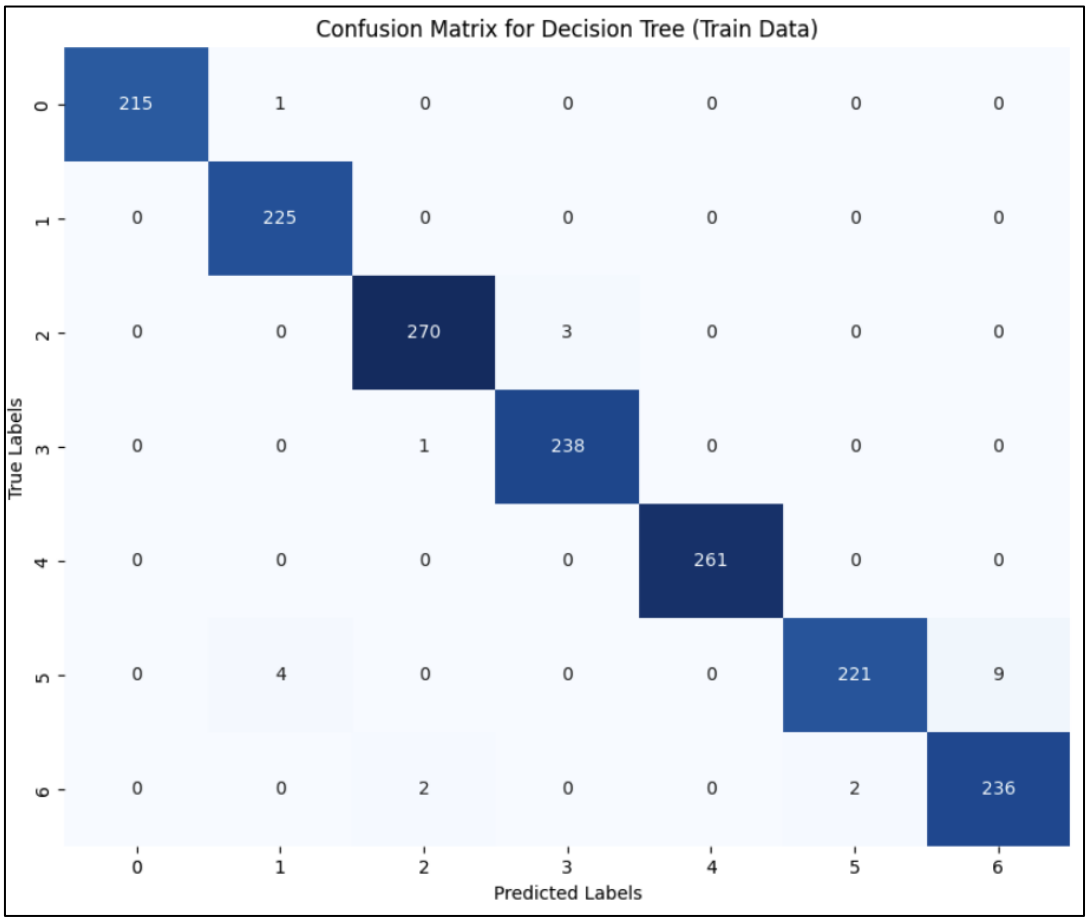

Рисунок 2.35 – Конфузійна матриця дерева рішень на тренувальних даних

Результати передбачення випадкового лісу на тренувальних даних і відповідна конфузійна матриця забражені на рисунках 2.36 та 2.37.

```
Model: Random Forest (Train Data)
Accuracy: 0.9768957345971564
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       216
           1       0.95      0.97      0.96       225
           2       1.00      0.99      0.99       273
           3       0.98      1.00      0.99       239
           4       1.00      1.00      1.00       261
           5       0.96      0.91      0.93       234
           6       0.94      0.97      0.96       240

    accuracy                           0.98      1688
   macro avg       0.98      0.98      0.98      1688
weighted avg       0.98      0.98      0.98      1688
```

Рисунок 2.36 – Результат передбачення випадкового лісу на тренувальних даних

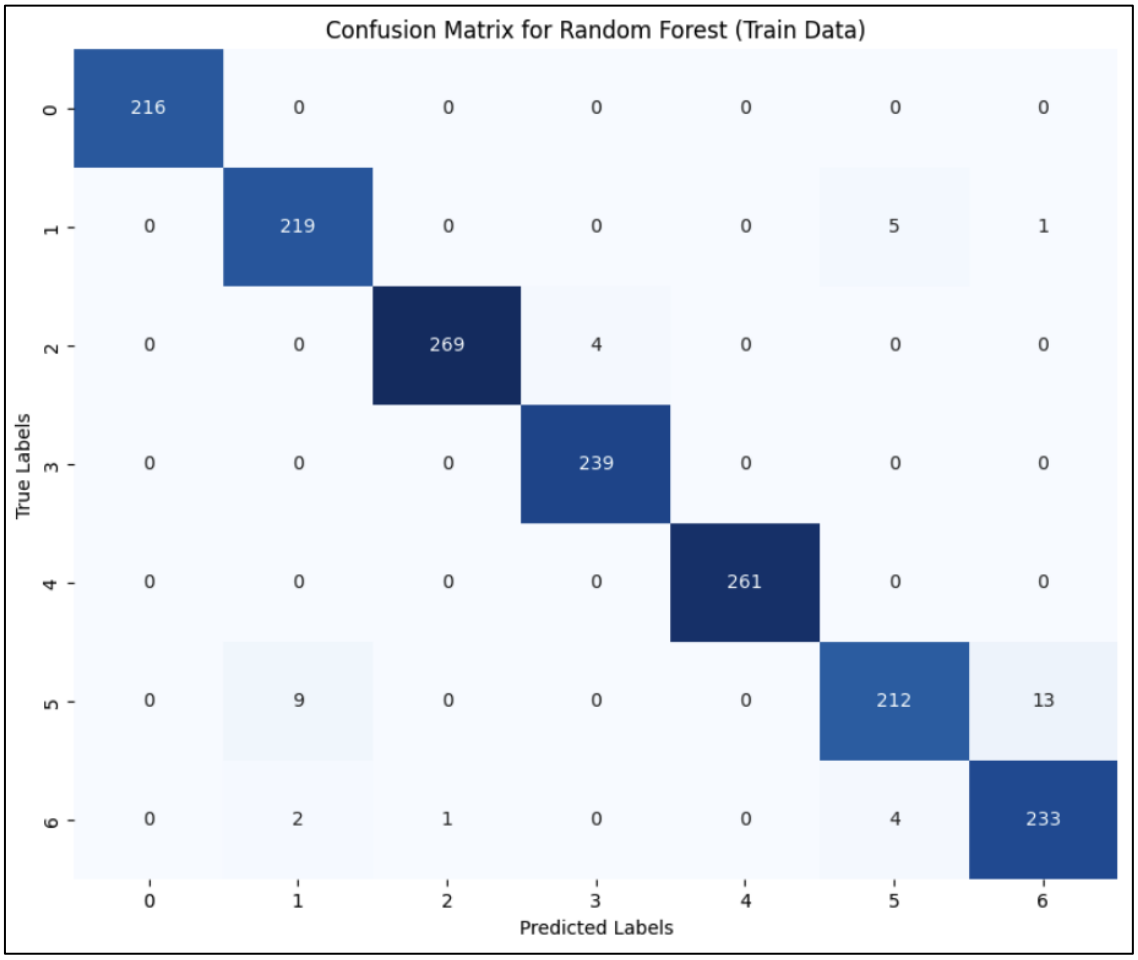

Рисунок 2.37 – Конфузійна матриця випадкового лісу на тренувальних даних

Результати передбачення SVM на тренувальних даних і відповідна конфузійна матриця забражені на рисунках 2.38 та 2.39.

```
Model: SVM (Train Data)
Accuracy: 0.9531990521327014
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.95      0.97       216
           1       0.86      0.92      0.89       225
           2       0.99      0.99      0.99       273
           3       1.00      1.00      1.00       239
           4       1.00      1.00      1.00       261
           5       0.88      0.88      0.88       234
           6       0.95      0.92      0.93       240

    accuracy                           0.95      1688
   macro avg       0.95      0.95      0.95      1688
weighted avg       0.95      0.95      0.95      1688
```

