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

Магістерська кваліфікаційна робота на тему:

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ РОЗЛАДУ СНУ ЛЮДИНИ»

Виконав: студент 2 курсу, групи 2ІСТ-22м
спеціальності 126 «Інформаційні системи та технології»
_____ Дмитро ІЩУК

Керівник: к.т.н., доц. каф. САІТ
_____ Сергій ЖУКОВ
«06» _____12_____ 2023 р.

Опонент: к.т.н., доц. каф. КН
_____ Володимир ОЗЕРАНСЬКИЙ
«14» _____12_____ 2023 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«08» _____12_____ 2023 р.

Вінниця ВНТУ – 2023 рік

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

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

д.т.н., проф. Віталій МОКІН

«01» 09 2023 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Іщук Дмитру Володимировичу

1. Тема роботи: «Інформаційна технологія аналізу та передбачення розладу сну людини»
керівник роботи: Сергій ЖУКОВ, к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «18» 09 2023 року №247
2. Термін подання студентом роботи 30.11.2023 р.
3. Вихідні дані до роботи:
   1) Kaggle Dataset „Sleep health and lifestyle 2023". URL: https://www.kaggle.com/datasets/sleep-health-and-lifestyle-dataset;
4. Зміст текстової частини:
   1) Загальна характеристика об'єкту досліджень;
   2) Вибір оптимальних моделей для розв'язання поставленої задачі;
   3) Розроблення інформаційної технології аналізу та передбачення розладу сну людини;
   4) Економічна частина.
5. Перелік ілюстративного матеріалу:
   1) Алгоритм роботи інформаційної технології;
   2) Кореляційна матриця;
   3) Діаграми характеристик з датасету
   4) Діаграми змінних з датасету;
   5) Діаграми варіабельних розладів сну по показниках гендеру та ІМТ;
   6) Результат моделі Random Forest;
   7) Крива навчання для лінійного класифікатора;
   8) Кінцевий результат навчальних та тестових даних.

6. Консультанти розділів роботи

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

7. Дата видачі завдання «04» 09 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	04.09	20.09	Вик
2	Вибір оптимальних налаштувань для розв'язання поставленої задачі	20.09	10.10	Вик
3	Розроблення інформаційної технології аналізу та передбачення розладу сну людини	10.10	30.10	Вик
4	Економічна частина	01.11	10.11	Вик
5	Оформлення матеріалів до захисту МКР	10.11	30.11	Вик

Студент _____ Дмитро ЩУК

Керівник роботи _____ Сергій ЖУКОВ

АНОТАЦІЯ

УДК 004.09

Іщук Д.В. Інформаційна технологія аналізу та передбачення розладу сну людини. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2023. 108 с.

На укр. мові. Бібліогр.: 33 назв; рис.: 35; табл.: 9.

Розроблено інформаційну технологію аналізу та передбачення роздаду сну людини. Проведено загальну характеристику об’єкту дослідження, здійснено огляд та аналіз аналогів та технологію для виявлення розладу сну людини, обґрунтовано доцільність даного дослідження. Проведено опис моделей, які використовувались для передбачення розладу сну людини, здійснено вибір мови програмування та середовища розробки. Здійснено розвідувальний аналіз даних. Проведено передбачення даних з використанням дев’яти різних моделей машинного навчання, а також візуалізацію отриманих результатів, та оцінку моделей за різними метриками.

Ілюстративна частина складається з 8 плакатів.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної технології аналізу та передбачення розладу сну людини.

Ключові слова: інформаційні технології, машинне навчання, здоровий сон, аналіз, передбачення, python.

ABSTRACT

Ishchuk D.V. Information technology of analysis and prediction of human sleep disorder. Master's thesis in specialty 126 - Information systems and technologies, educational and professional program - Information technologies of data and image analysis. Vinnytsia: VNTU, 2023. 108 c.

In Ukrainian. Bibliography: 33 titles; Figs. 35; Tables: 9.

An information technology for analyzing and predicting human sleep distribution is developed. A general characterization of the research object is carried out, a review and analysis of analogues and technology for detecting human sleep disorders is carried out, the feasibility of this study is substantiated. The models used to predict human sleep disorders are described, the programming language and development environment are selected. An exploratory analysis of the data is carried out. The data was predicted using nine different machine learning models, and the results were visualized and evaluated using various metrics.

The illustrative part consists of 8 posters.

The economic section discusses the feasibility of developing and implementing information technology for analyzing and predicting human sleep disorders.

Keywords: information technology, machine learning, healthy sleep, analysis, prediction, python.

ЗМІСТ

ВСТУП

Актуальність теми. Актуальність теми полягає в зростанні важливості інформаційних технологій (ІТ) у сучасному світі та їх використанні для аналізу та передбачення розладу сну у людини. Зараз ІТ-технології широко застосовуються в різних сферах життя, включаючи медицину та здоров'я. Відзначається, що існують різноманітні інноваційні підходи до використання технологій для аналізу та покращення якості сну людини.

Технологічні досягнення у сфері ІТ дозволяють створювати і вдосконалювати методи аналізу даних про сон, що стає важливим для здоров'я людини. Одним з перспективних напрямків в цьому контексті є використання алгоритмів машинного навчання для передбачення розладів сну та їх наслідків.

Дослідження в сфері аналізу та передбачення розладу сну має на меті розробку та вдосконалення інформаційних технологій для точного передбачення та діагностики проблем зі сном у людини.

Мета і завдання роботи. Метою дослідження є підвищення точності передбачення розладу сну людини.

Для досягнення цієї мети необхідно вирішити наступні завдання:

– провести аналіз проблеми розладу сну та визначити основні фактори, що впливають на нього;

– визначити оптимальні моделі машинного навчання для передбачення розладів сну та розробити методи їх застосування;

– розробити і впровадити інформаційну технологію, яка забезпечить аналіз та передбачення розладу сну на основі зібраних даних.

Такий підхід до використання ІТ-технологій для аналізу та передбачення розладу сну відкриває нові можливості для покращення якості життя людей шляхом ранньої діагностики та ефективного управління цими розладами [1].

Об’єктом дослідження є процес розробки інформаційної технології аналізу та передбачення розладу сну людини.

Предметом дослідження є інформаційна технологія аналізу та передбачення розладу сну людини, яка дозволятиме детально аналізувати фізіологічні, психологічні та інші фактори, які можуть впливати на якість та характер сну людини.

Новизна одержаних результатів. Дістала подальший розвиток інформаційна технологія аналізу та передбачення розладу сну, за рахунок використання методів машинного навчання, що дозволило підвищити точність передбачення.

Практичне значення. Практична цінність цього дослідження полягає в покращенні точності передбачення розладу сну людини. Отримані результати є важливими для розробки ефективних стратегій у медичній та науковій галузях, спрямованих на покращення якості сну та здоров'я.

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

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано тези на LIII Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2023-2024 рр.) [1].

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ

1.1 Аналіз предметної області

Об'єктом дослідження є сон, його важливість та процеси, пов'язані з фізіологією та якістю сну у людини. Сучасне суспільство стикається з рядом факторів, таких як стрес, сидячий спосіб життя і різні зовнішні впливи, які значно впливають на якість сну. Сон відіграє критичну роль у фізичному та психічному здоров'ї людини, є необхідним для відновлення організму та підтримання оптимального рівня функціонування.

Сон – це природний біологічний процес, під час якого відбуваються важливі фізіологічні й психологічні процеси, такі як відновлення тілесних функцій, підсилення пам'яті, регулювання гормонального балансу та зменшення стресу. Неправильний сон або його недостатність може призвести до різноманітних проблем зі здоров'ям та погіршення якості життя [2].

Моніторинг сну набуває наростаючого значення в сучасному світі, оскільки дозволяє відстежувати та аналізувати якість та режим сну людини. Сучасні технології, такі як портативні пристрої, смартфони або сенсорні пристрої для сну, надають можливість збирати докладну інформацію про звички під час сну. Це дозволяє виявляти шаблони та фактори, що впливають на якість сну, і допомагає коригувати їх для поліпшення якості сну та загального стану організму.

Безперечно, збір та аналіз даних про сон завдяки інформаційним технологіям може відкрити нові можливості для розуміння впливу сну на загальне здоров'я та добробут людини. Зокрема, інтеграція цих технологій у медичні практики може сприяти ранньому виявленню та діагностиці різних захворювань, пов'язаних із сном, та дозволить розробляти персоналізовані підходи до лікування [2].

Додатки та пристрої для моніторингу сну можуть надати корисну інформацію для лікарів у виявленні порушень сну, які можуть бути важливими симптомами різних медичних станів. Наприклад, регулярний аналіз даних про сон може допомогти виявити симптоми апное сну, що може бути поруч із проблемами

з серцем та дихальною системою. Завдяки цьому лікарі матимуть більше інформації для прийняття обґрунтованих рішень у лікуванні пацієнтів.

Більше того, інформація, зібрана внаслідок моніторингу сну, може допомогти у розробці індивідуальних рекомендацій для поліпшення сну та здоров'я. Особисті дані про сон можуть бути використані для створення персоналізованих стратегій сну, які включають оптимальні режими сну, підходящі методи релаксації перед сном та корекцію звичок для покращення якості сну [3].

Отже, використання інформаційних технологій у моніторингу та аналізі сну відкриває нові можливості для розуміння важливості сну для здоров'я та може сприяти розвитку більш ефективних підходів до діагностики та лікування різних розладів сну.

Розробка та застосування методів машинного навчання у сфері аналізу сну відкриває широкі перспективи для вдосконалення процесу моніторингу та управління якістю сну.

Машинне навчання дозволяє аналізувати великі обсяги даних про сон та ідентифікувати складні зв'язки та патерни, які можуть бути складно виявлені за допомогою традиційних методів аналізу. Застосування алгоритмів машинного навчання допомагає у виявленні залежностей між факторами сну та його якістю, враховуючи різноманітні параметри, такі як тривалість сну, час засинання, глибина сну тощо [3].

Наприклад, моделі машинного навчання можуть аналізувати вплив різних факторів, таких як активність перед сном, температура в кімнаті, погодні умови, на якість сну. Це дозволяє зрозуміти, які чинники найбільше впливають на задоволеність сном та визначити специфічні рекомендації для кожної особи.

Технології машинного навчання також можуть допомогти у розробці персоналізованих планів для кожної людини. Аналізуючи історичні дані про сон та поведінку, алгоритми можуть навчитися передбачати оптимальні режими сну для кожної людини та розробляти рекомендації для його покращення [4].

Усе це відкриває перспективи для вдосконалення та персоналізації методів лікування розладів сну, а також надає можливість кожній особі більш ефективно

контролювати та покращувати свій режим сну за допомогою високотехнологічних рішень.

Розробка та застосування методів машинного навчання у сфері аналізу сну відкриває широкі перспективи для вдосконалення процесу моніторингу та управління якістю сну.

Застосування алгоритмів машинного навчання допомагає у виявленні залежностей між факторами сну та його якістю, враховуючи різноманітні параметри, такі як тривалість сну, час засинання, глибина сну тощо.

Наприклад, моделі машинного навчання можуть аналізувати вплив різних факторів, таких як активність перед сном, температура в кімнаті, погодні умови, на якість сну. Це дозволяє зрозуміти, які чинники найбільше впливають на задоволеність сном та визначити специфічні рекомендації для кожної особи.

Технології машинного навчання також можуть допомогти у розробці персоналізованих планів для кожної людини. Аналізуючи історичні дані про сон та поведінку, алгоритми можуть навчитися передбачати оптимальні режими сну для кожної людини та розробляти рекомендації для його покращення [4].

Усе це відкриває перспективи для вдосконалення та персоналізації методів лікування розладів сну, а також надає можливість кожній особі більш ефективно контролювати та покращувати свій соновий режим за допомогою високотехнологічних рішень.

Розвиток і вдосконалення технологій машинного навчання дозволяє виявляти складні патерни в поведінці людини під час сну та розуміти їх зв'язок з різними фізіологічними та психологічними аспектами здоров'я [4].

Одним з напрямків застосування машинного навчання є створення моделей прогнозування ефективності різних лікувальних методик для конкретних типів розладів сну. Шляхом аналізу даних про сон та відповіді пацієнтів на різні методи лікування, можливо розробити персоналізовані підходи до терапії, що максимально відповідають потребам кожного конкретного пацієнта.

Додатково, використання машинного навчання може сприяти в удосконаленні пристроїв для моніторингу сну. Аналізуючи великі обсяги даних,

отриманих від різноманітних сенсорів, таких як акселерометри, електрокардіографи, термометри тощо, можна розробити більш точні та надійні пристрої для вимірювання параметрів сну та оцінки його якості [5].

Засоби моніторингу сну, такі як носимі пристрої та сенсори, стали популярними інструментами для вимірювання сну. Вони дозволяють користувачам отримувати детальну інформацію про свій сон, включаючи тривалість, структуру сну (фази REM та не REM), частоту прокидань під час ночі і якість дихання. Ці дані допомагають розуміти власні сонові звички та вживати кроки для їх поліпшення. Більше того, вони можуть бути корисні для лікарів і дослідників при вивченні сону та його впливу на здоров'я.

Сучасний світ і технології дуже впливають на якість сну людей. Електронні пристрої, такі як смартфони та планшети, можуть викликати затяжний синдром синього світла, який порушує природний циркадний ритм і гальмує виділення мелатоніну, гормону сну. Це може призвести до проблем зі засинанням і глибокість сну. Споживання кави і алкоголю, особливо в пізні вечори, також може негативно впливати на якість сну [5].

Тема здорового сну та способу життя надзвичайно важлива в контексті загального стану здоров'я і попередження різних хронічних захворювань. Недостатній сон асоціюється із збільшеним ризиком розвитку ожиріння, діабету, серцево-судинних захворювань та інших проблем. Дотримання здорового способу життя, включаючи правильний режим сну, дієту та фізичну активність, може значно підвищити якість життя та продовжити тривалість життя.