Рисунок 2.38 – Результат передбачення SVM на тренувальних даних

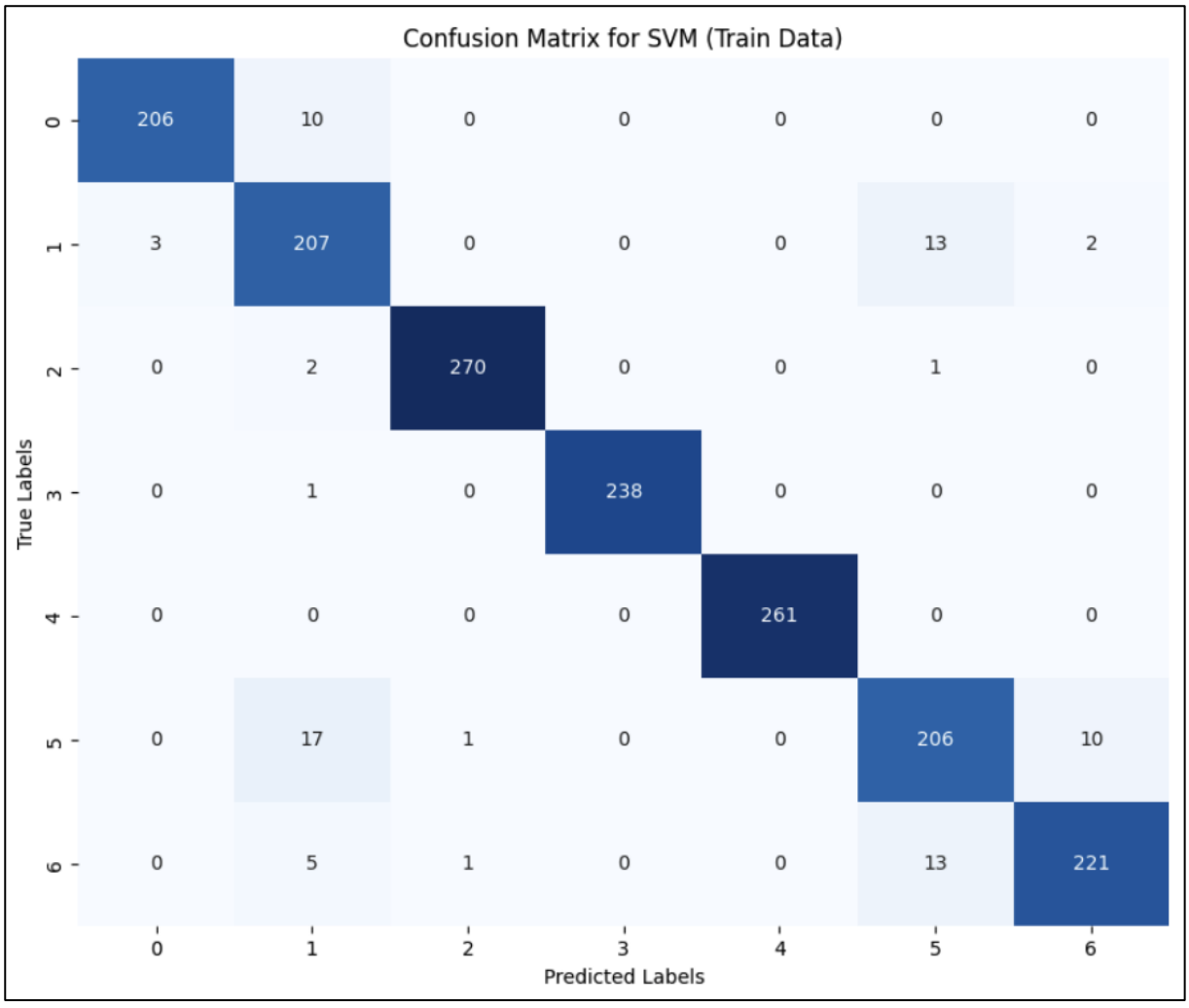

Рисунок 2.39 – Конфузійна матриця SVM на тренувальних даних

Зведемо результати передбачення всіх моделей на тренувальних даних в одну таблицю (рис.2.40).

```
Accuracy Table:
                 Model  Train Accuracy
0  Logistic Regression        0.927725
1        Decision Tree        0.986967
2        Random Forest        0.976896
3                  SVM        0.953199
```

Рисунок 2.40 – Зведена таблиця результатів моделей

Як видно з рисунку 2.40, найкращі результати передбачення за метрикою accuracy_score показали моделі Decision Tree (0,98) та Random Forest (0,97). В наступному розділі перевіримо роботу натренованих моделей на тестових даних.

2.3 Висновки

В другому розділі бакалаврської роботи проведено розвідувальний аналіз даних. Було додано нову ознаку в датасет – індекс маси тіла, яка є загальноприйнятим інструментом для класифікації ступеня ожиріння. Також була проведена конвертація деяких ознак в числовий формат для більш коректної роботи моделей. Виконано масштабування даних для покращення точності і стабільності моделей.

Для рішення задачі передбачення стадії ожиріння натреновано чотири моделі машинного навчання: логістична регресія (Logistic Regression), дерева рішень (Decision Trees), випадковий ліс (Random Forest), метод опорних векторів (Support Vector Machine, SVM). Найкращі результати передбачення на тренувальних даних за метрикою accuracy_score показали моделі Decision Tree (0,98) та Random Forest (0,97).

3 ПЕРЕДБАЧЕННЯ СТУПЕНЯ ОЖИРІННЯ

3.1 Передбачення ступеня ожиріння методами машинного навчання

Застосуємо натреновані моделі машинного навчання до тестової вибірки даних (рис.3.1).

```
# Отримання прогнозів для кожної моделі на тестових даних
logistic_regression_pred_test = logistic_regression.predict(X_test)
decision_tree_pred_test = decision_tree.predict(X_test)
random_forest_pred_test = random_forest.predict(X_test)
svm_pred_test = svm.predict(X_test)
```

Рисунок 3.1 – Передбачення моделей на тестових даних

Отримані результати роботи моделей покажемо у вигляді метрик та у вигляді конфузійних матриць для кожної моделі для тестових даних.

Результати передбачення логістичної регресії на тестових даних і відповідна конфузійна матриця забражені на рисунках 3.2 та 3.3.

```
Model: Logistic Regression (Test Data)
Accuracy: 0.8983451536643026
Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        56
           1       0.94      0.74      0.83        62
           2       0.96      0.90      0.93        78
           3       0.90      0.97      0.93        58
           4       1.00      1.00      1.00        63
           5       0.80      0.79      0.79        56
           6       0.78      0.90      0.83        50

    accuracy                           0.90       423
   macro avg       0.90      0.90      0.89       423
weighted avg       0.90      0.90      0.90       423
```

Рисунок 3.2 – Результат передбачення логістичної регресії на тестових даних

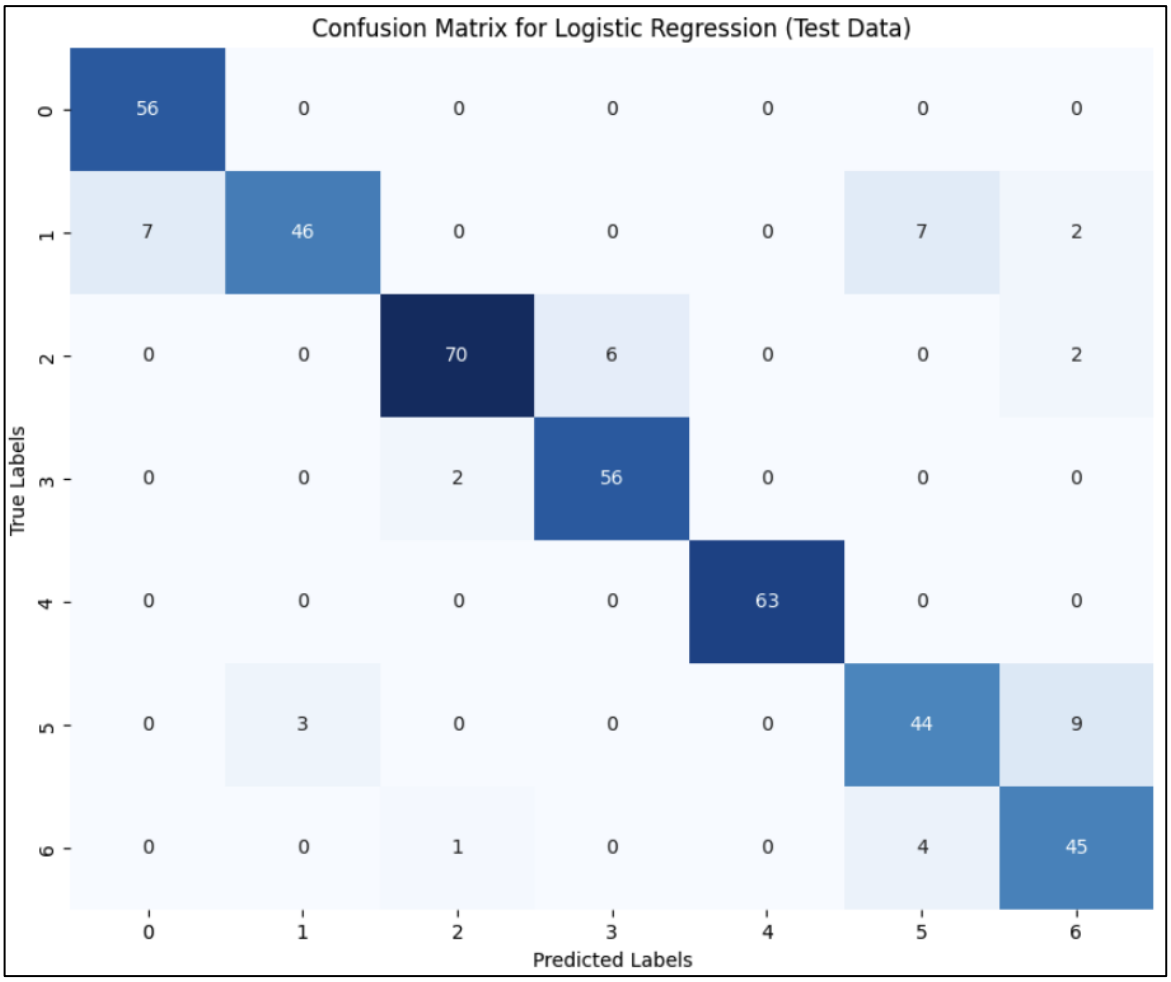

Рисунок 3.3 – Конфузійна матриця логістичної регресії на тестових даних

Результати передбачення дерева рішень на тестових даних і відповідна конфузійна матриця забражені на рисунках 3.4 та 3.5.

```
Model: Decision Tree (Test Data)
Accuracy: 0.9645390070921985
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.95      0.95        56
           1       0.91      0.97      0.94        62
           2       1.00      0.96      0.98        78
           3       0.95      1.00      0.97        58
           4       1.00      1.00      1.00        63
           5       1.00      0.88      0.93        56
           6       0.93      1.00      0.96        50

    accuracy                           0.96       423
   macro avg       0.96      0.96      0.96       423
weighted avg       0.97      0.96      0.96       423
```