Загалом, вивчення сну та способу життя стає більш важливим у світлі сучасних викликів і може призвести до розробки ефективних стратегій для підтримки здорового сну і способу життя, що в свою чергу позитивно впливає на фізичне та психічне здоров'я людини.

Це дозволить не лише збирати більше точних даних про сон, а й розширити можливості для розуміння факторів, що впливають на якість сну, та розробки нових підходів до його покращення. Такі технології можуть стати допоміжним

інструментом як для індивідуального контролю за сном, так і для клінічної медицини у виявленні та лікуванні розладів сну [5].

1.2 Переваги та обмеження технологій

Розвиток і вдосконалення технологій машинного навчання дозволяє виявляти складні патерни в поведінці людини під час сну та розуміти їх зв'язок з різними фізіологічними та психологічними аспектами здоров'я.

Одним з напрямків застосування машинного навчання є створення моделей прогнозування ефективності різних лікувальних методик для конкретних типів розладів сну. Шляхом аналізу даних про сон та відповіді пацієнтів на різні методи лікування, можливо розробити персоналізовані підходи до терапії, що максимально відповідають потребам кожного конкретного пацієнта [6].

Додатково, використання машинного навчання може сприяти в удосконаленні пристроїв для моніторингу сну. Аналізуючи великі обсяги даних, отриманих від різноманітних сенсорів, таких як акселерометри, електрокардіографи, термометри тощо, можна розробити більш точні та надійні пристрої для вимірювання параметрів сну та оцінки його якості.

Це дозволить не лише збирати більше точних даних про сон, а й розширити можливості для розуміння факторів, що впливають на якість сну, та розробки нових підходів до його покращення. Такі технології можуть стати допоміжним інструментом як для індивідуального контролю за сном, так і для клінічної медицини у виявленні та лікуванні розладів сну [6].

Однією з проблем може бути неоднозначність та інтерпретація даних про сон. Інформаційні технології можуть забезпечити обширний обсяг даних, але їх аналіз та визначення впливу конкретних параметрів на якість сну може бути складним. Різні люди можуть мати різні індивідуальні особливості та реагувати по-різному на ті ж самі фактори, що ускладнює універсальний підхід до аналізу даних про сон.

Іншим обмеженням є приватність та безпека даних. Збір чутливих медичних даних про сон може створювати питання про їхню безпеку та конфіденційність. Запевнення захисту цих особистих даних від несанкціонованого доступу та використання є важливою проблемою, особливо в контексті використання медичних пристроїв та додатків для моніторингу сну.

Крім того, однією з ключових проблем може бути доступність та фінансова вартість сучасних технологій моніторингу сну. Деякі пристрої або додатки можуть бути дорогими для більшої частини споживачів, що обмежує їхню доступність та застосування в широкому масштабі.

Ще однією ключовою проблемою використання інформаційних технологій для аналізу сну є висока складність алгоритмів та моделей машинного навчання у визначенні та передбаченні розладів сну. Навіть за наявності значного обсягу даних про сон та його параметри, розробка точних та надійних моделей для передбачення розладу сну є складним завданням. Це пов'язано з великою кількістю факторів, що впливають на якість сну, та їх взаємозв'язком, який може бути складним для моделювання.

Крім того, недостатня кількість або недостатня якість даних можуть ускладнити розробку точних моделей для передбачення розладів сну. Якщо дані, які збираються пристроями моніторингу сну, є неповними або мають деякі помилки, це може вплинути на якість та достовірність аналізу. Недоліки у зборі та обробці даних можуть викликати неточності та спотворення результатів [6].

Також, слід враховувати, що машинне навчання для аналізу сну є досить новим напрямком досліджень. Часто виникає проблема інтерпретації результатів моделей машинного навчання, які можуть бути складними для розуміння та пояснення, особливо для фахівців, які не мають глибоких знань в галузі машинного навчання або обробки даних.

Таким чином, використання інформаційних технологій для аналізу сну людини може стикатися зі складними алгоритмічними, технічними та методологічними викликами, які потребують подальших досліджень та розвитку для ефективного застосування в медицині та покращення якості сну людей.

Тому інформаційні технології створюють значний потенціал для аналізу та передбачення розладів сну, вони також мають свої обмеження, які потребують уваги та вирішення для максимального використання їх можливостей у цій сфері.

Іншою проблемою є неоднозначність та недостатня стандартизація у визначенні параметрів сну. Існує ряд медичних та психологічних критеріїв, які використовуються для оцінки якості сну. Однак, недостатньо узгоджених стандартів у визначенні цих параметрів може ускладнити спільне розуміння результатів дослідження та порівняння даних між різними дослідниками.

Додатково, варіативність індивідуальних характеристик сну та різниця у вимірювальних технологіях можуть призвести до різних визначень та інтерпретації даних. Наприклад, один параметр, такий як глибина сну або кількість рухів, може бути виміряний різними пристроями з різною точністю, що може призвести до різних результатів інтерпретації.

Також варто враховувати, що не всі види розладів сну можуть бути точно виявлені чи передбачені за допомогою інформаційних технологій. Деякі види розладів сну можуть мати складніші або менш очевидні симптоми, що ускладнює їхнє діагностування за допомогою сучасних технологій.

Інструменти експлораторного аналізу даних (EDA) є важливою складовою при аналізі даних про сон людини. EDA дозволяє докладно дослідити вихідні дані, розглянути їхню структуру та знайти ключові показники, які можуть бути важливими для подальшого аналізу.

Один з методів EDA – це візуалізація даних, яка дозволяє побудувати графіки, діаграми та інші візуальні представлення даних. В контексті аналізу сну, це може включати графіки, які відображають зміни показників сну протягом різних часових періодів або їхній взаємозв'язок з іншими факторами, такими як фізична активність чи стрес [6].

Інший метод EDA - статистичний аналіз, який дозволяє описати структуру даних та виявити кореляції між різними параметрами сну. Наприклад, аналіз кореляції може вказати на зв'язок між тривалістю сну та якістю сну, або визначити, які параметри можуть бути більш важливими для загального здоров'я.

Аналіз також включає обробку даних, яка включає в себе очищення, видалення дублікатів та виправлення помилок. У випадку аналізу сну, це може включати обробку даних, зібраних від сенсорів чи моніторів сну, для забезпечення точності результатів аналізу.

Обробка EDA відіграє важливу роль у розумінні параметрів сну та їх взаємозв'язку, а також може допомогти виявити особливості в сні для різних груп людей. Однак, успішність та точність аналізу може залежати від якості та обсягу доступних даних про сон.

Експлораторний аналіз даних (EDA) має кілька переваг, які допомагають у розумінні та візуалізації великого обсягу інформації. Візуальне представлення даних через графіки, діаграми та інші графічні інструменти сприяє легшому розумінню та сприйняттю важливої інформації. Це дозволяє аналітикам швидше виявляти зв'язки, тенденції та патерни у даних про сон, допомагаючи виокремити ключові фактори, які впливають на якість сну. Крім того, EDA дозволяє проводити попередній аналіз даних, що сприяє формулюванню гіпотез та стратегій для подальшого дослідження [6].

Проте EDA має обмеження. Суб'єктивність візуального аналізу може призвести до різноінтерпретованих результатів, оскільки сприйняття та інтерпретація графіків може бути суб'єктивною та залежати від індивідуального дослідника. Це може призвести до розбіжностей у сприйнятті даних та відхилень в результаті.

Додатково, результати EDA можуть бути необ'єктивними та не точними у випадку використання неповних, недостовірних або неякісних даних про сон. Також, виявлені тенденції або кореляції можуть бути помилково сприйняті як причинно-наслідкові зв'язки без проведення додаткових методів перевірки.

Тому EDA є корисним інструментом для першого огляду даних, важливо враховувати його обмеження та використовувати в поєднанні з іншими методами аналізу, щоб отримати більш об'єктивні та точні результати.

Отже, неоднозначність у визначенні параметрів сну, індивідуальні відмінності та різноманітність технологій можуть стати обмеженнями

використання інформаційних технологій для аналізу та передбачення розладів сну людини.

1.3 Огляд аналогів та існуючих технологій для виявлення розладу сну людини

Якісний сон має вирішальне значення для здоров’я та благополуччя людини. Розлади сну, такі як обструктивне апное уві сні, мають високу поширеність у всьому світі і значно погіршують якість життя та спричиняють серцево-судинні захворювання та смертність. Більшість розладів сну можна контролювати та запобігати за допомогою своєчасної діагностики. Полісомнографія (ПСГ) під час сну є золотим медичним стандартом для діагностики розладів сну [7].

Згідно з правилами підрахунку балів Rechtschaffen і Kales і Американської академії медицини сну, рекорд PSG поділено на 30-секундні періоди. Кожна епоха оцінюється як одна зі стадій сну: неспання, стадія швидких рухів очей (REM) і три стадії без REM (N1 або N2: легкий сон і N3: глибокий сон). Стадії сну пов'язані з фізичним відновленням, консолідацією пам'яті та підвищенням імунітету. Структура сну зазвичай розглядається як цикли сну та послідовності стадій протягом ночі. Моніторинг стадій сну та аналіз структури сну є одними з основних підходів до діагностики розладів сну та оцінки ефекту від лікування.

Щоб отримати PSG, до людини потрібно приєднати кілька датчиків, зокрема електроенцефалограму (ЕЕГ), електрокардіограму (ЕКГ), нагрудну стрічку та периферичну пульсоксиметрію. Ці прив’язані датчики нав’язливо впливають на людей і потенційно заважають їм заснути. Інтрузивний вплив надалі негативно вплине на структуру сну та результати оцінки якості сну [7].

Крім того, фінансові витрати, витрата часу та сувора робота PSG також обмежують його використання та доступність довгострокового моніторингу сну в нелікарняних установах. Діагностичний підхід до обструктивного апное сну має перейти від лабораторних досліджень до контролю сну вдома.

В останні роки були розроблені розумні пристрої, які забезпечують ефективний, комфортний і зручний моніторинг сну. Вчені розробили носимий актіграфічний датчик, і було досягнуто високої точності понад 90% оцінки стану пробудження та сну порівняно з ручними результатами PSG.

Науковці використовували вбудовані акустичні датчики на смартфонах, щоб виявити тонку фазу сну. Ефективність триетапної класифікації сну склала близько 64,5%.

Розробники оцінили систему моніторингу сну під матрацом для безконтактного аналізу сну або неспання та повідомили про точність 77,5%. Однак ці методи потребують подальшого вдосконалення, перш ніж їх можна буде прийняти для потенційного практичного застосування [8].

Еволюція сенсорних технологій призвела до появи бездротових систем для моніторингу фізіологічних сигналів без контакту з тілом або через прив'язку.

У цих системах радарні датчики передають малопотужний радіочастотний (РЧ) сигнал і виділяють життєво важливі показники людини з відбитого сигналу. Було продемонстровано, що кардіореспіраторні сигнали, зафіксовані радарами, можна використовувати для моніторингу сну.

За останнє десятиліття дослідження продемонстрували велику доцільність використання радарних датчиків для прогнозування стадій сну. Радарні вимірювання також надають цінну інформацію для оцінки якості сну .

У порівнянні з традиційними методами PSG, такі безконтактні системи дешеві, зручні для користувача та зручні в налаштуванні, що забезпечує більший потенціал для тривалого моніторингу фази сну. У дослідженні Chazal та інших, був розроблений новий безконтактний датчик біоруху для ідентифікації моделей сну або неспання для дорослих, шляхом виявлення рухів та ідентифікації моделей дихання. Загальна чутливість уві сні становила 87,3%, а чутливість у стані неспання – 50,1% порівняно із золотим стандартом PSG [8].

У дослідженні Tataraidze et al. ступінчасто-частотний біомедичний радар використовувався для триступеневої класифікації (неспання, REM і NREM) на

основі циклічних особливостей дихальних рухів і середньої точності Виконано 75,1% [8].

Інші науковці взяли до уваги сигнали, пов’язані зі сном, включаючи дихання, серцебиття та рухи тіла. Алгоритм підпростору k-найближчого сусіда перевершив чотириетапне прогнозування (неспання, швидкий сон, глибокий сон і легкий сон). Однак метод із використанням ручних функцій і класифікатора сильно залежав від знань дослідників і був чутливим до невідомого шуму. Тому важко застосувати ці методи до інших нових методів зондування та нових середовищ тестування.

В аналізі сну за допомогою PSG підходи глибокого навчання показали високу точність стадій сну завдяки усуненню ручних ознак. Однак аналіз сну на основі глибокого навчання та радіочастотного сигналу все ще залишається на ранній стадії розвитку. Лише деякі дослідження використовують і порівнюють різні структури моделей глибокого навчання, що здійснюються з використанням радіочастотних сигналів [9].

Тому, існуючі методи зосереджені виключно на вивченні інформативних ознак з радіочастотних сигналів і прогнозуванні ознак на поточному кроці. Сильна перехідна структура станів сну не була ретельно врахована. Таким чином, важлива інформація про динаміку може бути проігнорована вищезгаданими підходами, що базуються на навчанні.

Існує безконтактна система прогнозування структури сну (NSSP) для визначення стадій сну. NSSP використовує фізіологічні атрибути дихання, серцебиття та рухів тіла, зафіксовані радіолокаційними датчиками. Модель нейронної мережі використовується для поєднання згорткових шарів, рекурентних шарів і нейронних умовних випадкових полів (CRF) для навчання і фіксації часової інформації та опису переходу від стадії до стадії сну. Надійність і переносимість запропонованого методу підтверджено експериментами на різних радіочастотних пристроях і різних об'єктах [9].

У дослідженні на людях було отримано середню точність 75,3% для різних стадій сну на зразках 60 ГГц, а середня точність 79,2% і каппа Коена 0,679 були досягнуті при перехресній перевірці на наборі даних радіочастотного сну 6 ГГц. Порівняно з

методами, що базуються на інших розумних пристроях та ручному методі машинного навчання, NSSP демонструє вищу точність виявлення та оцінки.

За результатими роботи було продемонстровано можливість широкого застосування в середовищі "розумного" будинку для проведення щоденного аналізу структури сну. Основний внесок роботи полягає в новій трирівневій системі моніторингу сну з радіолокаційними датчиками міліметрового діапазону хвиль і новій схемі прогнозування переходу до фаз сну на основі машинного навчання [?].

Система NSSP циклічно проходить через стадії сну (тобто, неспання, швидку та повільну) протягом ночі, і кожна стадія має унікальну функцію, тобто, стадія REM пов'язана з консолідацією пам'яті, а стадія N3 дозволяє м'язам людини повністю розслабитися. Різні функції стадій призводять до зміни життєво важливих показників, як показано на рисунку 1a. Коли сон поступово поглиблюється в стадіях N1, N2 і N3, дихання і серцебиття сповільнюються, а величина рухів тіла стає меншою і рідшою.

У фазі швидкого сну серцебиття зазвичай стає швидшим і менш ритмічним, ніж в інших фазах. Апаратний дизайн має трирівневу архітектуру, радарні датчики, точку доступу та хмару, як показано на рисунку 1b. Система розроблена для моніторингу сну в режимі реального часу за допомогою радіолокаційного датчика, розміщеного навколо ліжка.

Багатоканальний радар 60 ГГц використовується для передачі безперервних хвиль і прийому сигналів, відбитих від об'єктів. Дихання людини, серцебиття та рухи кінцівок спричиняють зміну відбитого радіочастотного сигналу.

Порівняно з попередніми радіочастотними пристроями, радарний датчик, що використовується в NSSP, має менший розмір, вузьку ширину променя та високе співвідношення сигнал/шум.

Крім того, сигнали датчика стійкі до перешкод, що генеруються іншими пристроями бездротового зв'язку, розташованими поблизу (як показано на Малюнку S1a,b, "Допоміжна інформація"). Система NSSP (Рис.1.1) ілюструє прогнозування стадій нічного сну за допомогою технології.

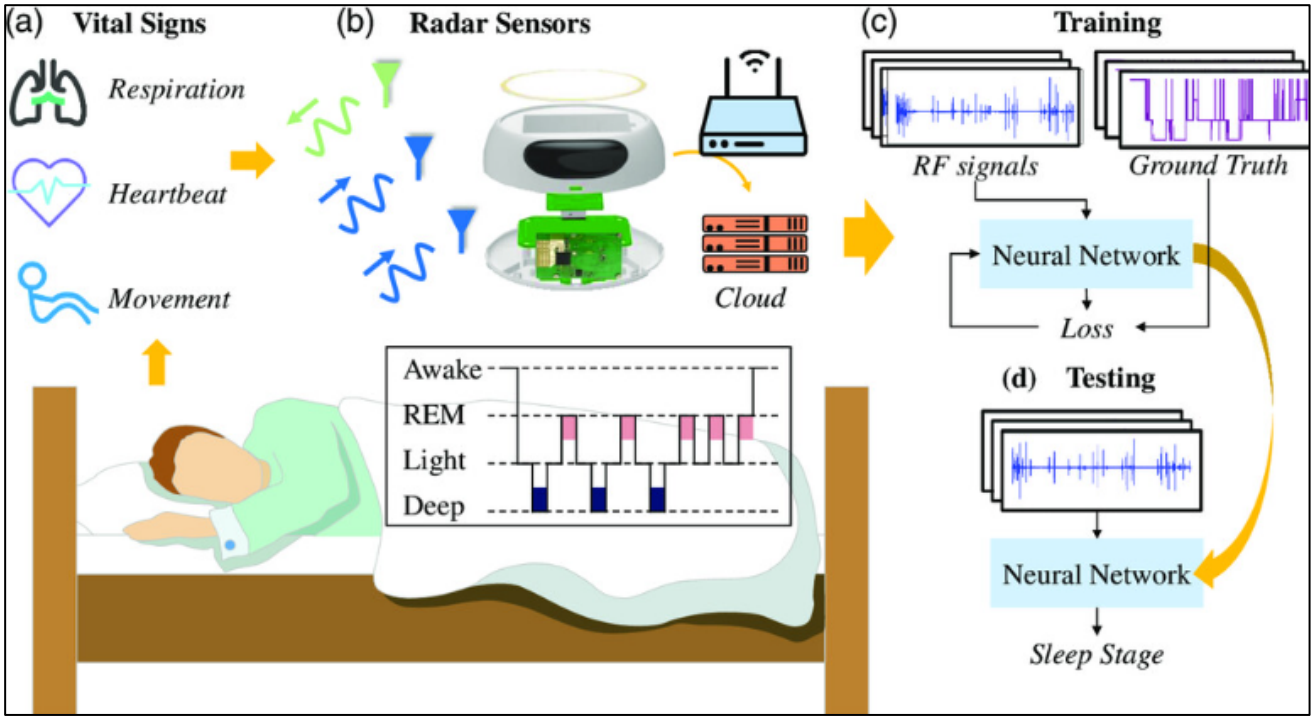


Рисунок 1.1 – Принцип роботи системи NSSP

Робочий процес системи NSSP.

a) Стадії сну з різними функціями спричиняють варіації дихання, серцебиття та рухів тіла.

b) Трирівнева радіолокаційна система використовується для передачі та прийому радіочастотних сигналів та вимірювання життєво важливих ознак і рухів.

c) NN-модель навчається для отримання знань про перебіг сну.

d) Навчена NN-модель прогнозує стадію сну, використовуючи захоплені радіочастотні сигнали.

Щоб отримати набір даних для навчання моделі та прогнозування, зразки сну записуються безперервно протягом ночі (наприклад, 8 годин). Кожен зразок бездротовим способом надсилається до хмарної бази даних. Після того, як безконтактна радіолокаційна система зафіксувала нічний сон суб'єкта, вона розбивається на навчальний і тестовий набори даних [9].

Завдання прогнозування стадій сну полягає в тому, щоб позначити кожну 30-секундну епоху міткою. Кожна мітка відповідає одній з чотирьох стадій: неспання, легкий сон (N1 або N2), глибокий сон (N3) та швидкий сон. Конкретна постановка задачі сформульована у допоміжній інформації.

На вході обчислювальної задачі радіочастотні сигнали різних частот, ймовірно, мають різну просторову роздільну здатність і коефіцієнти відбиття. Для усунення специфічних відмінностей датчиків і життєво важливих ознак, компоненти в радіосигналах виділяються і нормалізуються (див. детальний опис попередньої обробки радіосигналів у допоміжній інформації).

Як показано на рисунку 1.2. Вводиться NN-модель для реалізації прогнозування стадій сну. Вона складається зі згортково-рекурентної нейронної мережі (CRNN) для вивчення латентного представлення, механізму багатоканального голосування для оцінки надійності каналів і шару CRF для моделювання процесу переходу до стану сну [9].

Архітектура згорткової нейронної мережі (CNN) – рекурентної нейронної мережі (RNN) розроблена як кодер для вилучення незмінних у часі ознак з кожної 30-секундної епохи і навчання послідовностей серій епох. CNN використовується для захоплення інформації різних часових масштабів, і її структура проілюстрована на рисунку 1.2. Натхненні технологією обробки сигналів, використовується два ядра 1D згортки різного розміру в першому шарі для вилучення часової інформації з різночастотних компонентів. CNN складається з 16 шарів 1D згортки, і кожен шар 1D згортки є послідовністю своїх фільтрів, пакетної нормалізації та випрямленої лінійної активації одиниць.

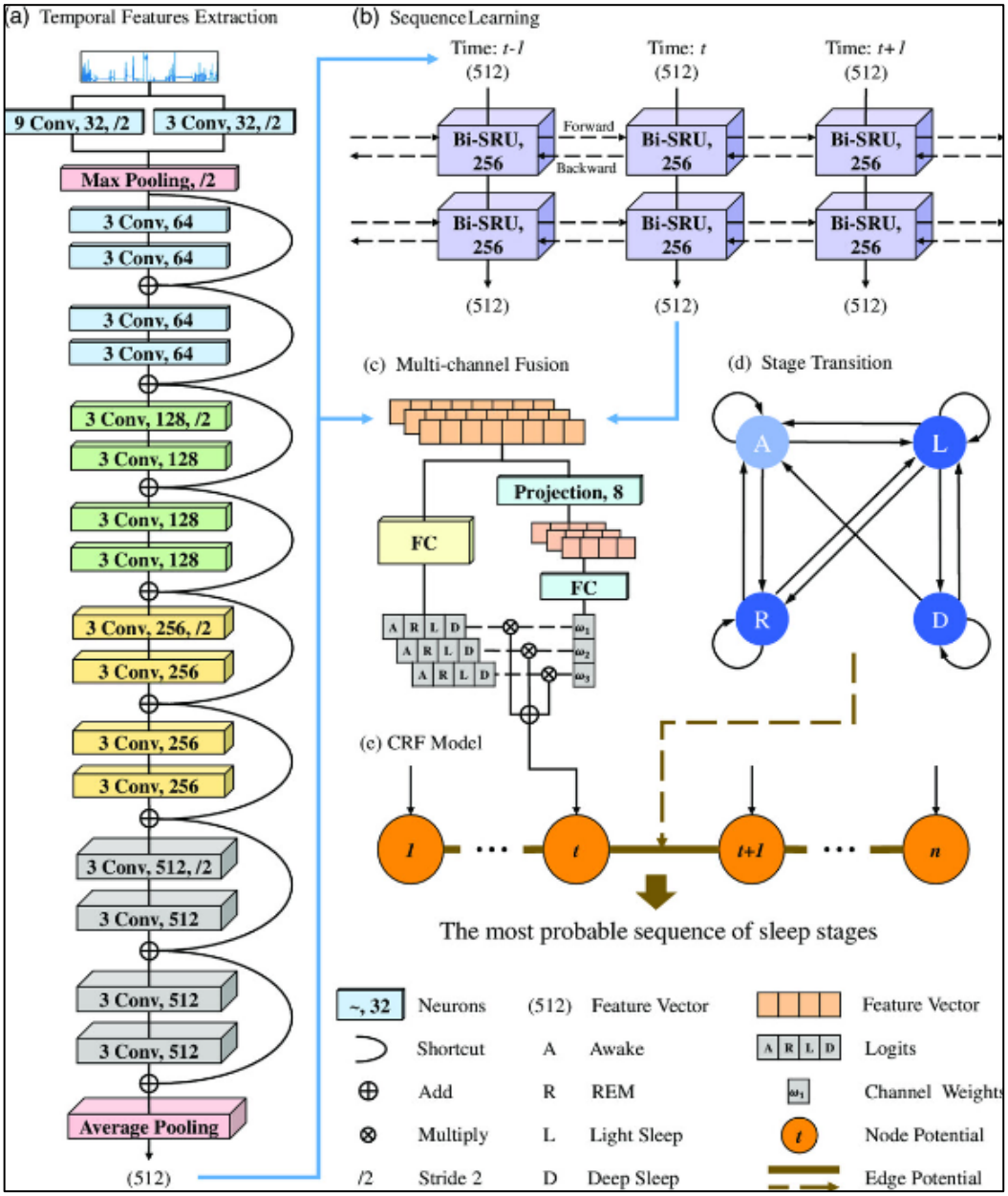


Рисунок 1.2 – NN-модель для реалізації прогнозування стадій сну

a) CNN для вилучення часової інформації в 30-секундних епохах.

b) RNN-модуль для вивчення послідовних знань.

c) Механізм багатоканального голосування для оцінки логів класів.

d) CRF-шар для моделювання процесу переходу зі стану сну в стан сну.

Як показано на рисунку 1.2, структура RNN, прості рекурентні одиниці (SRU), використовується для фіксації динаміки ознак і вивчення правил переходу ознак [9].

Наприклад, стадія легкого сну триває близько півгодини за цикл, а потім організм зазвичай занурюється в глибокий сон. У цьому випадку радіочастотні ознаки життєдіяльності під час легкого та глибокого сну схожі між собою, тоді як CNN може відрізнити їх від неспання або швидкого сну. Далі RNN здатний навчитися запам'ятовувати історію сну і передавати її підсумок в поточну комірку.

Шар CRF розглядається як глобальний предиктор в кінці NN-моделі. В результаті ефекту багатопроменевості та шуму, що спостерігається в різних радіочастотних каналах, вилучені ознаки, ймовірно, відрізняються від каналу до каналу [9].

Для оцінки кожного каналу використовується механізм голосування, який відображає його відносну важливість. Кодер CNN-RNN витягує найкращі ознаки з радіосигналів, і передбачення стадії сну залежить від кожного часового кроку. Однак відомо, що переходи між стадіями сну мають сильну структуру залежності [10].

Наприклад, стадія глибокого сну не може бути досягнута без проходження стадії легкого сну. Ця структура переходів для точного визначення стадій сну показана на рисунку 1.2. Використовуючи модель CRF, і спільну умовну ймовірність, можна досягти глобально оптимального рішення щодо стадій сну.

Результати прогнозування стадій сну. Що стосується реальних застосувань, то дані нічного сну зазвичай надходять від нових людей, нових середовищ і навіть нових пристроїв. Існують відмінності у фізіологічних особливостях стадій сну між різними суб'єктами, віком, станом здоров'я тощо. Щоб перевірити класифікаційну здатність NSSP, проводиться перевірка навчання з перенесенням на різних сенсорах та людях.