Рисунок 3.4 – Результат передбачення дерева рішень на тестових даних

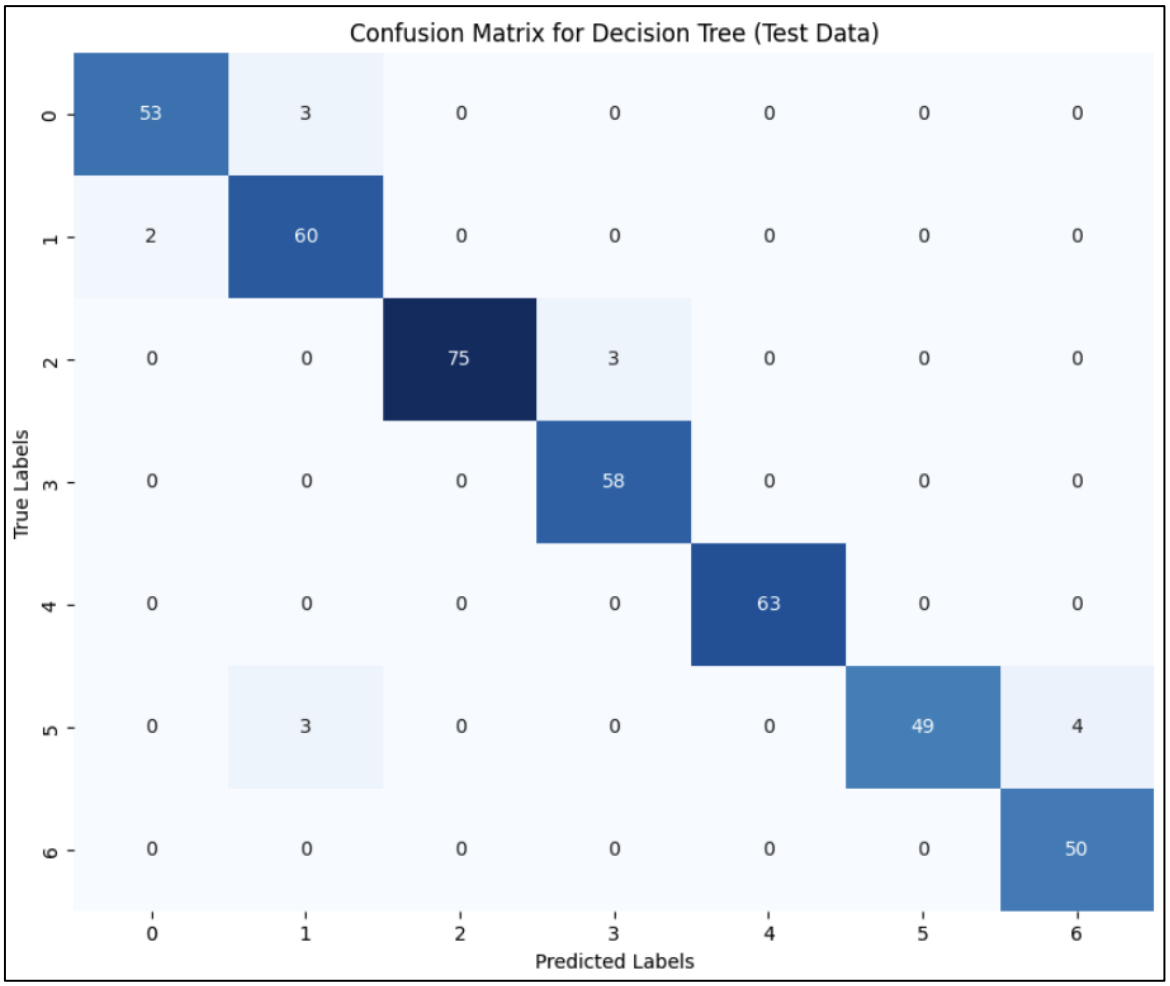

Рисунок 3.5 – Конфузійна матриця дерева рішень на тестових даних

Результати передбачення випадкового лісу на тестових даних і відповідна конфузійна матриця забражені на рисунках 3.6 та 3.7.

```
Model: Random Forest (Test Data)
Accuracy: 0.9598108747044918
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98        56
           1       0.88      0.98      0.93        62
           2       0.96      0.99      0.97        78
           3       0.98      0.98      0.98        58
           4       1.00      1.00      1.00        63
           5       0.94      0.88      0.91        56
           6       0.96      0.90      0.93        50

    accuracy                           0.96       423
   macro avg       0.96      0.96      0.96       423
weighted avg       0.96      0.96      0.96       423
```

Рисунок 3.6 – Результат передбачення випадкового лісу на тестових даних

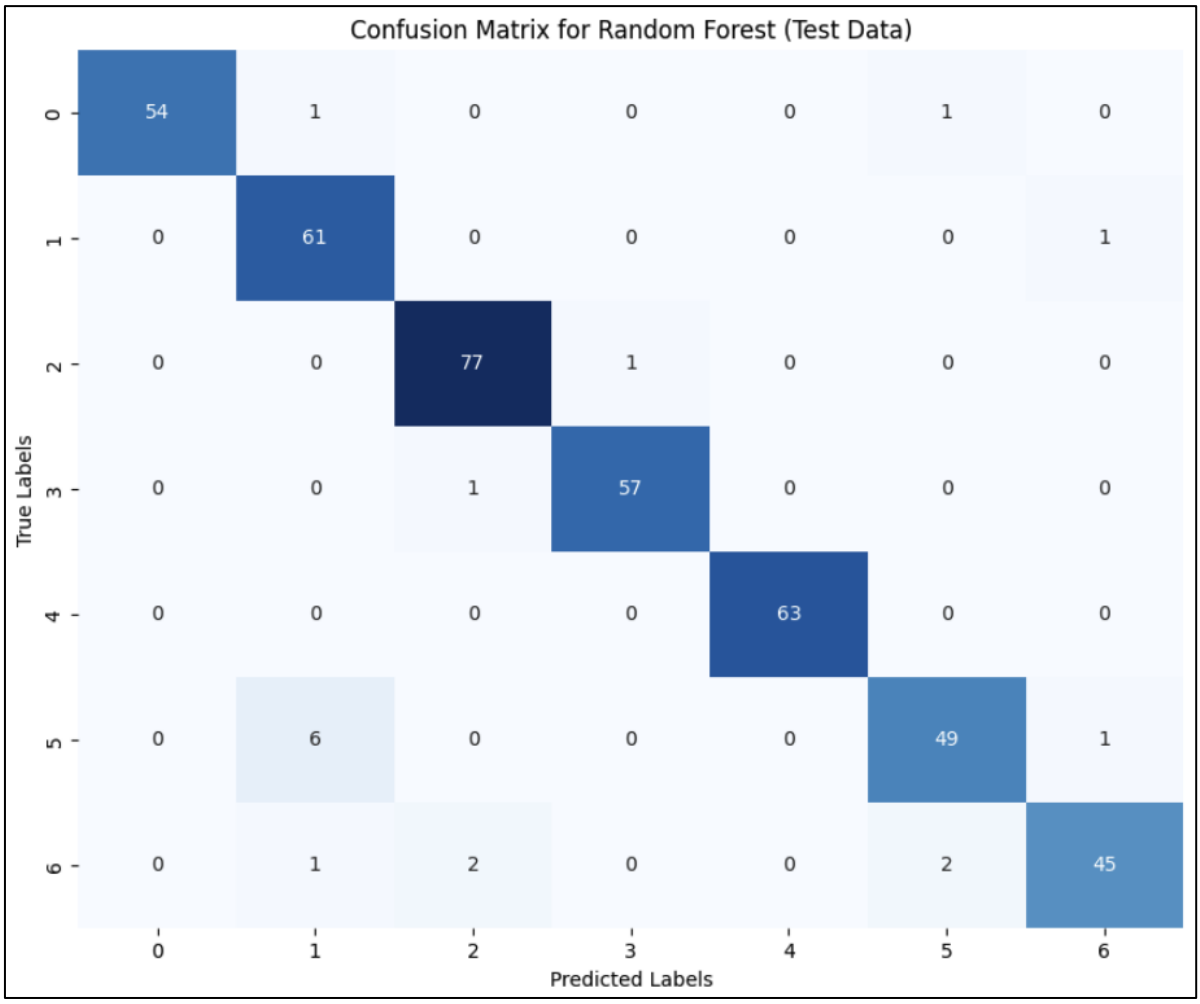

Рисунок 3.7 – Конфузійна матриця Random Forest на тестових даних

Результати передбачення SVM на тестових даних і відповідна конфузійна матриця забражені на рисунках 3.8 та 3.9.

```
Model: SVM (Test Data)
Accuracy: 0.9078014184397163
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.95      0.95        56
           1       0.75      0.89      0.81        62
           2       0.99      0.91      0.95        78
           3       0.97      0.98      0.97        58
           4       1.00      1.00      1.00        63
           5       0.79      0.75      0.77        56
           6       0.90      0.86      0.88        50

    accuracy                           0.91       423
   macro avg       0.91      0.91      0.91       423
weighted avg       0.91      0.91      0.91       423
```

Рисунок 3.8 – Результат передбачення SVM на тестових даних

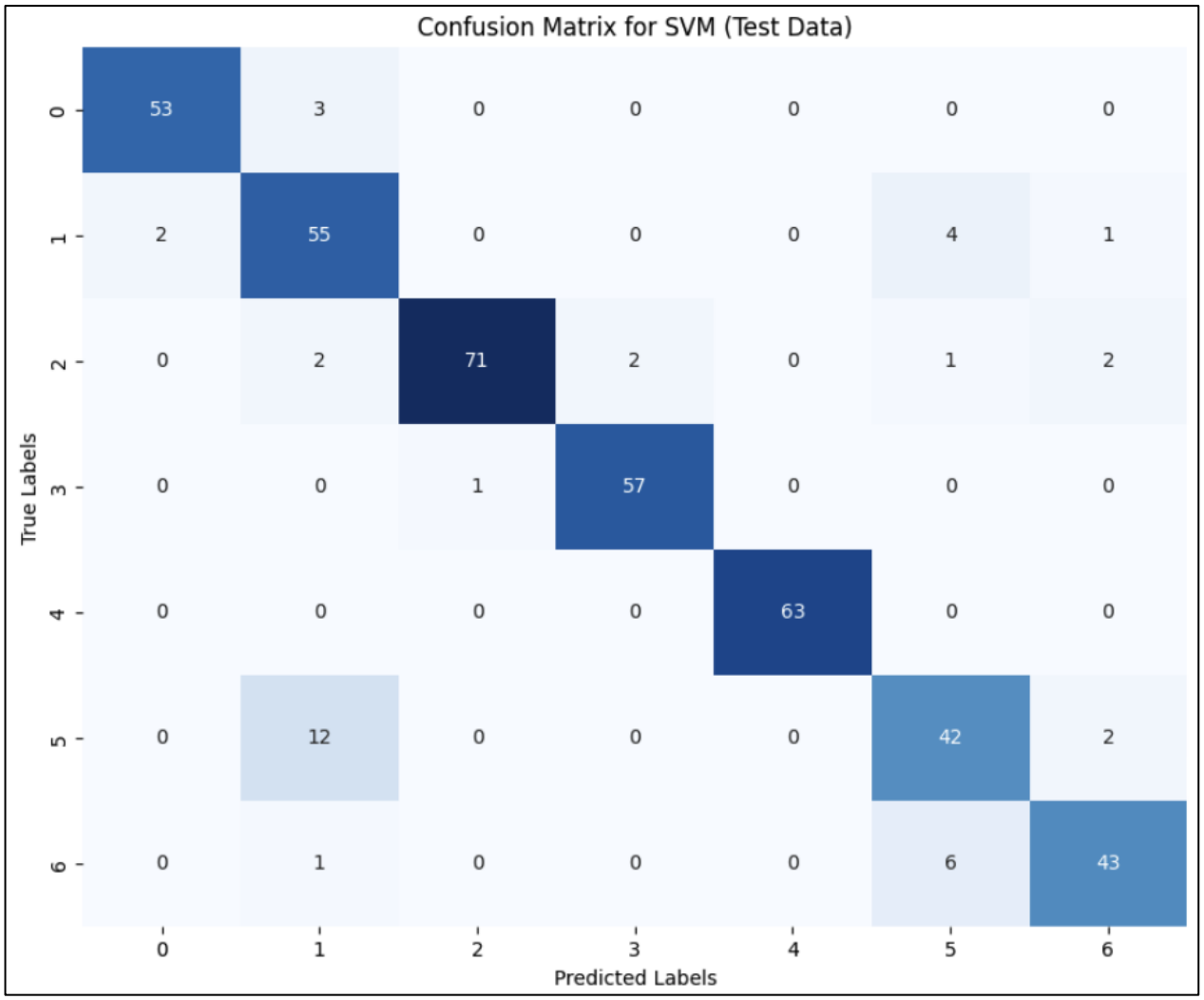

Рисунок 3.9 – Конфузійна матриця SVM на тестових даних

Зведемо результати всіх моделей в одну таблицю, а також візуалізуємо ці результати для кращого сприйняття (рис.3.10, 3.11).