Для оцінки продуктивності обчислюється точність прогнозування та каппа Коена к. Загалом, метод який описувався вище має загальну точність 79,2% і значення к = 0,679. Він продемонстрував здатність засвоювати знання про структуру сну з навчальної вибірки та видобувати інформацію з радіочастотних сигналів. На рисунку 1.3 показано приклад структури сну, позначеної експертами

як істинна і передбачена моделлю. Точність прикладу близька до середнього значення. Також показано загальну матрицю плутанини.

Точність прогнозування для стадій неспання, швидкого, легкого та глибокого сну становить 66%, 80%, 83% та 76% відповідно. Найвищу точність має ідентифікація легкого сну. Найгірші показники для стадії неспання.

Стадію неспання здебільшого помилково ідентифікують як стадію швидкого або повільного сну. Інші помилкові класифікації в основному стосуються стадій швидкого та глибокого сну. Ці дві стадії не належать до швидкого сну і мають схожі фізіологічні особливості.

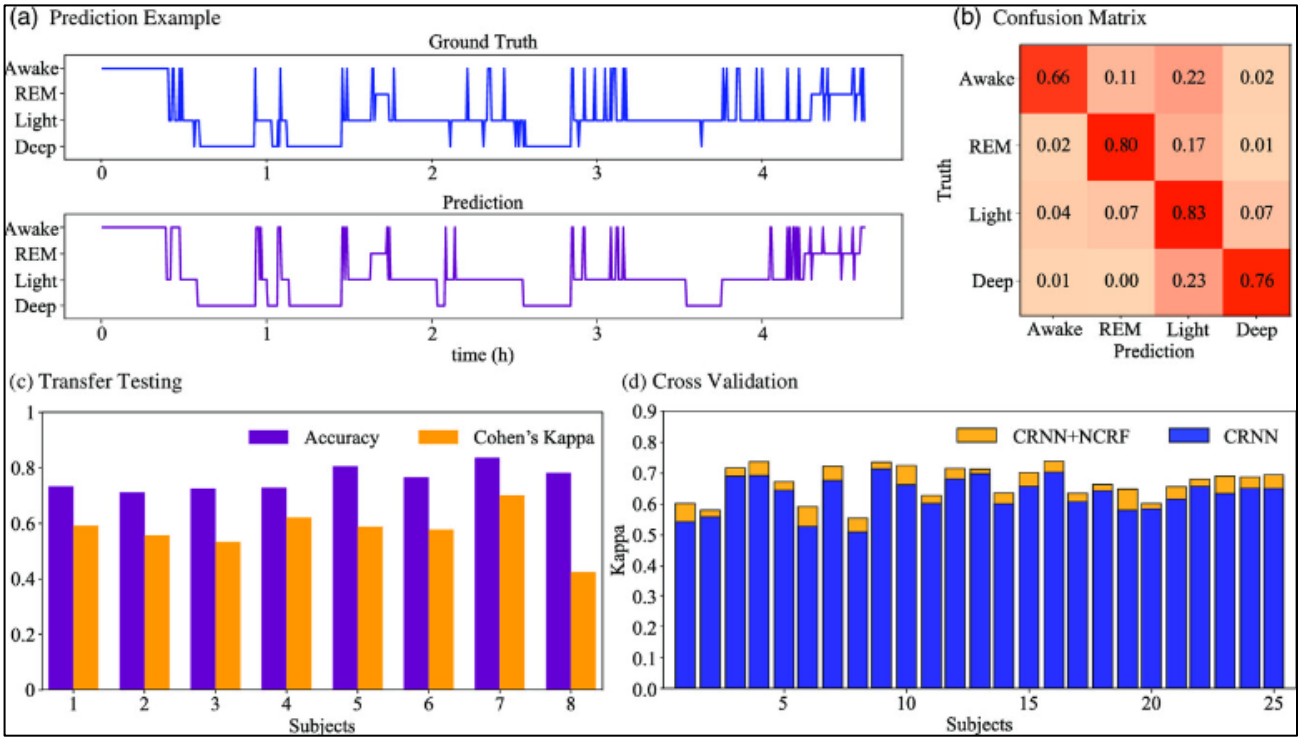


Рисунок 1.3 – Результати прогнозування стадій сну

На рисунку 1.3 використані такі позначення:

а) Приклад прогнозування стадій сну;

б) Загальна матриця плутанини результатів;

в) Результат прогнозування нового частотно-смугового пристрою та нових суб'єктів

г) Каппа Коена перехресної перевірки на різних суб'єктах у наборі даних Массачусетського технологічного інституту.

Для перевірки репрезентативності NN-моделі ми провели тестування на нових пристроях. Однак навчена модель без точного налаштування не змогла працювати на новому пристрої. В той же час, навчання лише на меншій кількості даних та випадковій ініціалізації моделі мало низьку швидкість та точність [9].

Тому ми попередньо навчили кодер CRNN з набором даних 6 ГГц і застосували метод навчання з перенесенням для точного налаштування наскрізної NN-моделі з радіосигналом 60 ГГц.

Ефективність на кожному об'єкті оцінюється за допомогою моделі, налаштованої за даними інших об'єктів. На рисунку 3в показано значення точності та κ для кожного об'єкта та нового радіолокаційного датчика. Середні значення точності та κ становлять 74,3% та 0,594 відповідно.

Результат навчання з перенавчанням демонструє, що модель може бути використана в різних пристроях за допомогою декількох кроків навчання.

Надійність NSSP також оцінюється за допомогою перехресної перевірки на наборі даних MIT. Рисунок 3 ілюструє значення κ на різних об'єктах, отримані за допомогою моделей CRNN та CRF. Стандартне відхилення становить 3,1%. Максимальний і мінімальний показники точності становлять 83,2% і 72,8% відповідно [9].

Значення κ має стандартне відхилення 0,053. Максимальне та мінімальне значення κ становить 0,736 та 0,552 відповідно. Результати перехресної валідації та трансферного навчання показали, що запропонований метод має високий потенціал для реальних застосувань.

Щоб пояснити механізм вилучення кодера, на рисунку 1.4, показано візуалізацію вхідної салівантності та виходу кодера, відповідно. Ці графіки допомагають зрозуміти, як працює NN модель. Як показано на рисунку 1.4, два компоненти (дихання і серцебиття) в радіочастотних сигналах нанесені на тлі відповідної карти значущості.

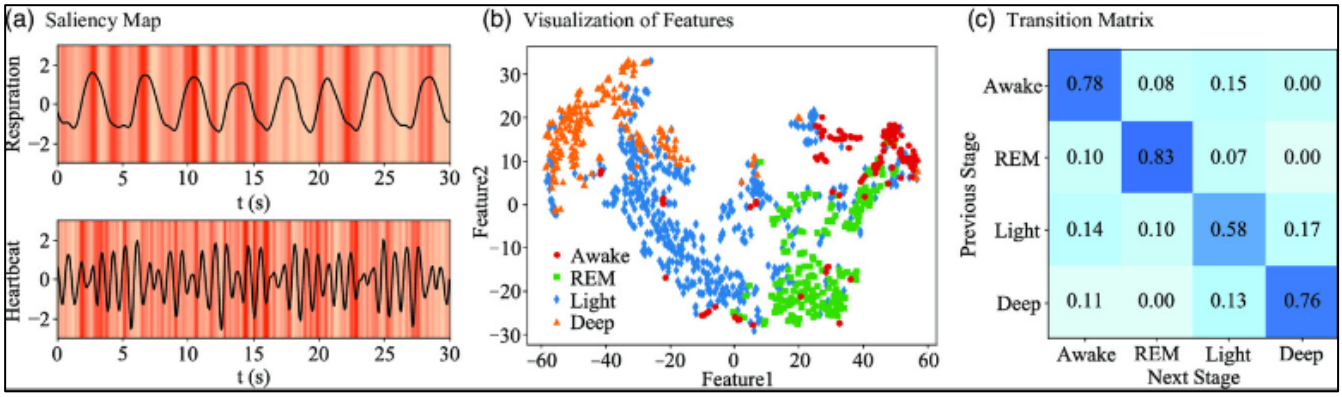


Рисунок 1.4 – Результати роботи CNN

На рисунку 1.4 використані такі позначення:

a) Карта значущості, де темніші лінії позначають вищу концентрацію мережі;

b) Візуалізація ознак, вилучених за допомогою CRNN;

c) Матриця ймовірностей переходу від попередньої стадії сну до наступної, де нульові значення означають заборону переходу від однієї стадії до іншої.

Темніші лінії на карті значущості означають більшу увагу кодера. Видно, що модель концентрується на піках і долинах сигналів дихання і серцебиття. За допомогою фільтрів малого та великого розміру в першому шарі згортки кодера ефективно вловлюються цикли життєво важливих ознак та варіації фізіології.

Ми використовуємо вбудовування t-SNE для візуалізації виходів CNN і RNN окремо. CNN витягує часові характеристики в кожну 30-секундну епоху і має первинну дискримінацію неспання і швидкої фази від нешвидкої фази. Однак, життєві показники під час легкого та глибокого сну схожі між собою, і CNN не може прийняти рішення між цими двома стадіями.

Вихідні дані RNN мають більш чіткі межі класів, ніж ознаки CNN. Модель RNN вивчає динаміку з послідовностей сну і дозволяє надалі визначати глибину сну. Крім того, конкатенація вихідних даних CNN покращує латентне навчання стадії неспання. Аналогічна модель без конкатенації цих двох типів ознак, і загальний показник точності знизився на 2,9%.

Це означає, що конкатенація покращує прогрес навчання і підвищує здатність моделі до класифікації. Що стосується предиктора шару CRF, то матриця

переходів між стадіями, отримана на основі навчальних даних, показує ймовірність сусідньої пари стадій, як показано на рисунку 1.4.

Нульові значення в матриці забороняють відповідні переходи стадій (наприклад, від глибокого сну до швидкої фази) у результатах прогнозування. Модель CRF комплексно використовує особливості каналів, ймовірність спільного переходу та приймає глобально оптимальне рішення. Порівняно з моделлю CRNN, модель переходу від сну до стадії швидкого сну покращує точність на 2,6% і значення κ на 0,036. Результати візуалізації NN-моделі підтверджують, що метод є пояснюваним і достовірним.

Аналіз структури сну. Параметри структури сну мають відношення до якості сну. Точна оцінка цих параметрів корисна як для пацієнта, так і для лікаря для моніторингу прогресу клінічної терапії. Наступні шість параметрів сну були обрані для перевірки придатності NSSP для аналізу структури сну.

Загальна тривалість сну визначається як сумарний час стадій неспання. Ефективність сну визначається як частка стадії неспання протягом всієї ночі сну для оцінки якості сну [8].

Латентність сну та латентність швидкого сну визначаються як час, що минув від початку засинання до перших 30 с, які оцінюються як сон та швидкий сон, відповідно. Пропорції глибокого та швидкого сну визначаються як загальний відсоток стадії глибокого сну та стадії швидкого сну протягом всієї ночі, відповідно.

Вищезгадані параметри розраховуються на основі всіх даних нічного сну, отриманих за допомогою радіолокаційних датчиків та методу PSG. На рисунку 1.5 показано розподіл параметрів серед обстежуваних та порівняння оцінки істинності між радіолокаційними та PSG-методами.

Пунктирні лінії на кожному рисунку позначають верхній і нижній квартилі підписаних помилок оцінювання. Абсолютна похибка оцінки та середня точність за різними параметрами проілюстровані в Таблицях S1 та S2 у Допоміжній інформації. Зокрема, оцінки загальної тривалості сну та ефективності сну досягають найвищої точності [9].

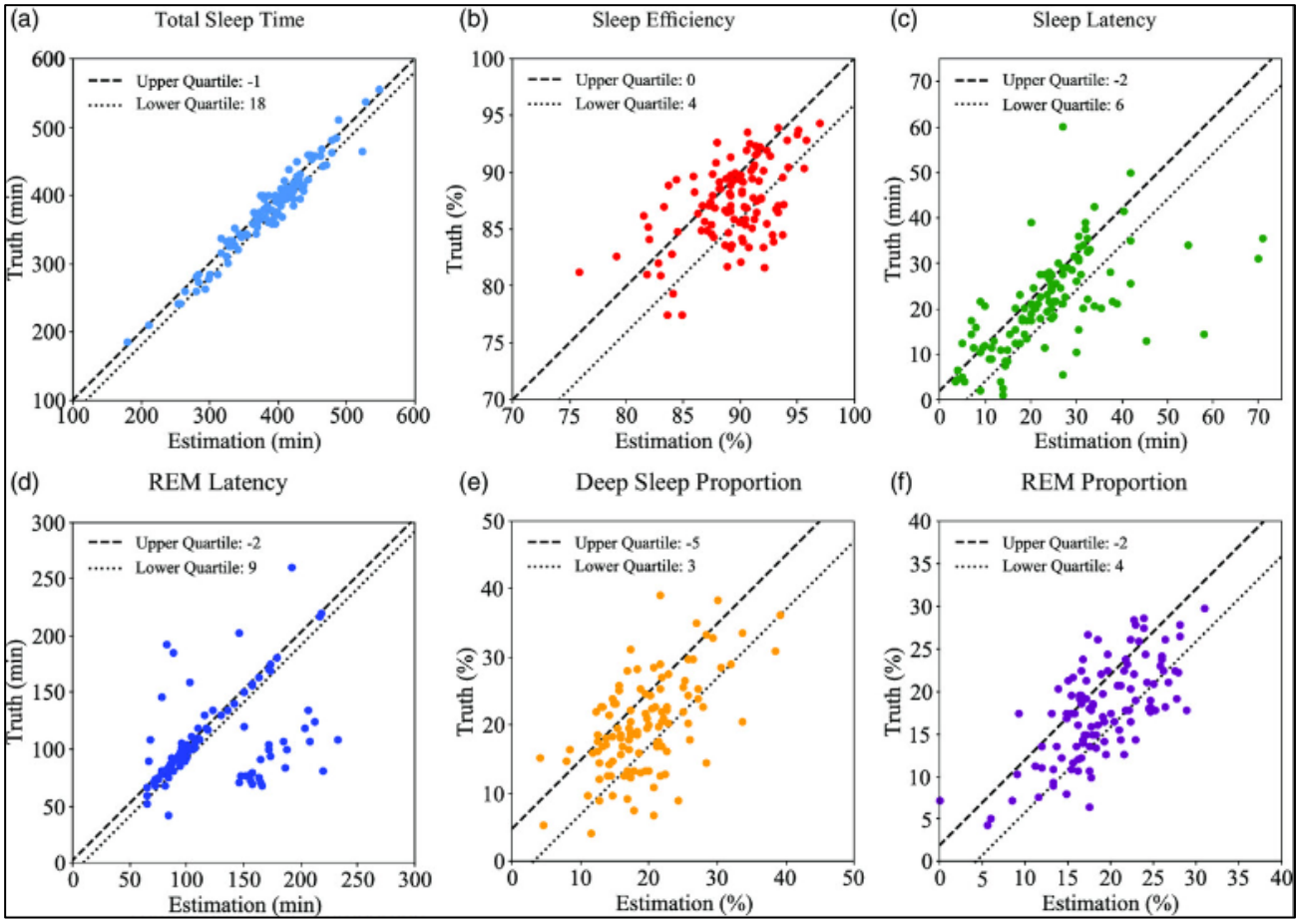


Рисунок 1.5 – Результати оцінки параметрів сну

На рисунку 1.5 використані такі позначення:

a) Порівняння оцінки з істиною для загальної тривалості сну.

b) Порівняння оцінки з істиною для ефективності сну.

c) Порівняння оцінки з істиною для латентності сну.

d) Порівняння оцінки з істиною для латентності швидкого сну.

e) Порівняння оцінки з істиною для частки глибокого сну.

f) Порівняння оцінки з істиною для частки фази швидкого сну.

Оцінка латентності сну, очевидно, є менш точною, ніж інші параметри, вона є результатом короткого проміжку часу, що минув перед тим, як суб'єкт заснув. Відповідна абсолютна похибка в середньому помітно мала і підтверджує можливість ідентифікації переходу від неспання до сну. Точність інших оцінених параметрів перевищує 70%.

Результати демонструють здатність NSSP оцінювати параметри сну для аналізу сну і його великий потенціал у клінічній діагностиці сну.

У висновку цих досліджень можна сказати що NSSP для прогнозування стадій сну та аналізу структури сну були продемонстровані на радіолокаційних датчиках з різною робочою частотою та на різних суб'єктах [9].

Завдяки технології моделі NN, високій кореляції життєво важливих показників сну з радіочастотними сигналами та значній кількості даних, тому NSSPS перевершила інші сучасні методи прогнозування стадій сну. Попередній результат також продемонстрував надійність методу на різних суб'єктах і різних сенсорних пристроях.

1.4 Висновки

У результаті проведеного аналізу предметної області було встановлено значущі аспекти, пов'язані з розладом сну. Це дозволило розглянути основні фактори, що впливають на якість сну та його розлади. Виявлено, що інформаційні технології можуть відігравати ключову роль у виявленні та управлінні цими розладами.

Оцінивши переваги та обмеження доступних технологій, з'ясувалось, що кожна з них має свої сильні та слабкі сторони. Інтеграція різних технологій може сприяти більш ефективному виявленню розладу сну та підвищенню точності передбачень.

Оглянувши існуючі технології для виявлення розладу сну, можемо зробити висновки, що деякі з них демонструють високу ефективність у реальних умовах, забезпечуючи точні та достовірні дані про якість сну. Однак, є простір для подальшого розвитку і вдосконалення існуючих технологій з метою забезпечення більш точних та доступних засобів аналізу та передбачення розладів сну.

Крім того, було проведено детальний аналіз існуючих аналогів, зосереджуючись на їх перевагах та недоліках, з метою виокремлення прогалин, які необхідно заповнити у цьому дослідженні. Це дозволяє нам уникнути повторення

попередніх помилок та зробити значний внесок у розвиток області інформаційної технології, сприяючи подальшому вдосконаленню інструментів та методів аналізу та передачі сигналу.

Автоматизований, ефективний та результативний моніторинг стадій сну та аналіз його структури є важливою процедурою для автоматизації охорони здоров'я. Для розумних пристроїв залишаються проблеми з точним визначенням стадії сну і мінімізацією втручання в перебіг сну. Також було розглянуто технологію, безконтактну систему прогнозування структури сну (NSSP), яка використовує один радарний датчик для аналізу структури сну без прив'язки до будь-якого пристрою.

Проведений огляд аналогів для передбачення та аналізу розладу сну показав різноманітність наявних підходів та методик. Були виявлені різні системи та технології, що використовуються для виявлення та аналізу параметрів сну. Деякі з цих аналогів демонструють високу точність та надійність у виявленні розладів сну, однак, існують певні обмеження, наприклад як складність використання. Для покращення цих технологій необхідно подальше дослідження та розвиток, спрямований на зменшення витрат та підвищення їх доступності для широкого кола користувачів.

Також, існує потенціал у поєднанні різних аспектів цих аналогів для створення більш комплексних та ефективних систем для аналізу та передбачення розладів сну.

2 ВИБІР ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА МОДЕЛЕЙ

2.1 Аналіз обраної платформи розробки

Kaggle – це онлайн-платформа для аналізу даних, машинного навчання та штучного інтелекту, яка об'єднує співтовариство даних та спеціалістів з усього світу. Заснована в 2010 році з метою створення місця, де люди з різних сфер можуть займатися проектами з аналізу даних, співпрацювати, змагатися та вчитися.

Платформа пропонує широкий спектр можливостей для своїх користувачів. Однією з головних функцій є доступ до різноманітних наборів даних. Kaggle містить велику базу даних з різних галузей, від фінансів та медицини до транспорту та спорту. Ці набори даних можна використовувати для аналізу, обробки та створення моделей машинного навчання [10].

Крім цього, Kaggle відомий своїми змаганнями з аналізу даних. Користувачі можуть брати участь у конкурсах, де їм потрібно розв'язати певні завдання за допомогою аналізу даних та створення моделей прогнозування. Ці змагання можуть мати реальний вплив на проблеми в галузі, такі як медицина, енергетика, екологія та багато інших

Також на Kaggle є "Кернели" (Kernels) - це середовище для написання та виконання коду. Це дозволяє користувачам створювати та ділитися своїми аналітичними ноутбуками, де вони можуть показувати свій код, робити висновки та виконувати різноманітні аналізи на доступних даних.

Крім цього, платформа надає доступ до навчальних ресурсів, включаючи курси, відеоуроки та матеріали для вивчення аналізу даних та машинного навчання.

Платформа створює активну спільноту, де користувачі можуть обмінюватися знаннями, досвідом та ідеями через форуми, коментарі, дискусії та співпрацювати над проектами.

В цілому, Kaggle є не лише платформою для доступу до даних та розв'язання завдань з машинного навчання, але й громадським простором, де люди з усього

світу можуть об'єднатися для розв'язання складних завдань, використовуючи дані та інноваційні підходи.

Kaggle є важливим інструментом для розробників та дослідників у сфері аналізу даних та машинного навчання. Ця платформа сприяє спільноті фахівців, студентів та всіх зацікавлених осіб у співпраці, вивченні, обміні знаннями та розв'язанні реальних проблем.

Набори даних на Kaggle створюють можливість для доступу до широкого спектру інформації, яка може бути використана для створення моделей, прогнозів та розв'язання реальних завдань. Вони представляють собою цінний ресурс для тих, хто шукає дані для власних досліджень, тестування алгоритмів або розробки нових моделей машинного навчання [10].

Крім цього, змагання на Kaggle створюють стимул для розробників у напрямку пошуку найкращих можливих рішень для конкретних завдань. Ці конкурси допомагають залучати увагу до важливих проблем та сприяють розвитку нових ідей та методів.

"Кернели" (Kernels) на Kaggle це віртуальні середовища, де користувачі можуть розробляти, виконувати та ділитися своїми ноутбуками з аналізом даних та кодом. Це дає можливість показувати та демонструвати роботу своїх моделей та аналітичних рішень.

Каггл також надає можливість навчання, пропонуючи курси та навчальні матеріали для тих, хто бажає покращити свої навички в області аналізу даних та машинного навчання.

Платформа об'єднує співтовариство аналітиків даних та спеціалістів з усього світу. Вона надає доступ до різноманітних даних, дозволяючи користувачам використовувати їх для розробки та тестування алгоритмів машинного навчання. Платформа організовує конкурси, що стимулюють розробників змагатися та знаходити кращі рішення для певних завдань аналізу даних.

Набори даних на Kaggle дозволяють звертатися до різних сфер і отримувати реальні дані для роботи та вдосконалення аналітичних навичок. Крім того, платформа забезпечує доступ до інструментів, бібліотек та навчальних ресурсів для

підвищення кваліфікації у сфері машинного навчання та аналізу даних. Вона також стимулює співпрацю, обмін знаннями та досвідом між учасниками, створюючи сприятливе середовище для розвитку та спільноти.

Головна сторінка Kaggle є важливим центральним місцем, куди користувачі звертаються для оновлення щодо актуальних подій і можливостей, які надає ця платформа. Вона забезпечує швидкий доступ до ключових розділів, таких як конкурси, набори даних, навчальні ресурси та проекти.

На головній сторінці можуть бути видимі найбільш актуальні та популярні конкурси, які заохочують учасників розвиватися та змагатися за рішення конкретних завдань з аналізу даних та машинного навчання.

Також, можна зустріти відібрані набори даних, які можуть бути корисними для розробки та тестування моделей. Представлення популярних або цікавих проектів та ноутбуків може стимулювати інтерес користувачів до певних аспектів аналізу даних та може надихнути на нові ідеї для власних проектів.

Загалом, головна сторінка Kaggle є динамічним центром, який надає широкий спектр інформації та можливостей для спілкування, навчання та розвитку в області аналізу даних та машинного навчання. Це місце, де користувачі можуть знайти все необхідне для своєї активної участі у спільноті Kaggle та реалізації своїх аналітичних проектів (рис. 2.1).

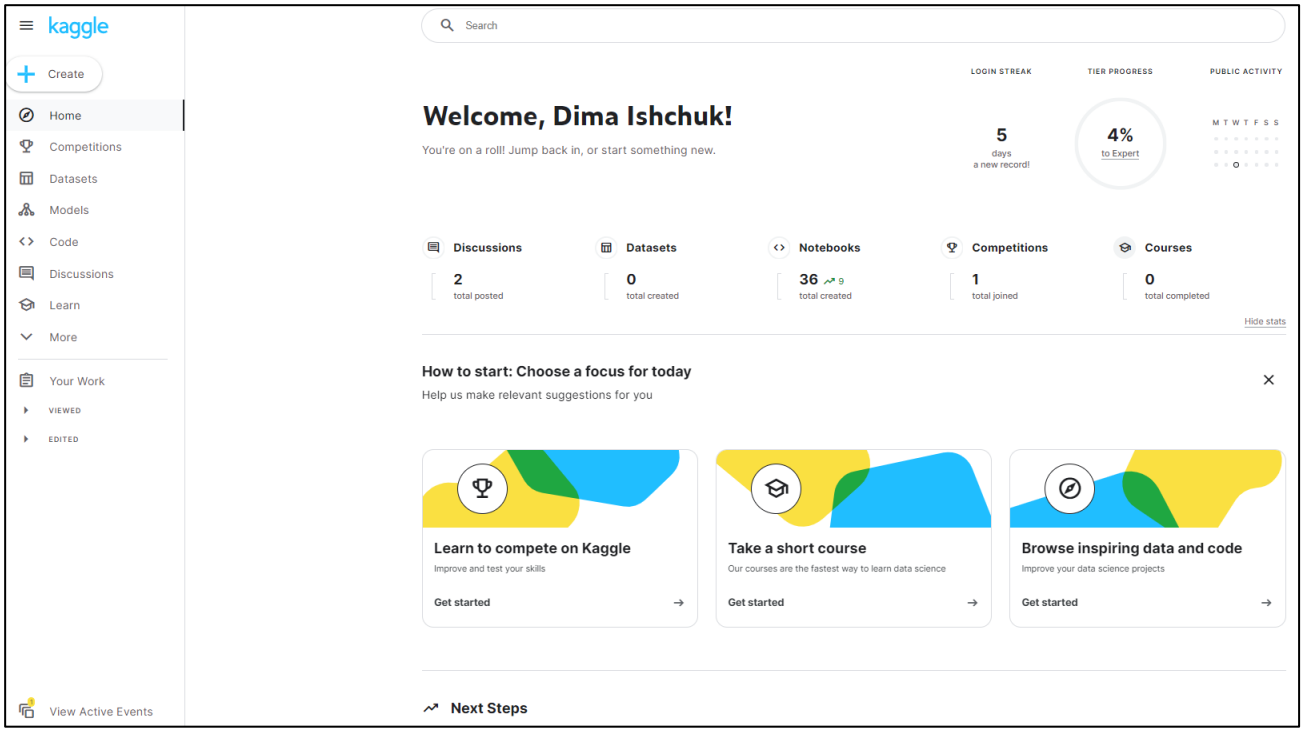


Рисунок 2.1 – Головне вікно платформи Kaggle

Загалом, Kaggle не лише сприяє доступу до різноманітних даних, але й створює середовище для співпраці, вивчення, вдосконалення навичок та спільної розробки ідей, що робить його потужним інструментом для розробників та дослідників у сфері аналізу даних та машинного навчання [11].