Accuracy Table:

	Model	Train Accuracy	Test Accuracy
0	Logistic Regression	0.927725	0.898345
1	Decision Tree	0.986967	0.964539
2	Random Forest	0.976896	0.959811
3	SVM	0.953199	0.907801

Рисунок 3.10 – Зведена таблиця результатів моделей

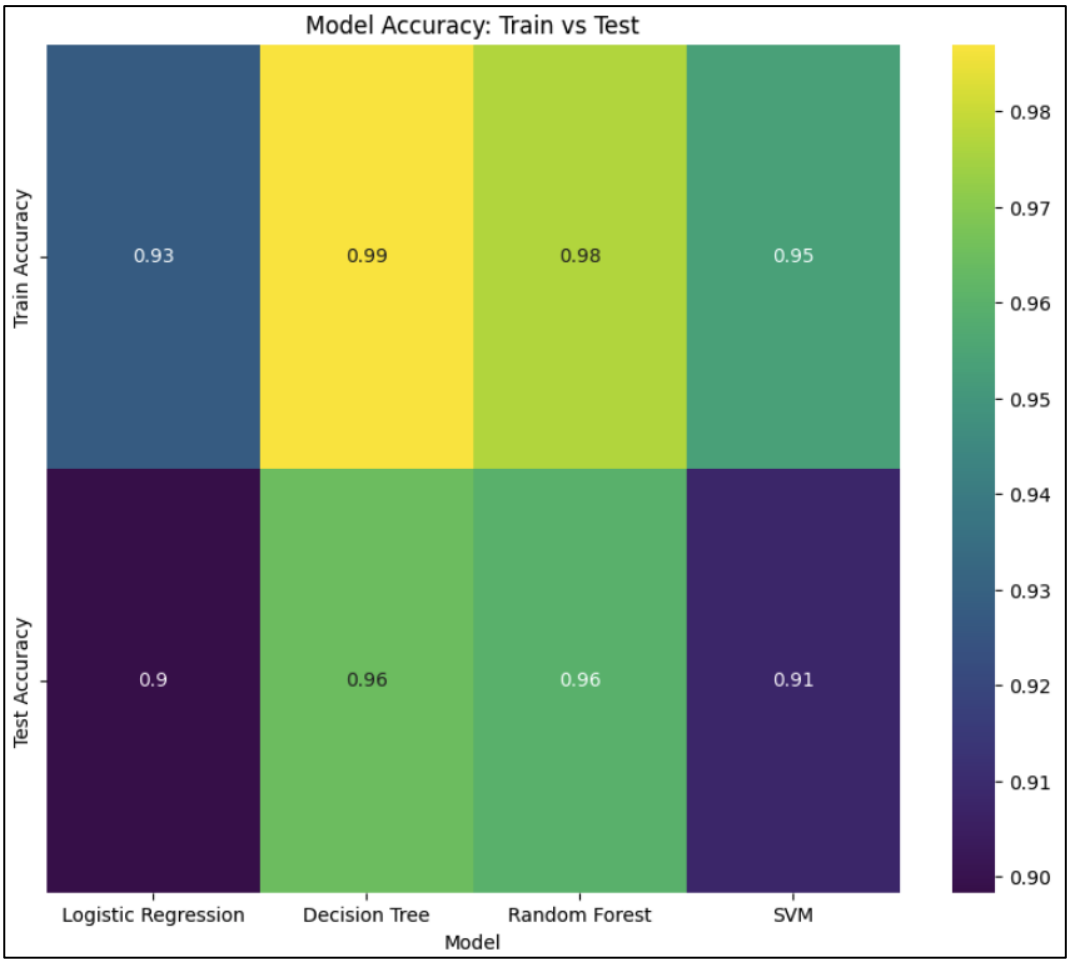

Рисунок 3.11 – Результати всіх моделей

За результатами роботи моделей можемо зробити висновки, що найкращою виявилась модель випадковий ліс з точністю 96% на тестових даних.

Візуалізуємо важливість ознак для цієї моделі (рис. 3.12, 3.13).

```
# Feature importance using Random Forest
importances = random_forest.feature_importances_
features = X.columns

feature_importance_df = pd.DataFrame({'Feature': features, 'Importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

# Plot feature importance
plt.figure(figsize=(12, 8))
sns.barplot(x='Importance', y='Feature', data=feature_importance_df)
plt.title('Feature Importance from Random Forest')
plt.show()
```

Рисунок 3.12 – Код для розрахунку важливості ознак

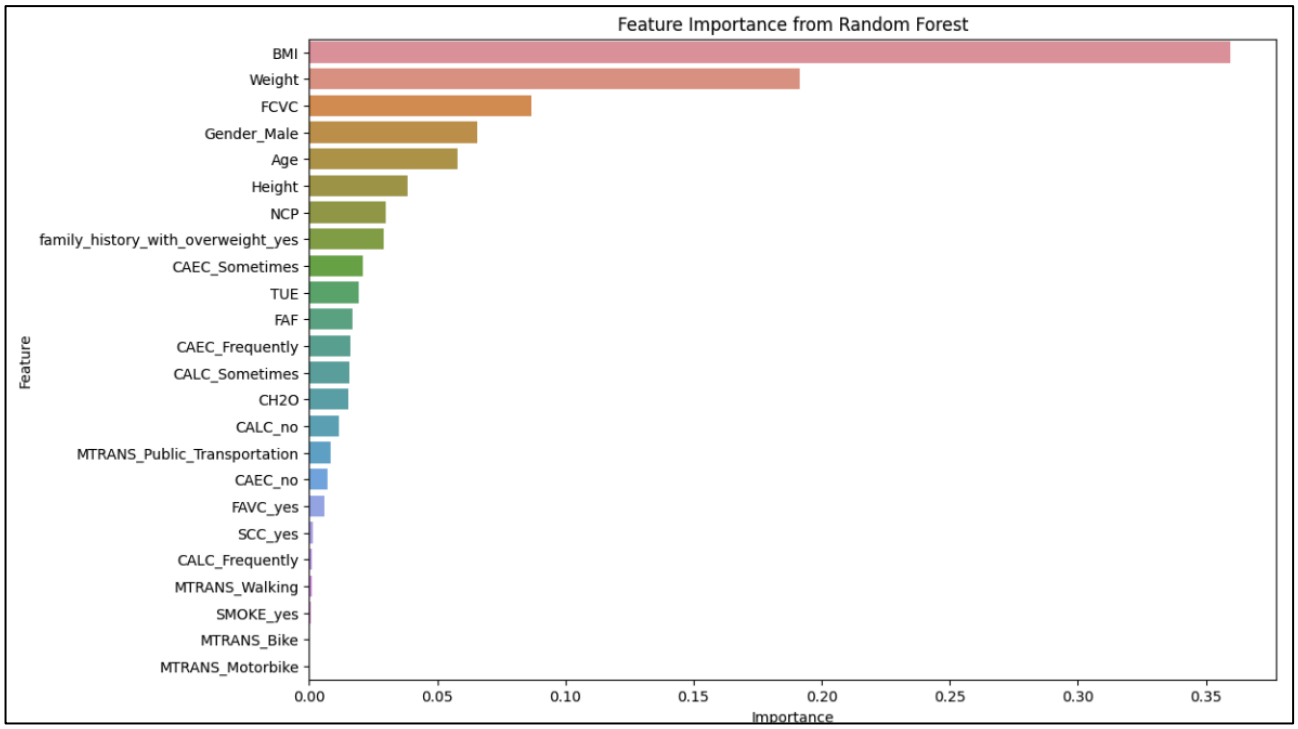

Рисунок 3.13 – Візуалізація важливості ознак для моделі Random Forest

Як видно з рисунку 3.13, такі ознаки як ІМТ, вага, та частота споживання овочів мали найбільший вплив на результат роботи моделі. А такі ознаки як використовуваний транспорт та курець чи ні – практично не впливають на результат передбачення. Тому їх можна видалити з датафрейму для підвищення точності передбачення.

3.2 Висновки

В третьому розділі проведено передбачення ступеня ожиріння на тестових даних, використовуючи натреновані моделі. Для оцінювання результатів передбачення використано метрику accuracy_score.

Оцінивши результати передбачення на тренувальних та тестових даних обрано оптимальну модель Random Forest з точністю передбачення на тестових даних accuracy_score = 0.96.

Також проведено дослідження впливу різних ознак на результати роботи моделі Random Forest. Дослідження показали що для моделі Random

Forest найвпливовішими ознаками є ІМТ, вага та частота споживання овочів. А найменш впливовими – такі ознаки, як використовуваний транспорт та курець чи ні.

ВИСНОВКИ

В роботі проведено аналіз та передбачення ступеня ожиріння методами машинного навчання.

В першому розділі проведено аналіз предметної області. Аналіз предметної області показав, що машинне навчання має великий потенціал для передбачення ступеня ожиріння. Використання сучасних алгоритмів ML дозволяє аналізувати великі обсяги даних та виявляти приховані патерни, що може сприяти розробці ефективних стратегій попередження та лікування ожиріння. Подальші дослідження у цій сфері можуть включати розробку більш точних моделей, інтеграцію додаткових джерел даних та створення інтелектуальних систем підтримки прийняття рішень для медичних фахівців.

Проаналізовано методи машинного навчання для передбачення ступеня ожиріння. Обрано декілька актуальних моделей машинного навчання часто використовуваних в передбаченні медичних діагнозів.

В другому розділі бакалаврської роботи проведено розвідувальний аналіз даних. Було додано нову ознаку в датасет – індекс маси тіла, яка є загальноприйнятим інструментом для класифікації ступеня ожиріння. Також було проведено конвертацію деяких ознак в числовий формат для більш коректної роботи моделей. Виконано масштабування даних для покращення точності і стабільності моделей.

Для рішення задачі передбачення стадії ожиріння натреновано чотири моделі машинного навчання: логістична регресія (Logistic Regression), дерева рішень (Decision Trees), випадковий ліс (Random Forest), метод опорних векторів (Support Vector Machine, SVM).

В третьому розділі проведено передбачення ступеня ожиріння на тестових даних, використовуючи натреновані моделі. Для оцінювання результатів передбачення використано метрику accuracy_score.

Оцінивши результати передбачення на тренувальних та тестових даних обрано оптимальну модель Random Forest з точністю передбачення на тестових даних accuracy_score = 0.96.

Також проведено дослідження впливу різних ознак на результати роботи моделі Random Forest. Дослідження показали що для моделі Random Forest найвпливовішими ознаками є ІМТ, вага та частота споживання овочів. А найменш впливовими – такі ознаки, як використовуваний транспорт та курець чи ні.

За результатами даної роботи зроблено доповідь на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2023-2024 рр.) з публікацією тез.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. World Health Organization. (2024). Obesity and overweight. URL: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight

2. American Heart Association. (2023). Major CVD event risk cut by 20% in adults without diabetes, with overweight or obesity. URL: https://newsroom.heart.org/news/major-cvd-event-risk-cut-by-20-in-adults-without-diabetes-with-overweight-or-obesity

3. Kyrgiafini M-A, Sarafidou T, Giannoulis T. Gene-by-Sex Interactions: Genome-Wide Association Study Reveals Five SNPs Associated with Obesity and Overweight in a Male Population. *Genes*. 2023; 14(4):799. DOI: https://doi.org/10.3390/genes14040799

4. Chatterjee A, Gerdes MW, Martinez SG. Identification of Risk Factors Associated with Obesity and Overweight—A Machine Learning Overview. *Sensors*. 2020; 20(9):2734. https://doi.org/10.3390/s20092734

5. Ferreras, A., Sumalla-Cano, S., Martínez-Licort, R. et al. Systematic Review of Machine Learning applied to the Prediction of Obesity and Overweight. *Journal of Medical Systems* 47, 8 (2023). DOI: https://doi.org/10.1007/s10916-022-01904-1

6. Пянкевич А.Д., Жуков С.О. Розвідувальний аналіз даних для інформаційної технології передбачення ступеня ожиріння методами машинного навчання. *Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2023-2024 рр.)»*. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/21758

7. Sarah E. Hampl, Sandra G. Hassink, Asheley C. Skinner, and others. Clinical Practice Guideline for the Evaluation and Treatment of Children and Adolescents With Obesity. *Pediatrics*. February 2023, 151 (2). DOI: https://doi.org/10.1542/peds.2022-060640

8. Haam JH, Kim BT, Kim EM, et al. Diagnosis of Obesity: 2022 Update of Clinical Practice Guidelines for Obesity by the Korean Society for the Study of Obesity. *J Obes Metab Syndr*. 2023;32(2):121-129. DOI:10.7570/jomes23031

9. Devajit Mohajan and Haradhan Kumar Mohajan. Body Mass Index (BMI) is a Popular Anthropometric Tool to Measure Obesity Among Adults, *JIMR*, vol. 2, no. 4, pp. 25–33, May 2023. URL: https://www.paradigmpress.org/jimr/article/view/578

10. Chatterjee A, Gerdes MW, Martinez SG. Identification of Risk Factors Associated with Obesity and Overweight – A Machine Learning Overview. *Sensors*. 2020; 20(9):2734. https://doi.org/10.3390/s20092734

11. J. -H. Jeong, I. -G. Lee, S. -K. Kim, T. -E. Kam, S. -W. Lee and E. Lee, "DeepHealthNet: Adolescent Obesity Prediction System Based on a Deep Learning Framework," in *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 4, pp. 2282-2293, April 2024, doi: 10.1109/JBHI.2024.3356580.

12. Solomon DD, Khan S, Garg S, Gupta G, Almjally A, Alabduallah BI, Alsagri HS, Ibrahim MM, Abdallah AMA. Hybrid Majority Voting: Prediction and Classification Model for Obesity. *Diagnostics*. 2023; 13(15):2610. https://doi.org/10.3390/diagnostics13152610

13. Mondal PK, Foysal KH, Norman BA, Gittner LS. Predicting Childhood Obesity Based on Single and Multiple Well-Child Visit Data Using Machine Learning Classifiers. *Sensors*. 2023; 23(2):759. https://doi.org/10.3390/s23020759

14. Junhwi Jeon, Sunmi Lee, Chunyoung Oh. Age-specific risk factors for the prediction of obesity using a machine learning approach. Front. Public Health, Volume 10 - 2022 | https://doi.org/10.3389/fpubh.2022.998782

15. Yagin FH, Gülü M, Gormez Y, Castañeda-Babarro A, Colak C, Greco G, Fischetti F, Cataldi S. Estimation of Obesity Levels with a Trained Neural Network Approach optimized by the Bayesian Technique. *Applied Sciences*. 2023; 13(6):3875. https://doi.org/10.3390/app13063875

16. Мокін В. Б., Дратований М. В. Наука про дані: машинне навчання та інтелектуальний аналіз даних : електронний навчальний посібник комбінованого (локального та мережевого) використання [Електронний ресурс]. Вінниця : ВНТУ, 2024, 258 с. URL: https://docs.vntu.edu.ua/card.php?id=8163

17. Kaggle Dataset «Cirrhosis Prediction Dataset» URL: https://www.kaggle.com/datasets/fedesoriano/cirrhosis-prediction-dataset

18. Кореляційна матриця Пірсона та Спірмена [Електронний ресурс]. – URL: https://www.guru99.com/uk/r-pearson-spearman-correlation.html .

19. AutoML documentation. URL: https://www.automl.org/automl/.

20. Штовба С. Д., Козачко О. М. «Machine Learning: стартовий курс Електронний навчальний посібник». Вінниця : ВНТУ, 2020. 81 с.

Додаток А

(обов’язковий)

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 бакалаврську дипломну роботу

ПЕРЕДБАЧЕННЯ СТУПЕНЯ ОЖИРІННЯ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

08-34.БДР.007.00.000 ТЗ

Керівник: к.т.н., доц.

_____ Сергій ЖУКОВ

«__» _____ 2024 р.

Розробив студент гр. СА-20б

_____ Андрій ПЯНКЕВИЧ

«__» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт

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

2. Джерела розробки

1) Kaggle Dataset «Cirrhosis Prediction Dataset» URL: https://www.kaggle.com/datasets/fedesoriano/cirrhosis-prediction-dataset;

2) Мокін В. Б., Дратований М. В. Наука про дані: машинне навчання та інтелектуальний аналіз даних : електронний навчальний посібник комбінованого (локального та мережевого) використання [Електронний ресурс]. Вінниця : ВНТУ, 2024, 258 с. URL: https://docs.vntu.edu.ua/card.php?id=8163

3. Мета і призначення роботи.

Метою дослідження є підвищення точності передбачення ступеня ожиріння за рахунок використання методів машинного навчання.

4. Вихідні дані для проведення робіт:

Kaggle Dataset «Obesity or CVD risk (Classify/Regressor/Cluster)», який містить дані пацієнтів для передбачення стадії ожиріння.

5. Методи дослідження:

В даній роботі використовується моделі класифікатори машинного навчання: SVM, Decision Tree, Logistic Regression, Random Forest.

6. Етапи роботи і терміни їх виконання

a) Аналіз предметної області ..... ____ – ____

b) Розвідувальний аналіз даних ..... ____ – ____

c) Тренування моделей машинного навчання ..... ____ – ____

d) Вибір оптимальної моделі машинного навчання та передбачення ступеня ожиріння ..... ____ – ____

e) Оформлення матеріалів до захисту БДР ..... ____ – ____

7. Очікувані результати та порядок реалізації

Отримання результатів по передбаченню стадії ожиріння з задовільною точністю.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)».

9. Порядок приймання роботи

Публічний захист «___» _____ 2024 р.

Початок розробки «___» _____ 2024 р.

Граничні терміни виконання БДР «___» _____ 2024 р.

Розробив студент групи СА-20б _____ Андрій ПЯНКЕВИЧ

Додаток Б

(обов'язковий)

Протокол перевірки бакалаврської дипломної роботи на наявність текстових запозичень

Назва роботи: «Передбачення ступеня ожиріння методами машинного навчання»
Тип роботи: бакалаврська дипломна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 88% 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 самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.

○ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Сергій ЖУКОВ
(підпис)

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

Автор роботи _____ Андрій ПЯНКЕВИЧ
(підпис)

Керівник роботи _____ Сергій ЖУКОВ
(підпис)

Додаток В

(довідниковий)

Лістинг програми

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC

# Load the dataset
#data = pd.read_csv('/kaggle/input/playground-series-s4e2/train.csv')
data = pd.read_csv('/kaggle/input/obesity-or-cvd-risk-classifyregressorcluster/ObesityDataSet.csv')

# Display the first few rows of the dataset
data.head()
# Check for missing values
print(data.isnull().sum())
# Basic statistics of the dataset
print(data.info())
print(data.describe(include='number'))
# Check for missing values
print(data.isnull().sum())

# Basic statistics of the dataset
print(data.info())
```

```
print(data.describe(include='all'))

# Distribution of the target variable
plt.figure(figsize=(10, 6))
sns.countplot(x='NObeyesdad', data=data)
plt.title('Distribution of the target variable')
plt.xticks(rotation=45)
plt.show()
# Перевірка унікальних значень цільової змінної
print(data['NObeyesdad'].unique())
# Побудова розподілу цільової змінної
plt.figure(figsize=(10, 6))
sns.countplot(x='NObeyesdad', data=data)
plt.title('Розподіл рівнів ожиріння')
plt.xlabel('Рівні ожиріння')
plt.ylabel('Кількість')
plt.xticks(rotation=45)
plt.show()
# Побудова розподілу віку
plt.figure(figsize=(10, 6))
sns.histplot(data['Age'], kde=True)
plt.title('Розподіл віку')
plt.xlabel('Вік')
plt.ylabel('Частота')
plt.show()
# Побудова розподілу за статтю
plt.figure(figsize=(10, 6))
#sns.countplot(data['Gender'])
sns.countplot(x='Gender', data=data)
plt.title('Розподіл за статтю')
plt.xlabel('Стать')
plt.ylabel('Кількість')
plt.show()
```