Python – це високорівнева, інтерпретована, універсальна мова програмування, яка здобула широку популярність у сфері аналізу даних, машинного навчання та штучного інтелекту. У спільноті Kaggle Python вважається однією з найбільш популярних мов програмування з багатьма перевагами. Синтаксис Python є простим та легко читається, що робить його дружнім для початківців та зручним для швидкого розроблення коду. Це особливо корисно у сфері аналізу даних, де зрозумілість коду грає важливу роль.

Широкі можливості та багато модулів: Python має велику кількість бібліотек, таких як NumPy, Pandas, Matplotlib, Scikit-learn та інші, які сприяють аналізу даних, обробці даних та машинному навчанню. Ці бібліотеки роблять Python потужним інструментом для роботи з даними на платформі Kaggle.

Python має велику активну спільноту, що означає, що користувачі можуть легко знайти рішення для багатьох проблем або підтримку, якщо у них виникають питання. Це робить його привабливим для тих, хто шукає допомогу чи поради.

Але при цьому Python в порівнянні з деякими мовами програмування, такими як C++ або Java, може бути менш ефективним за швидкодією обробки даних, оскільки він інтерпретований, а не компілюється [11].

У деяких випадках, коли маємо справу з великими обсягами даних, може виникнути проблема з обробкою швидкодії через обмежену продуктивність інтерпретатора Python.

Загалом, Python в Kaggle залишається однією з найпопулярніших мов програмування для аналізу даних, оскільки його переваги, такі як простота, багаті можливості та велика спільнота, переважають його недоліки.

Її привабливість у сфері Kaggle полягає у простоті синтаксису, що полегшує розуміння коду та зменшує час на розробку. Python відомий своєю читабельністю, що робить його особливо зручним для роботи з великими проектами, де розуміння коду є ключовим фактором.

Крім того, Python має велику кількість розширень та бібліотек, які постійно оновлюються та розвиваються. Це означає, що користувачі можуть легко знаходити нові інструменти, що полегшують їхню роботу та розвиток проектів. Багато з цих бібліотек знаходяться під активним розвитком, що дозволяє користувачам бути в тренді та використовувати оновлені методики та підходи.

Однією з ключових переваг Python також є його безкоштовність та відкритий код, що сприяє швидкому розповсюдженню та розвитку спільноти розробників. Це стимулює обмін знаннями, співпрацю та вдосконалення мови шляхом спільної роботи багатьох вкладників.

Не дивлячись на свої переваги, Python також може мати деякі обмеження, особливо при роботі з великими обсягами даних чи задачами, які потребують високої швидкодії. В таких ситуаціях, інші мови програмування, які працюють ближче до машинного коду, можуть бути ефективнішими.

Python в Kaggle також відрізняється своєю універсальністю. Він не обмежується лише аналізом даних чи машинним навчанням, але також використовується у веб-розробці, автоматизації завдань, наукових дослідженнях, розробці ігор та багатьох інших галузях. Ця універсальність дає можливість користувачам використовувати свої навички Python в різних контекстах та проектах [11].

Крім того, Python має велику кількість розширень та бібліотек, які постійно оновлюються та розвиваються. Це означає, що користувачі можуть легко знаходити нові інструменти, що полегшують їхню роботу та розвиток проектів. Багато з цих бібліотек знаходяться під активним розвитком, що дозволяє користувачам бути в тренді та використовувати оновлені методики та підходи.

Однією з ключових переваг Python також є його безкоштовність та відкритий код, що сприяє швидкому розповсюдженню та розвитку спільноти розробників. Це стимулює обмін знаннями, співпрацю та вдосконалення мови шляхом спільної роботи багатьох вкладників.

Не дивлячись на свої переваги, Python також може мати деякі обмеження, особливо при роботі з великими обсягами даних чи задачами, які потребують високої швидкодії. В таких ситуаціях, інші мови програмування, які працюють ближче до машинного коду, можуть бути ефективнішими.

Загалом, Python в Kaggle вважається сильним інструментом завдяки своїм широким можливостям, простоті та універсальності, що забезпечують користувачам потужний інструмент для роботи над різноманітними проектами та завданнями у сфері аналізу даних та машинного навчання. Тому вибір Kaggle для роботи над темою аналізу та передбачення розладів сну може бути обґрунтованим через доступність реальних даних, можливості співпраці та обміну знаннями з іншими фахівцями, а також застосування потужних інструментів для розвитку моделей та аналізу даних. В цілому, Python є потужним інструментом для аналізу даних та машинного навчання в Kaggle, завдяки своїй простоті, багатим можливостям, великій спільноті та наявності широких ресурсів для вивчення та розвитку.

2.2 Вибір моделі машинного навчання «Naive Bayes» та «Decision Tree»

"Наївний Баєс" (Naive Bayes) - це простий, але ефективний метод машинного навчання, особливо в області класифікації. Ця модель ґрунтується на теоремі Баєса, яка використовує ймовірності для прогнозування класів на основі вхідних ознак. Основне припущення полягає в тому, що ознаки є незалежними одна від одної, хоча у реальних даних це може бути не завжди правдою [12].

Цей алгоритм особливо корисний, коли маємо велику кількість ознак та потрібно швидко і точно класифікувати дані. Він широко застосовується в таких областях, як фільтрація спаму в електронній пошті, класифікація текстів, аналіз відгуків користувачів та багато інших.

Підхід "Наївного Баєса" передбачає, що ми маємо набір даних, в якому кожен запис представляє об'єкт з рядком ознак та відповідним класом або міткою. Під час навчання моделі обчислюються ймовірності належності кожного класу для вхідних даних за допомогою теореми Баєса. Для цього обчислюються апріорні ймовірності класів та умовні ймовірності ознак для кожного класу [12].

Одним з головних переваг "Наївного Баєса" є його ефективність при роботі з великими обсягами даних та швидкість навчання. Також він вважається досить стійким до невеликої кількості даних для навчання. Крім того, цей метод простий у реалізації та вимагає менше обчислювальних ресурсів порівняно з іншими складнішими алгоритмами машинного навчання.

Проте "Наївний Баєс" має свої обмеження. Він може давати грубі або неочікувані результати, якщо припущення про незалежність ознак не виконується. Також, якщо вхідні дані мають велику кореляцію між ознаками, це може призвести до менш точних прогнозів. Для покращення точності моделі можуть застосовуватися методи попередньої обробки даних або модифікації самого алгоритму [13].

Однак важливо враховувати певні обмеження цього методу. Наприклад, якщо дані мають високу кореляцію між ознаками або якщо припущення про незалежність ознак не виконується, точність класифікації може знизитися.

Також слід зазначити, що через використання ймовірнісного підходу "Наївний Баєс" може виявляти проблеми з несбалансованими класами даних або з недостатньою кількістю інформації для деяких категорій. У таких ситуаціях може знадобитися уважне налаштування моделі або використання інших методів роботи з дисбалансом класів.

Однак не зважаючи на ці обмеження, "Наївний Баєс" залишається важливим інструментом в аналізі даних та машинному навчанні, особливо для завдань класифікації тексту, фільтрації спаму, аналізу настроїв і багатьох інших сфер. Його простота та швидкість роботи роблять його популярним в багатьох сферах наукових та технологічних досліджень.

Даний алгоритм використовує теорему Баєса для класифікації даних. Основна ідея полягає в застосуванні ймовірностного підходу до прийняття рішень.

Припустимо, ми маємо набір даних, що складається з об'єктів із вхідними ознаками (наприклад, словами в тексті) та відповідними класами (наприклад, категоріями текстів). Алгоритм "Наївний Баєс" припускає, що кожна ознака незалежна від інших, що є "наївним" припущенням, і використовує це для спрощення обчислень [13].

Під час навчання моделі "Наївний Баєс" обчислює ймовірності належності кожного класу для вхідних даних за допомогою теореми Баєса. Це включає розрахунок апріорних ймовірностей для кожного класу та умовних ймовірностей ознак для кожного класу. Умовні ймовірності ознак визначають, яка ймовірність того, що певна ознака з'явиться у певному класі.

Коли модель отримує нові дані для класифікації, вона використовує ці ймовірності для визначення ймовірності належності об'єкта до кожного класу. Для цього вона обчислює добуток умовних ймовірностей для кожної ознаки об'єкта для кожного класу та множить на апріорну ймовірність класу. Клас, для якого отримана максимальна значення ймовірності, стає прогнозованим класом для даного об'єкта.

Одним із прикладів використання "Наївного Баєса" є класифікація електронних листів на спам і не спам. Алгоритм аналізує слова, які зазвичай зустрічаються в спам-листах і тих, що зазвичай використовуються в звичайних

листах. Він обчислює ймовірність того, що певне слово з'явиться в спамі або не спамі, і за цими даними класифікує нові листи.

Незважаючи на свою "наївність", цей метод може бути ефективним у багатьох ситуаціях, особливо у випадках, коли дані розглянуті як незалежні. Однак варто бути уважним при застосуванні до реальних даних, оскільки припущення про незалежність ознак може бути не завжди реалістичним [13].

Наступна модель Decision Tree — непараметричний контрольований алгоритм навчання. Він має ієрархічну структуру дерева, яка складається з кореневого вузла, гілок, внутрішніх вузлів і листових вузлів, простий приклад на рисунку 2.2.

Дерева рішень є основою для багатьох класичних алгоритмів машинного навчання, таких як випадкові ліси, пакетування та прискорені дерева рішень. Його ідея полягала в тому, щоб представити дані у вигляді дерева, де кожен внутрішній вузол позначає перевірку атрибута (загалом це умова), кожна гілка представляє результат тесту, а кожен кінцевий вузол (кінцевий вузол) містить мітку класу [14].

Дерево рішень можна використовувати як для класифікації, так і для регресії. Методики дещо інші, але принципи однакові. Дерева рішень використовують алгоритм CART (дерева класифікації та регресії). В обох випадках рішення базуються на умовах щодо будь-якої функції. Внутрішні вузли представляють умови, а кінцеві вузли представляють рішення на основі умов.

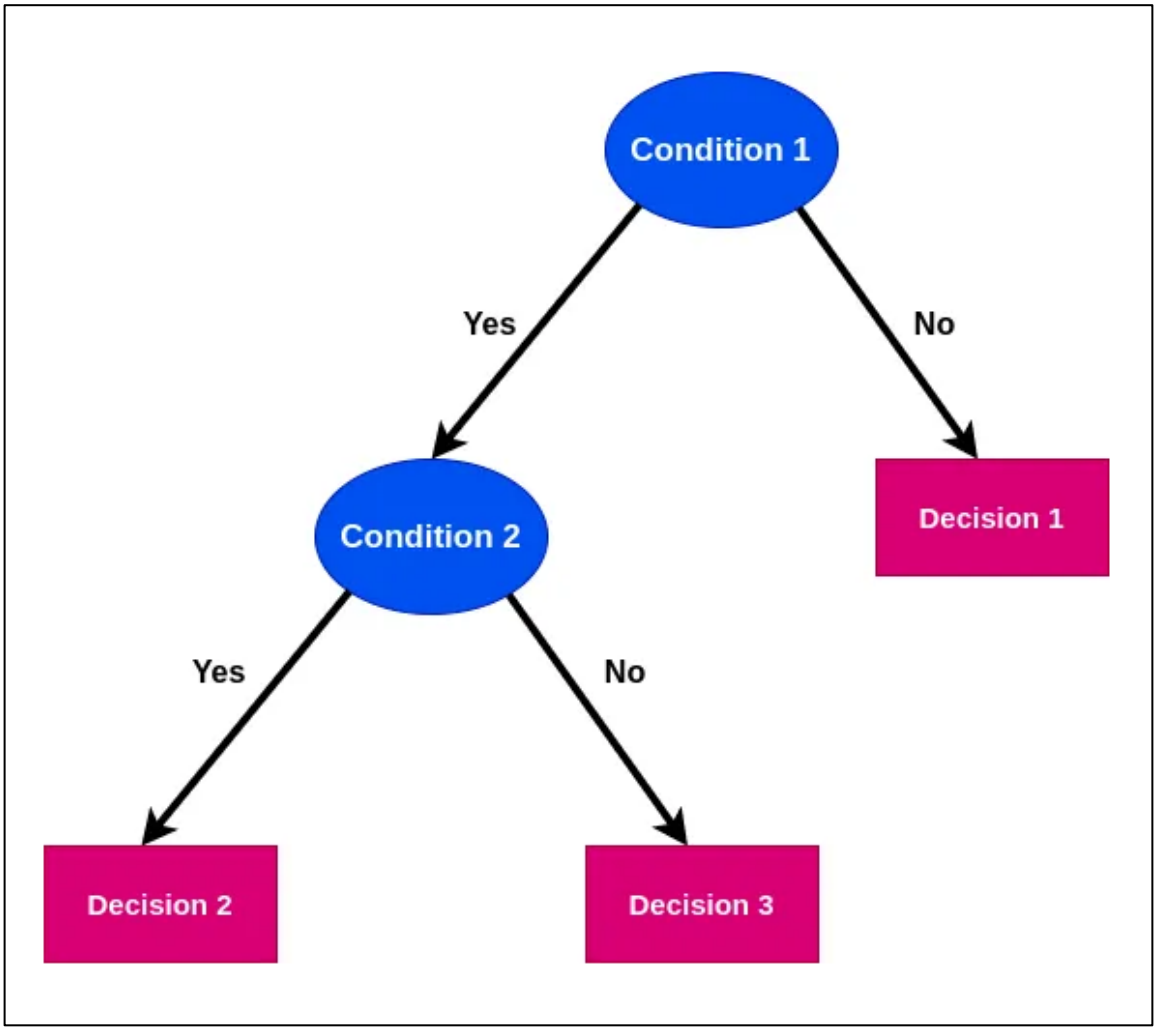


Рисунок 2.2 – Простий приклад робити алгоритму «Decision Tree»

Наприклад дерево рішень будує моделі класифікації або регресії у формі деревовидної структури. Він розбиває набір даних на дедалі менші підмножини, водночас поступово розробляючи відповідне дерево рішень. Кінцевим результатом є дерево з вузлами рішень і листовими вузлами. Вузол прийняття рішень (наприклад, Outlook) має дві або більше гілок (наприклад, Sunny, Clouds і Rainy). Листовий вузол (наприклад, Play) представляє класифікацію або рішення. Найвищий вузол прийняття рішень у дереві, який відповідає найкращому предиктору, називається кореневим вузлом**.** Дерева рішень можуть обробляти як категоричні, так і числові дані, приклад на рисунку 2.3.