```
# Вибір тільки числових стовпців
numeric_columns = data.select_dtypes(include='number').columns

# Побудова гістограм для всіх числових ознак
data[numeric_columns].hist(bins=30, figsize=(15, 10))
plt.tight_layout()
plt.show()
# Побудова розподілу зросту
plt.figure(figsize=(10, 6))
sns.histplot(data['Height'], kde=True)
plt.title('Розподіл зросту')
plt.xlabel('Зріст')
plt.ylabel('Частота')
plt.show()
# Побудова розподілу ваги
plt.figure(figsize=(10, 6))
sns.histplot(data['Weight'], kde=True)
plt.title('Розподіл ваги')
plt.xlabel('Вага')
plt.ylabel('Частота')
plt.show()
# Побудова розподілу «частота споживання овочів» (FCVC)
plt.figure(figsize=(10, 6))
sns.histplot(data['FCVC'], kde=True)
plt.title('Розподіл частоти споживання овочів')
plt.xlabel('Частота споживання овочів')
plt.ylabel('Частота')
plt.show()
# Побудова розподілу «кількість основних прийомів їжі» (NCP)
plt.figure(figsize=(10, 6))
sns.histplot(data['NCP'], kde=True)
plt.title('Розподіл кількісті основних прийомів їжі')
plt.xlabel('Кількість основних прийомів їжі')
```

```
plt.ylabel('Частота')
plt.show()
# Побудова розподілу «щоденне споживання води» (CH20)
plt.figure(figsize=(10, 6))
sns.histplot(data['CH2O'], kde=True)
plt.title('Розподіл щоденного споживання води')
plt.xlabel('Щоденне споживання води')
plt.ylabel('Частота')
plt.show()
# Побудова розподілу "час використання технологічних пристроїв" (TUE)
plt.figure(figsize=(10, 6))
sns.histplot(data['TUE'], kde=True)
plt.title('Розподіл часу використання технологічних пристроїв')
plt.xlabel('Час використання технологічних пристроїв')
plt.ylabel('Частота')
plt.show()
# Побудова розподілу «частота фізичної активності» (FAF)
plt.figure(figsize=(10, 6))
sns.histplot(data['FAF'], kde=True)
plt.title('Розподіл частоти фізичної активності')
plt.xlabel('Частота фізичної активності')
plt.ylabel('Частота')
plt.show()
# Вибір тільки числових стовпців
numeric_columns = data.select_dtypes(include='number').columns

# Замінюємо значення Inf на NaN
data[numeric_columns] = data[numeric_columns].replace([np.inf, -np.inf], np.nan)

# Виключаємо ознаку 'id'
#numeric_columns = numeric_columns.drop('id')

# Побудова графіків розподілу для всіх числових ознак
```

```
plt.figure(figsize=(15, 10))
for i, col in enumerate(numeric_columns, 1):
    plt.subplot(len(numeric_columns) // 3 + 1, 3, i)
    sns.histplot(data[col].dropna(), kde=True)  # Виключаємо NaN значення
    plt.title(f'Розподіл {col}')
    plt.xlabel(col)
    plt.ylabel('Частота')

plt.tight_layout()
plt.show()
# Вибір тільки булевих стовпців
bool_columns = data.select_dtypes(include='bool').columns
# Вибір тільки нечислових (категоріальних) стовпців
non_numeric_columns = data.select_dtypes(include=['object', 'category']).columns

# Перевірка чи є нечислові стовпці
if len(non_numeric_columns) == 0:
    print("У датасеті немає нечислових стовпців.")
else:
    # Побудова графіків розподілу для всіх нечислових ознак
    # plt.figure(figsize=(15, len(non_numeric_columns) * 5))
    plt.figure(figsize=(15, 15))
    for i, col in enumerate(non_numeric_columns, 1):
        plt.subplot(len(non_numeric_columns) // 3 + 1, 3, i)
        sns.countplot(x=col, data=data)
        plt.title(f'Розподіл {col}')
        plt.xlabel(col)
        plt.ylabel('Частота')
        plt.xticks(rotation=45)  # Поворот підписів на осях x для кращої видимості

    plt.tight_layout()
    plt.show()
from sklearn.preprocessing import LabelEncoder
```

```
# Кодування цільової змінної
label_encoder = LabelEncoder()
data['NObeyesdad'] = label_encoder.fit_transform(data['NObeyesdad'])

# Select only numeric columns for correlation heatmap
numeric_data = data.select_dtypes(include=[np.number])

# Correlation heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm')
#plt.title('Correlation Heatmap')
plt.title('Матриця кореляцій')
plt.show()
data['BMI'] = data['Weight'] / (data['Height'] ** 2)

# Boxplot IMT проти рівня ожиріння
plt.figure(figsize=(14, 8))
sns.boxplot(x='NObeyesdad', y='BMI', data=data)
plt.title('IMT проти рівня ожиріння')
plt.xlabel('Рівень ожиріння')
plt.ylabel('IMT')
plt.xticks(rotation=45)
plt.show()
from sklearn.preprocessing import LabelEncoder

# Кодування цільової змінної
label_encoder = LabelEncoder()
data['NObeyesdad'] = label_encoder.fit_transform(data['NObeyesdad'])

# Select only numeric columns for correlation heatmap
numeric_data = data.select_dtypes(include=[np.number])
```

```
# Correlation heatmap
plt.figure(figsize=(12, 8))
#sns.heatmap(numeric_data.corr(method='spearman'), annot=True, cmap='coolwarm')
sns.heatmap(numeric_data.corr(method='spearman'), annot=True)
#plt.title('Correlation Heatmap')
plt.title('Матриця кореляцій')
plt.show()
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Перевірка наявності ознак типу bool
bool_columns = data.select_dtypes(include=['bool']).columns
print(f"Boolean columns: {bool_columns}")

# Конвертація bool ознак у числовий формат (0 та 1)
data[bool_columns] = data[bool_columns].astype(int)

# Виділення особливостей та цільової змінної
X = data.drop('NObeyesdad', axis=1)
y = data['NObeyesdad']

# Розбиття даних на тренувальний та тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Масштабування даних
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Моделі
logistic_regression = LogisticRegression(max_iter=1000)
decision_tree = DecisionTreeClassifier(max_depth=5)
random_forest = RandomForestClassifier(n_estimators=100, max_depth=5)
svm = SVC()

# Навчання моделей
logistic_regression.fit(X_train, y_train)
decision_tree.fit(X_train, y_train)
random_forest.fit(X_train, y_train)
svm.fit(X_train, y_train)

# Отримання прогнозів для кожної моделі на тестових даних
logistic_regression_pred_test = logistic_regression.predict(X_test)
decision_tree_pred_test = decision_tree.predict(X_test)
random_forest_pred_test = random_forest.predict(X_test)
svm_pred_test = svm.predict(X_test)

# Отримання прогнозів для кожної моделі на тренувальних даних
logistic_regression_pred_train = logistic_regression.predict(X_train)
decision_tree_pred_train = decision_tree.predict(X_train)
random_forest_pred_train = random_forest.predict(X_train)
svm_pred_train = svm.predict(X_train)

# Функція для побудови кореляційної матриці
def plot_correlation_matrix(y_true, y_pred, model_name, dataset_type):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(10, 8))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
```

```
    plt.title(f'Confusion Matrix for {model_name} ({dataset_type})')
    plt.xlabel('Predicted Labels')
    plt.ylabel('True Labels')
    plt.show()

# Оцінка моделей на тестових і тренувальних даних
models = ['Logistic Regression', 'Decision Tree', 'Random Forest', 'SVM']
predictions_test = [logistic_regression_pred_test, decision_tree_pred_test,
random_forest_pred_test, svm_pred_test]
predictions_train = [logistic_regression_pred_train, decision_tree_pred_train,
random_forest_pred_train, svm_pred_train]

# Список для зберігання даних точності
accuracy_data = []

for i, (pred_test, pred_train) in enumerate(zip(predictions_test, predictions_train)):
    train_accuracy = accuracy_score(y_train, pred_train)
    test_accuracy = accuracy_score(y_test, pred_test)

    accuracy_data.append({
        'Model': models[i],
        'Train Accuracy': train_accuracy,
        'Test Accuracy': test_accuracy
    })

    print(f"Model: {models[i]} (Test Data)")
    print("Accuracy:", test_accuracy)
    print("Classification Report:\n", classification_report(y_test, pred_test))
    plot_correlation_matrix(y_test, pred_test, models[i], "Test Data")

    print(f"Model: {models[i]} (Train Data)")
    print("Accuracy:", train_accuracy)
    print("Classification Report:\n", classification_report(y_train, pred_train))
```

```
        plot_correlation_matrix(y_train, pred_train, models[i], "Train Data")
        print("\n")
    # Створення таблиці з точністю моделей
    accuracy_df = pd.DataFrame(accuracy_data)
    # Відображення таблиці з точністю моделей
    print("Accuracy Table:")
    print(accuracy_df)

    # Побудова теплової карти точності моделей
    plt.figure(figsize=(10, 8))
    sns.heatmap(accuracy_df.set_index('Model').T, annot=True, cmap='viridis', cbar=True)
    plt.title('Model Accuracy: Train vs Test')
    plt.show()
    # Feature importance using Random Forest
    importances = random_forest.feature_importances_
    features = X.columns
    feature_importance_df = pd.DataFrame({'Feature': features, 'Importance': importances})
    feature_importance_df = feature_importance_df.sort_values(by='Importance',
ascending=False)
    # Plot feature importance
    plt.figure(figsize=(12, 8))
    sns.barplot(x='Importance', y='Feature', data=feature_importance_df)
    plt.title('Feature Importance from Random Forest')
    plt.show()
    # Analyze results
    print("Model performance analysis:")
    for i, pred in enumerate(predictions):
        accuracy = accuracy_score(y_test, pred)
        print(f"{models[i]} Accuracy: {accuracy}")
    # Display feature importance
    print("Top 10 important features according to Random Forest:")
    print(feature_importance_df.head(10))
```

Додаток Г

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ПЕРЕДБАЧЕННЯ СТУПЕНЯ ОЖИРІННЯ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

```
data
```

	id	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SM
0	0	Male	24.443011	1.699998	81.669950	yes	yes	2.000000	2.983297	Sometimes	
1	1	Female	18.000000	1.560000	57.000000	yes	yes	2.000000	3.000000	Frequently	
2	2	Female	18.000000	1.711460	50.165754	yes	yes	1.880534	1.411685	Sometimes	
3	3	Female	20.952737	1.710730	131.274851	yes	yes	3.000000	3.000000	Sometimes	
4	4	Male	31.641081	1.914186	93.798055	yes	yes	2.679664	1.971472	Sometimes	
...	...	...	...	...	...	...	...	...	...	...	
20753	20753	Male	25.137087	1.766626	114.187096	yes	yes	2.919584	3.000000	Sometimes	
20754	20754	Male	18.000000	1.710000	50.000000	no	yes	3.000000	4.000000	Frequently	
20755	20755	Male	20.101026	1.819557	105.580491	yes	yes	2.407817	3.000000	Sometimes	
20756	20756	Male	33.852953	1.700000	83.520113	yes	yes	2.671238	1.971472	Sometimes	
20757	20757	Male	26.680376	1.816547	118.134898	yes	yes	3.000000	3.000000	Sometimes	

20758 rows × 18 columns

```
data
```

'C	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
es	2.000000	2.983297	Sometimes	no	2.763573	no	0.000000	0.976473	Sometimes	Public_Transportation	Overweight_Level_II
es	2.000000	3.000000	Frequently	no	2.000000	no	1.000000	1.000000	no	Automobile	Normal_Weight
es	1.880534	1.411685	Sometimes	no	1.910378	no	0.866045	1.673584	no	Public_Transportation	Insufficient_Weight
es	3.000000	3.000000	Sometimes	no	1.674061	no	1.467863	0.780199	Sometimes	Public_Transportation	Obesity_Type_III
es	2.679664	1.971472	Sometimes	no	1.979848	no	1.967973	0.931721	Sometimes	Public_Transportation	Overweight_Level_II
...	...	...	...	...	...	...	...	...	...	...	...
es	2.919584	3.000000	Sometimes	no	2.151809	no	1.330519	0.196680	Sometimes	Public_Transportation	Obesity_Type_II
es	3.000000	4.000000	Frequently	no	1.000000	no	2.000000	1.000000	Sometimes	Public_Transportation	Insufficient_Weight
es	2.407817	3.000000	Sometimes	no	2.000000	no	1.158040	1.198439	no	Public_Transportation	Obesity_Type_II
es	2.671238	1.971472	Sometimes	no	2.144838	no	0.000000	0.973834	no	Automobile	Overweight_Level_II
es	3.000000	3.000000	Sometimes	no	2.003563	no	0.684487	0.713823	Sometimes	Public_Transportation	Obesity_Type_II

Рисунок Г.1 – Дані датасету