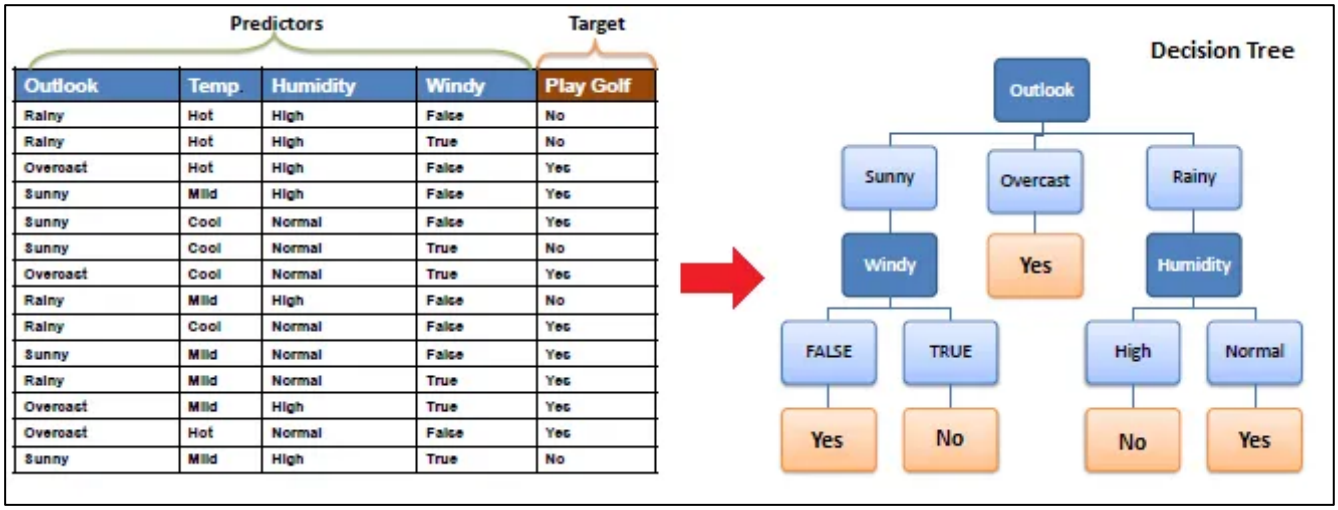


Рисунок 2.3 – Приклад розбиття даних у алгоритмі «Decision Tree»

Дерева рішень перевернуті, що означає, що корінь знаходиться вгорі, а потім цей корінь розбивається на декілька вузлів. Дерева рішень — це не що інше, як купа тверджень «якщо-інакше», кажучи непрофесіоналом. Він перевіряє, чи справжня умова, і якщо так, то переходить до наступного вузла, приєднаного до цього рішення [14].

Індекс Джіні — це міра домішок або чистоти, яка використовується під час створення дерева рішень в алгоритмі CART (Дерево класифікації та регресії). Атрибуту з низьким індексом Джині слід віддавати перевагу порівняно з високим індексом Джині. Він створює лише двійкові розділення, а алгоритм CART використовує індекс Джіні для створення двійкових розділень.

Індекс Джині можна розрахувати за такою формулою:

$$Gini = 1 - \sum_{i=1}^{n}(p_i), \tag{2.1}$$

де *pi* — ймовірність належності певного елемента до певного класу.

Індекс Джині є мірою невпорядкованості в контексті алгоритму дерев рішень використовується для визначення того, як добре певний вузол розділяє дані за певним критерієм [14].

Цей індекс оцінює ймовірність того, що випадково обраний елемент з набору даних буде неправильно класифікований, враховуючи розподіл класів в цьому

вузлі. Чим менше ця ймовірність, тим менша невпорядкованість, і тим краще вузол розділяє дані.

При побудові дерева рішень алгоритм намагається зменшити індекс Джині в кожному вузлі, розділяючи дані на підгрупи таким чином, щоб кожна нова група була більш однорідною. Це допомагає зробити прогнози точнішими, оскільки розділення на більш однорідні групи дозволяє краще узагальнювати результати на нових даних [14].

Тому індекс Джині використовується як критерій для визначення оптимальної точки розділення в дереві рішень, де кожен вузол розбиває дані на більш однорідні групи, що сприяє покращенню точності прогнозів моделі.

У підсумках можна сказати що це алгоритм керованого навчання, який можна використовувати як для класифікації, так і для регресії. Основна мета дерева рішень це розділити набір даних у вигляді дерева на основі набору правил і умов.

Ентропія в контексті алгоритму дерева рішень є мірою невпорядкованості чи неоднорідності у наборі даних. Вона вимірює ступінь безладу в системі, враховуючи розподіл класів в даних.

При побудові дерева рішень, алгоритм спробує розділити дані на частини таким чином, щоб мінімізувати ентропію в кожному вузлі. Тобто, розділення даних на більш однорідні групи сприяє зменшенню ентропії. Коли дані розділяються на менш різноманітні групи (з меншою ентропією), модель стає здатнішою до точного прогнозування на нових даних.

Використання ентропії в алгоритмі дерева рішень допомагає визначати оптимальні точки розділення, де кожен вузол дерева розділяє дані на більш однорідні групи, що дозволяє покращити точність моделі при прийнятті рішень.

$$-\sum_{i=1}^{c} p_i \, log_2(p_i), \tag{2.2}$$

де c - кількість класів у вузлі;

pi - ймовірність випадку, коли об'єкт належить до класу i.

Ця формула вираховує ентропію для вузла. Чим менше ентропія, тим менше невпорядкованість у вузлі, що означає більшу однорідність даних.

Ентропія є поняттям з теорії інформації та статистики, що використовується у машинному навчанні для вимірювання невпорядкованості чи неоднорідності в наборі даних.

Це важлива метрика, яка дозволяє оцінювати ступінь неочікуваності або несподіваності у системі. У машинному навчанні, особливо в алгоритмах дерев рішень, ентропія використовується як критерій для визначення оптимального розділення вузлів для покращення точності моделі.

В алгоритмі дерева рішень ентропія використовується для оцінки невпорядкованості вузлів. Це дозволяє визначити, як добре певний вузол розділяє дані за певним критерієм. Чим менше ентропія, тим менша невпорядкованість, і тим більш однорідним є вузол. Критерій розділення, який зменшує ентропію, вважається більш оптимальним, оскільки він робить модель більш точною та менш схильною до перенавчання [14].

Тому для можна використовувати у задачах класифікації, коли алгоритм дерева рішень розглядає розділення даних за певними ознаками, він обчислює ентропію для кожного потенційного розділення та обирає те, яке мінімізує загальну ентропію вузла. Це дозволяє алгоритму розділити дані таким чином, що кожен вузол стає більш однорідним і поліпшує точність моделі.

Таким чином, ентропія використовується в машинному навчанні для покращення точності моделей, забезпечуючи їхню здатність ефективно вирішувати задачі класифікації чи прогнозування на основі ступеня неоднорідності та невпорядкованості в наборі даних.

GridSearch є методом гіперпараметричної оптимізації, який використовується для пошуку найкращих гіперпараметрів моделі шляхом систематичного перебору всіх можливих комбінацій гіперпараметрів і вибору того, який має найкращі показники ефективності.

У контексті дерева рішень GridSearch використовується для підбору оптимальних значень гіперпараметрів моделі дерева рішень. Наприклад,

гіперпараметри можуть включати глибину дерева, критерій розділення, максимальну кількість листків у вузлі, мінімальну кількість зразків у листку тощо.

GridSearch перебирає різні комбінації значень гіперпараметрів, які ми хочемо оптимізувати. Для кожної комбінації гіперпараметрів GridSearch проводить крос-валідацію (зазвичай з використанням стратегії крос-валідації типу k-fold), оцінює модель і зберігає метрику ефективності, таку як точність чи f1-score. Після того, як всі комбінації перевірені, GridSearch повертає комбінацію гіперпараметрів, яка показала найкращі показники ефективності моделі.

Наприклад, для дерева рішень GridSearch може перебирати різні значення глибини дерева, критерії розділення (наприклад, ентропія чи коефіцієнт Джині), максимальну кількість листків у вузлі та інші параметри.

Шляхом систематичного перебору цих значень GridSearch допомагає знайти найоптимальніші гіперпараметри для моделі дерева рішень, що може підвищити її точність та здатність узагальнювати на нові дані.

2.3 Вибір моделі машинного навчання «Random Forest» та «Extra Trees»

"Random Forest" та "Extra Trees" - це обидва алгоритми ансамблю дерев рішень, які використовуються у машинному навчанні для задач класифікації та регресії.

Random Forest це метод є ансамблем, який складається з кількох дерев рішень, де кожне дерево побудоване на випадковій підвибірці даних і випадковому виборі ознак для кожного вузла. При прийнятті рішення, враховується прогноз кожного дерева, і результат визначається більшістю голосів або шляхом усереднення. "Random Forest" відомий своєю високою точністю, відсутністю перенавчання та здатністю обробки великої кількості ознак [15].

Однією великою перевагою випадкового лісу є те, що його можна використовувати як для задач класифікації, так і для регресії, які складають більшість сучасних систем машинного навчання.

Якщо розглянути випадковий ліс у класифікації, іноді вважається будівельним блоком машинного навчання. На рисунку 2.4, можна побачити, як виглядала б випадкова модель лісу з двома деревами.

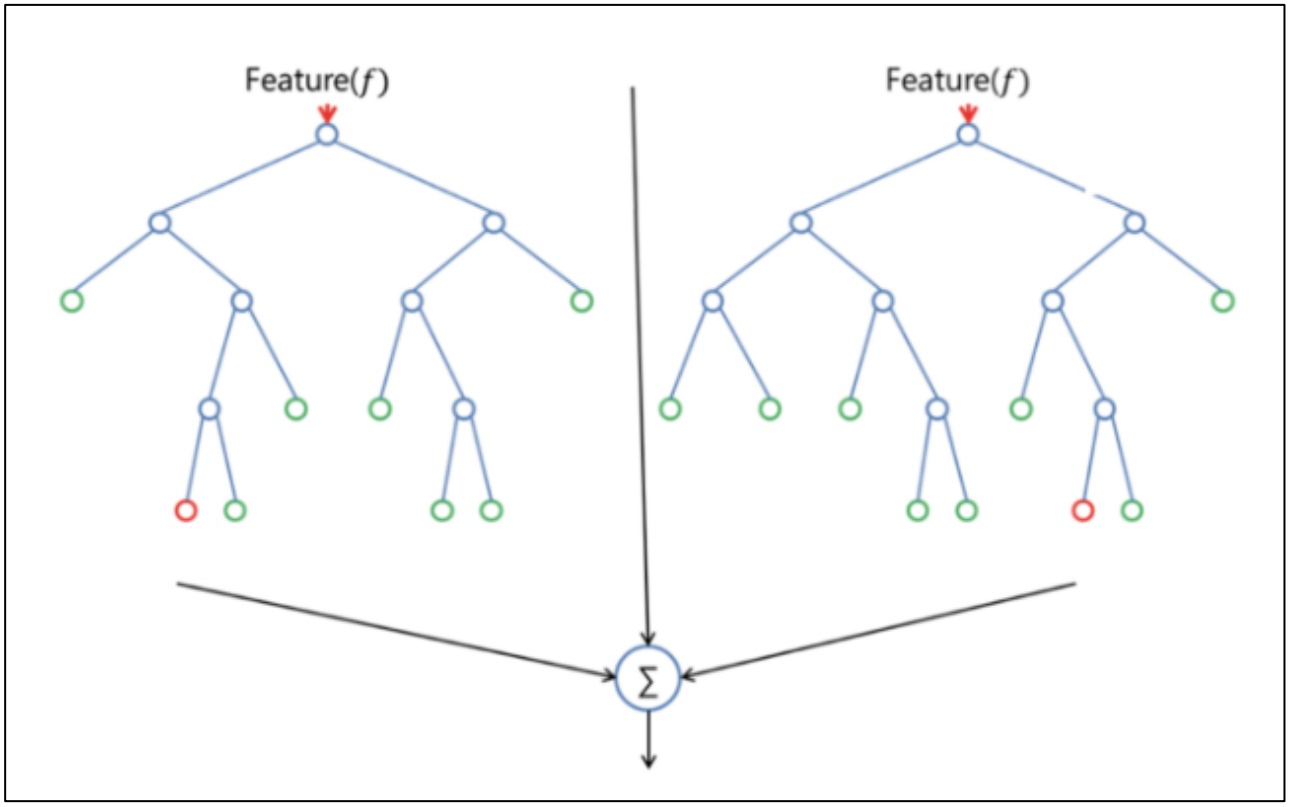


Рисунок 2.4 – Модель Random Forest з двома деревами

Випадковий ліс у класифікації та регресії має майже такі самі гіперпараметри, як дерево рішень або класифікатор пакетування. На щастя, немає потреби поєднувати дерево рішень із класифікатором пакетування, тому що ви можете легко використовувати клас класифікатора випадкового лісу. З випадковим лісом ви також можете мати справу з завданнями регресії за допомогою регресора алгоритму.