```python
# Побудова розподілу цільової змінної
plt.figure(figsize=(10, 6))
sns.countplot(x='NObeyesdad', data=data)
plt.title('Розподіл рівнів ожиріння')
plt.xlabel('Рівні ожиріння')
plt.ylabel('Кількість')
plt.xticks(rotation=45)
plt.show()
```

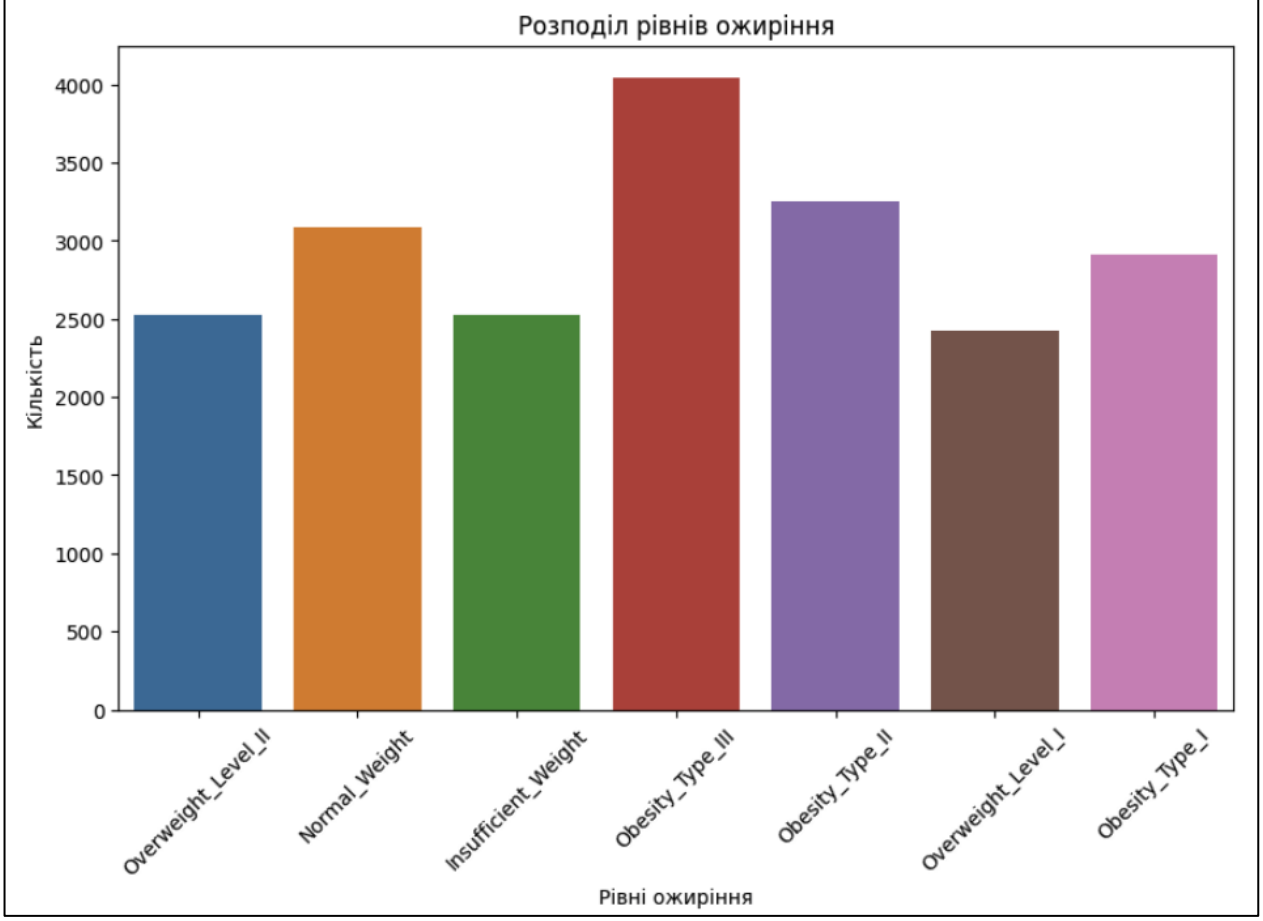

Рисунок Г.2 – Розподіл цільової змінної

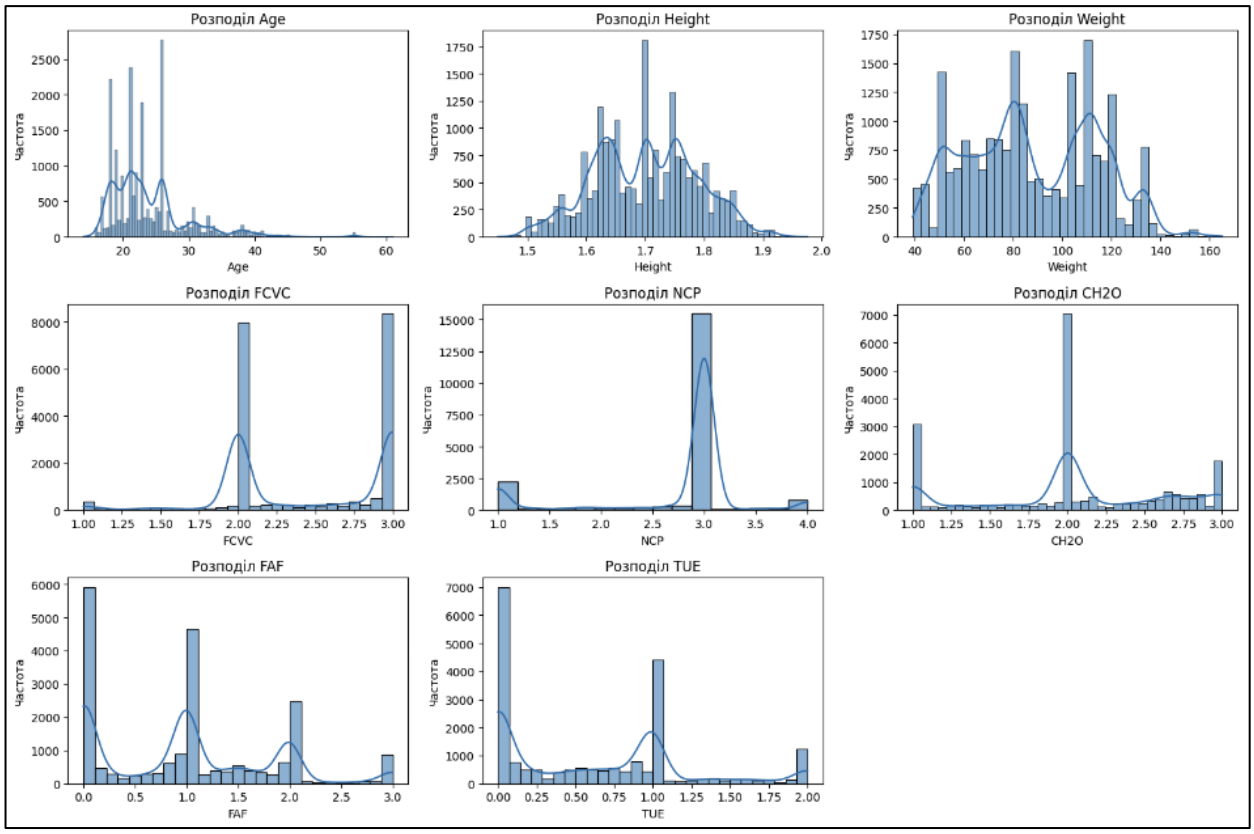

Рисунок Г.3 – Розподіл числових ознак

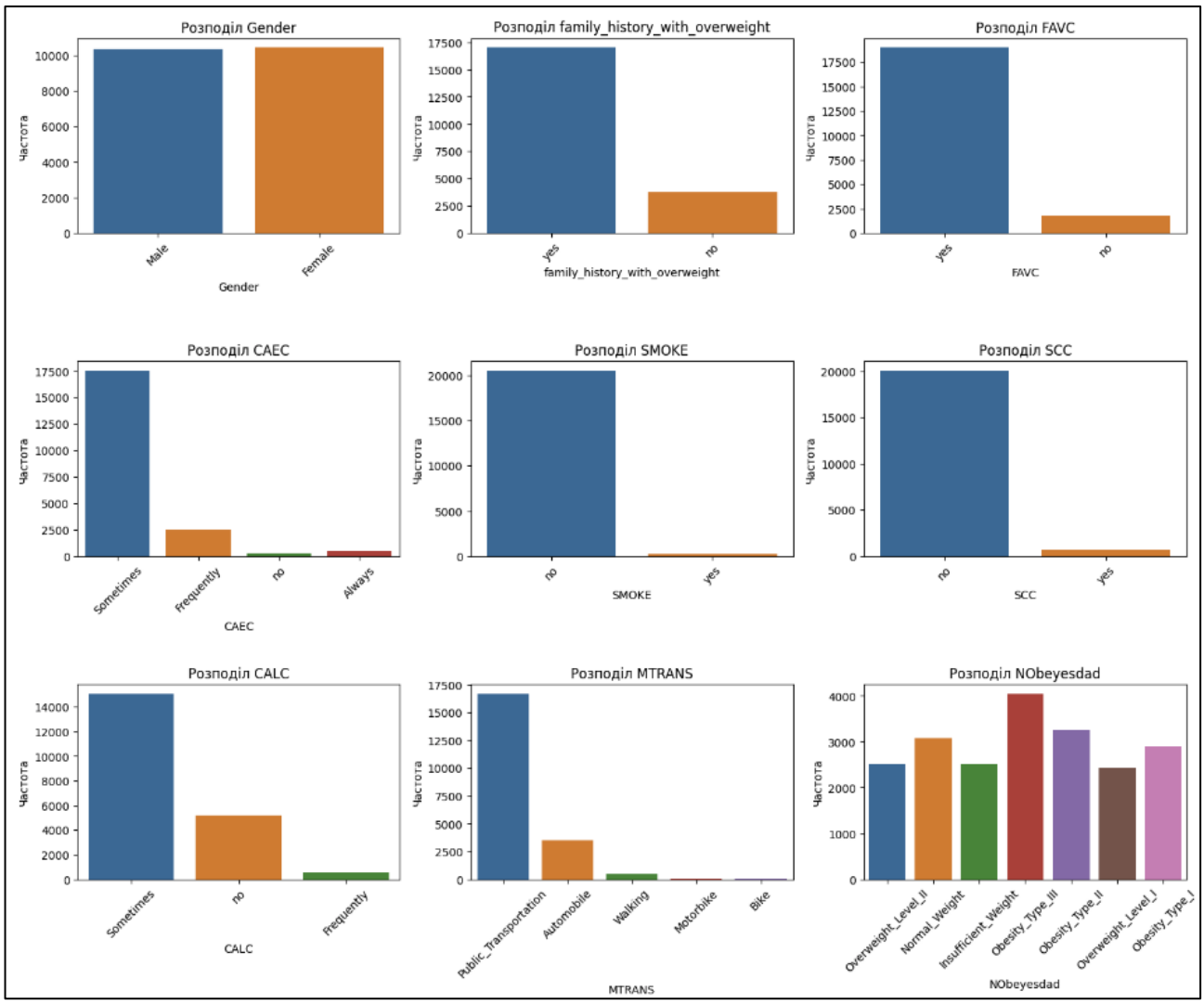

Рисунок Г.4 – Розподіл не числових ознак

Accuracy Table:

	Model	Train Accuracy	Test Accuracy
0	Logistic Regression	0.927725	0.898345
1	Decision Tree	0.986967	0.964539
2	Random Forest	0.976896	0.959811
3	SVM	0.953199	0.907801

Рисунок Г.5 – Зведена таблиця результатів моделей

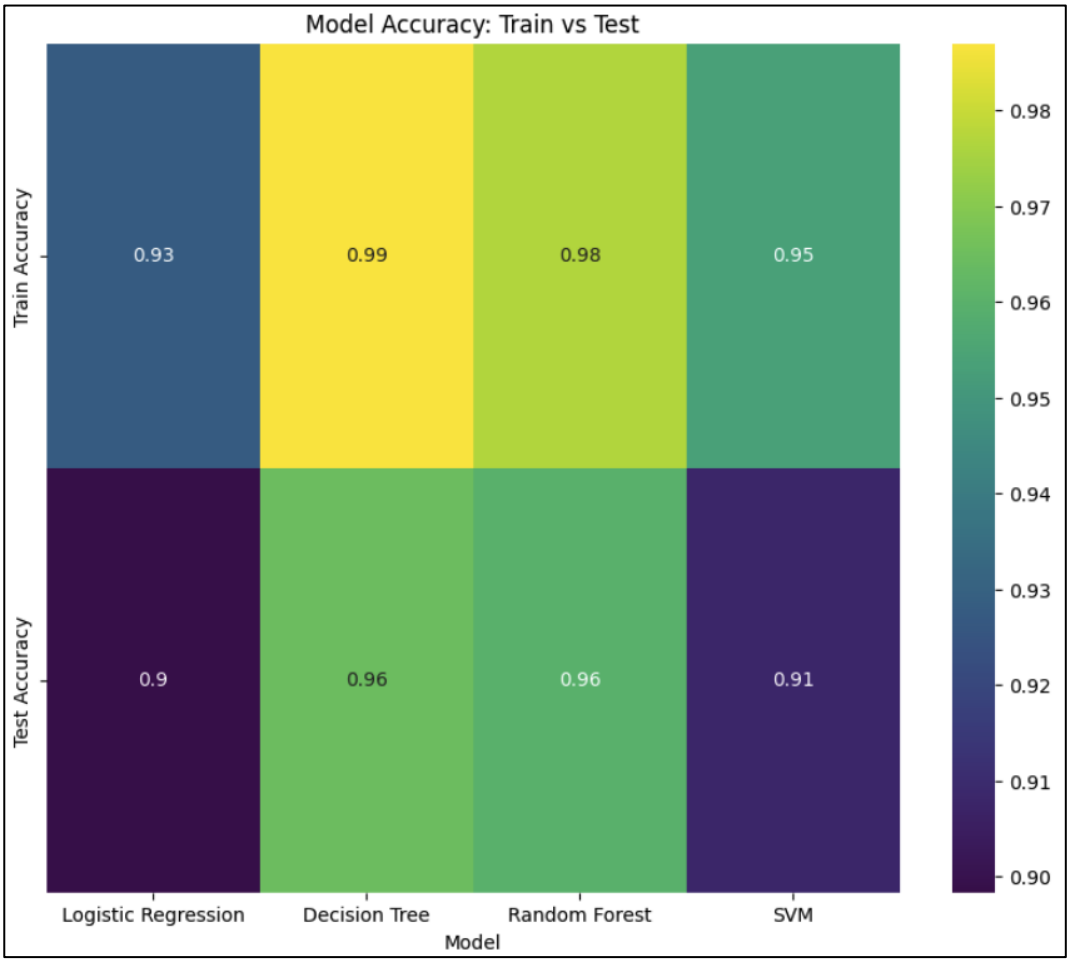

Рисунок Г.6 – Результати всіх моделей

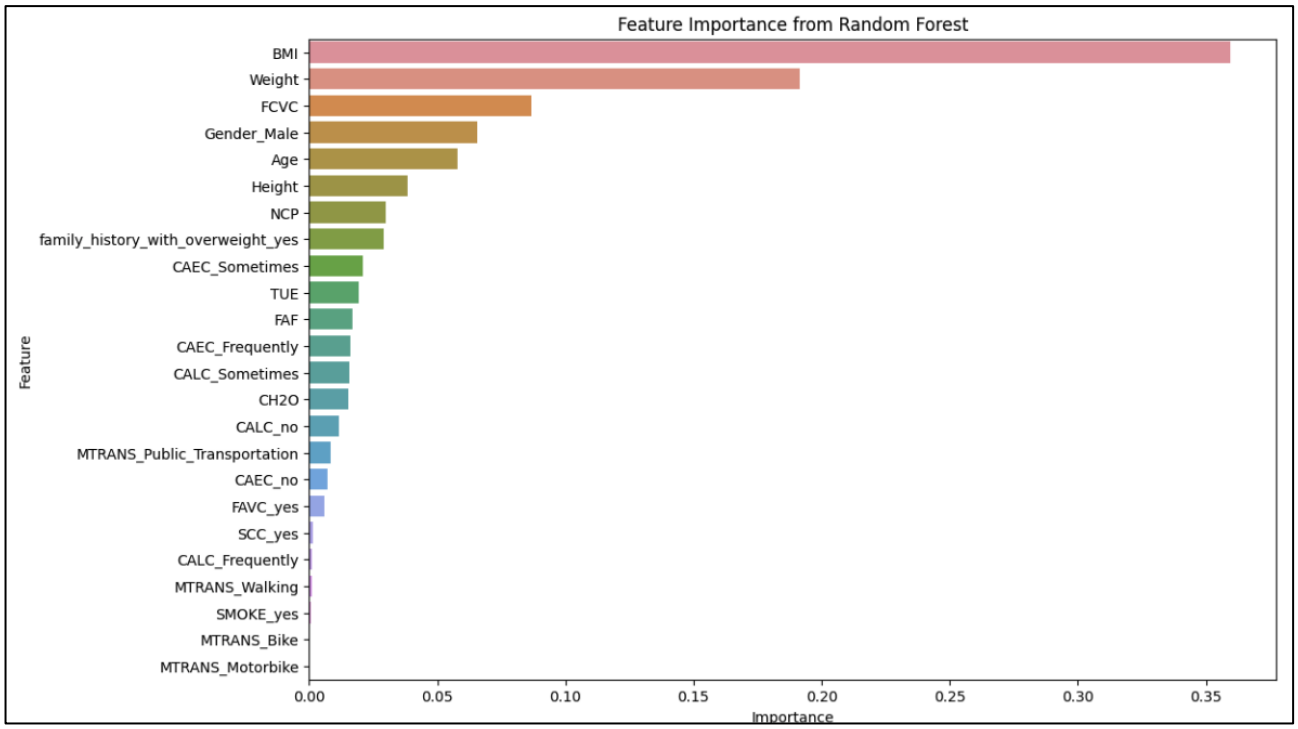

Рисунок Г.7 – Візуалізація важливості ознак для моделі Random Forest