Випадковий ліс додає моделі додаткову випадковість, одночасно вирощуючи дерева. Замість пошуку найважливішої функції під час поділу вузла він шукає найкращу функцію серед випадкової підмножини функцій. Це призводить до широкого розмаїття, що загалом призводить до кращої моделі [15].

Таким чином, у класифікаторі випадкового лісу лише випадкова підмножина ознак береться до уваги алгоритмом поділу вузла. Ви навіть можете зробити дерева більш випадковими, додатково використовуючи випадкові пороги для кожної функції, а не шукати найкращі можливі пороги (як це робить звичайне дерево рішень).

Модель випадкового лісу проти дерев прийняття рішень має деякі відмінності .

Наприклад присутній навчальний набір даних із функціями та мітками в дерево рішень, він сформулює певний набір правил, які використовуватимуться для прогнозування.

Щоб передбачити, чи натисне людина онлайн-рекламу, можна зібрати оголошення, на які ця особа натискала в минулому, і деякі характеристики, які описують її рішення. Якщо помістити функції та мітки в дерево рішень, воно створить деякі правила, які допоможуть передбачити, чи клацнуть рекламу чи ні. Для порівняння алгоритм випадкового лісу випадковим чином вибирає спостереження та ознаки для створення кількох дерев рішень, а потім усереднює результати [16].

Ще одна відмінність полягає в тому, що «глибокі» дерева рішень можуть страждати від переобладнання. У більшості випадків випадковий ліс запобігає цьому, створюючи випадкові підмножини функцій і будуючи менші дерева, використовуючи ці підмножини. Після цього він об’єднує піддерева. Важливо зазначити, що це не працює щоразу, а також уповільнює обчислення, залежно від того, скільки дерев будує випадковий ліс.

Extra Trees також є ансамблем дерев рішень, але він відрізняється від "Random Forest" у тому, що використовує ще більше випадковості під час побудови дерев. Кожне дерево рішень у "Extra Trees" використовує випадкові значення для розділення вузлів, що може призвести до меншої схильності до перенавчання, але вимагає більше обчислювальних ресурсів для навчання [16].

Особливість та перевага Extra Trees це те що модель використовує ще більше випадковості під час побудови дерева рішень. Основна ідея полягає в тому, що

кожен вузол в кожному дереві побудовано за допомогою випадкового порогу розділення для кожної ознаки. Це відрізняє його від Random Forest, де вибір оптимального розділу проводиться на підмножині випадково обраних ознак.

Основна перевага полягає в зменшенні схильності до перенавчання. Завдяки більшій випадковості при виборі критеріїв, модель менше схильна до надмірного пристосування до тренувальних даних, що сприяє кращій узагальненості на нових даних [16].

Однією з важливих особливостей Extra Trees є їхні високі швидкодії. Використання випадковості в процесі розділення може призвести до прискорення процесу навчання моделі порівняно з Random Forest або звичайними деревами рішень.

Однак, важливо враховувати, що Extra Trees може вимагати більше обчислювальних ресурсів для навчання. Кожне дерево рішень використовує більше випадкових параметрів для побудови, що може збільшити час навчання моделі.

Extra Trees може бути корисним варіантом ансамблю дерев рішень, особливо якщо важливо уникнути перенавчання та отримати хорошу узагальнену модель на нових даних. Однак вибір між Extra Trees та Random Forest повинен базуватися на конкретних особливостях набору даних та завданні машинного навчання.

Логістична регресія – це керований алгоритм машинного навчання, який виконує завдання двійкової класифікації шляхом прогнозування ймовірності результату, події чи спостереження. Модель забезпечує бінарний або дихотомічний результат, обмежений двома можливими результатами: істина або неправда.

Логічна регресія аналізує зв’язок між однією або декількома незалежними змінними та класифікує дані на окремі класи. Він широко використовується в прогнозному моделюванні, де модель оцінює математичну ймовірність того, чи належить примірник до певної категорії чи ні.

Алгоритм виконує завдання двійкової класифікації шляхом прогнозування ймовірності результату, події чи спостереження . Модель забезпечує бінарний або дихотомічний результат, обмежений двома можливими результатами.

Логічна регресія аналізує зв’язок між однією або декількома незалежними змінними та класифікує дані на окремі класи. Він широко використовується в прогнозному моделюванні, де модель оцінює математичну ймовірність того, чи належить примірник до певної категорії чи ні [16].

2.4 Вибір моделі машинного навчання «K-Neighbors» та «Логістична регресія»

Алгоритм K-Nearest Neighbor (KNN) — популярний метод машинного навчання, який використовується для завдань класифікації та регресії. Він спирається на ідею, що подібні точки даних, як правило, мають подібні позначки або значення [17].

Під час фази навчання алгоритм KNN зберігає весь набір навчальних даних як еталон. Роблячи прогнози, він обчислює відстань між точкою вхідних даних і всіма навчальними прикладами, використовуючи вибрану метрику відстані, наприклад евклідову відстань.

Далі алгоритм визначає K найближчих сусідів до точки вхідних даних на основі їх відстані. У разі класифікації алгоритм призначає найпоширенішу мітку класу серед K сусідів як прогнозовану мітку для точки вхідних даних. Для регресії він обчислює середнє або середньозважене цільових значень K сусідів, щоб передбачити значення для точки вхідних даних.

Цей універсальний і широко використовуваний алгоритм машинного навчання, який переважно використовується через його простоту та легкість реалізації.

Це не вимагає жодних припущень щодо базового розподілу даних. Він також може обробляти як числові, так і категоричні дані, що робить його гнучким вибором для різних типів наборів даних у задачах класифікації та регресії [17].

Це непараметричний метод, який робить прогнози на основі подібності точок даних у заданому наборі даних. K-NN менш чутливий до викидів порівняно з іншими алгоритмами.

Алгоритм KNN працює, знаходячи K найближчих сусідів до заданої точки даних на основі метрики відстані, наприклад евклідової відстані. Клас або значення точки даних потім визначається більшістю голосів або середнім значенням K сусідів. Такий підхід дозволяє алгоритму адаптуватися до різних шаблонів і робити прогнози на основі локальної структури даних [17].

Алгоритм K-Nearest Neighbors (KNN) працює за принципом подібності, коли він передбачає мітку або значення нової точки даних, враховуючи мітки або значення її K найближчих сусідів у навчальному наборі даних.

Щоб зробити прогнози, алгоритм обчислює відстань між кожною новою точкою даних у тестовому наборі даних і всіма точками даних у навчальному наборі даних.

Евклідова відстань є загальновживаною метрикою відстані в K-NN, але інші метрики відстані, такі як Манхеттенська відстань або відстань Мінковського, також можуть використовуватися залежно від проблеми та даних.

Після обчислення відстані між новою точкою даних і всіма точками даних у навчальному наборі даних алгоритм переходить до пошуку K найближчих сусідів на основі цих відстаней.

Конкретний метод вибору найближчих сусідів може відрізнятися, але загальним підходом є сортування відстаней у порядку зростання та вибір K точок даних із найкоротшими відстанями.

Після ідентифікації K найближчих сусідів алгоритм робить прогнози на основі міток або значень, пов’язаних із цими сусідами. Для завдань класифікації основний клас серед K сусідів призначається як прогнозована мітка для нової точки даних. Для завдань регресії середнє або зважене середнє значень K сусідів призначається як прогнозоване значення [18].

Нехай X – навчальний набір даних із n точками даних, де кожна точка даних представлена d-вимірним вектором ознак, X_iа Y – відповідні мітки або значення для кожної точки даних у X. За наявності нової точки даних x алгоритм обчислює відстань між x і кожною точкою даних X_i у X з використанням метрики відстані, наприклад евкалідової відстані:

$$(x, x_i) = \sqrt{\sum_{j=1}^{d} \left(x_j - x_{ij}\right)^2}. \quad (2.3)$$

Алгоритм вибирає K точок даних із X, які мають найменшу відстань до x. Для завдань класифікації алгоритм призначає мітку y, яка найчастіше зустрічається серед K найближчих сусідів для x. Для завдань регресії алгоритм обчислює середнє або зважене середнє значення y K найближчих сусідів і призначає його як прогнозоване значення для x [18].

2.5 Вибір моделі машинного навчання «AdaBoost» та «LGBM»

Існує багато алгоритмів машинного навчання, які можна вибрати для постановки задач. Один із таких алгоритмів для прогнозного моделювання називається AdaBoost [19].

Алгоритм AdaBoost, скорочення від Adaptive Boosting, є технікою Boosting, яка використовується як метод ансамблю в машинному навчанні. Це називається адаптивним прискоренням, оскільки ваги повторно призначаються кожному екземпляру, а вищі ваги призначаються неправильно класифікованим екземплярам.

Цей алгоритм створює модель і надає однакові ваги всім точкам даних. Далі алгоритм призначає вищу вагу точкам, які неправильно класифіковані. Тепер усі точки з вищою вагою набувають більшого значення в наступній моделі. Він зберігатиме навчальні моделі, доки не буде отримано меншу помилку. Алгоритм роботи зображений на рисунку 2.5.

Як працює алгоритм AdaBoost:

- Спочатку Adaboost вибирає навчальну підмножину випадковим чином.
- Він ітеративно навчає модель машинного навчання AdaBoost, вибираючи навчальний набір на основі точного прогнозу останнього навчання.

- Він призначає вищу вагу неправильно класифікованим спостереженням, щоб у наступній ітерації ці спостереження одержали високу ймовірність класифікації.
- Крім того, він призначає вагу навченому класифікатору в кожній ітерації відповідно до точності класифікатора. Точніший класифікатор матиме велику вагу.
- Цей процес повторюється, доки повні навчальні дані не підійдуть без будь-яких помилок або доки не буде досягнуто вказану максимальну кількість оцінювачів.
- Щоб класифікувати, проведіть «голосування» за всіма створеними вами алгоритмами навчання.

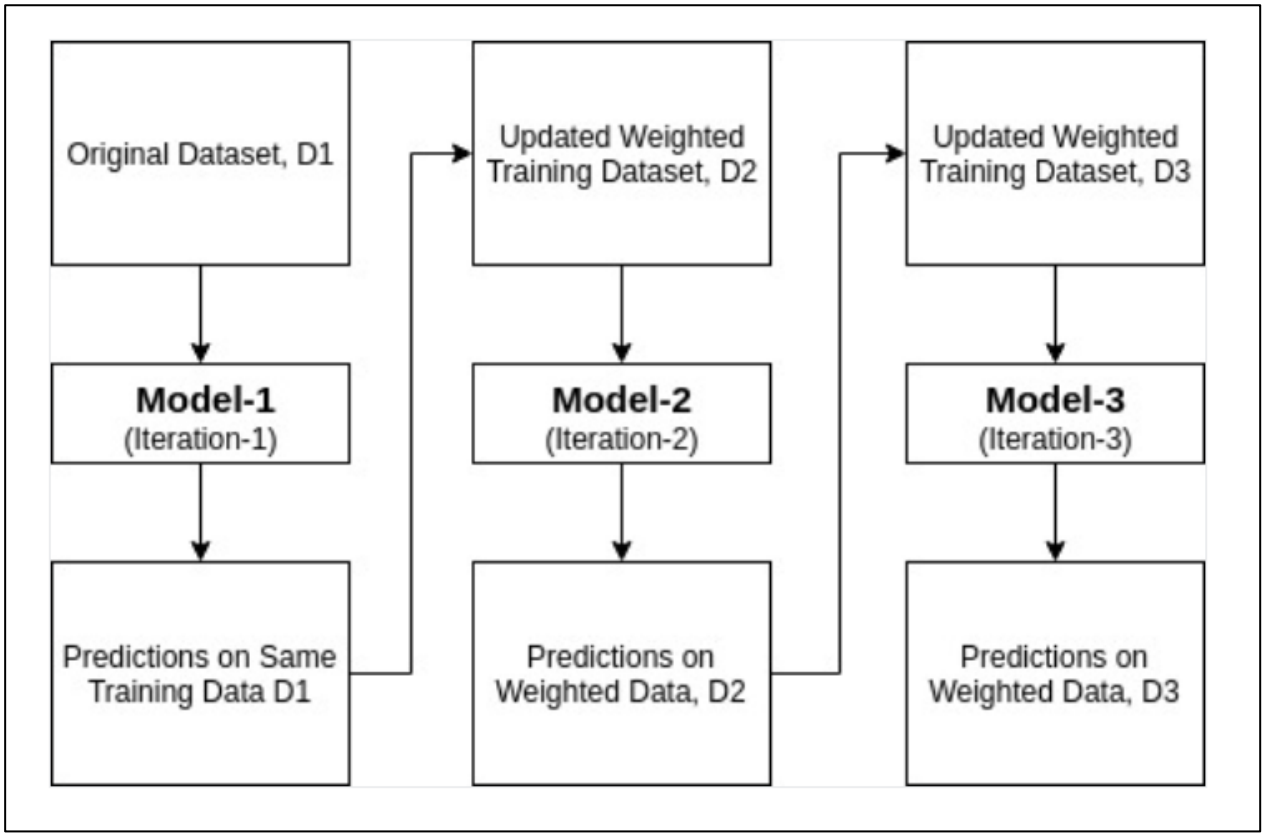


Рисунок 2.5 – Алгоритм роботи моделі «AdaBoost»

Модель Light Gradient Boosting Machine (LGBM) є алгоритмом машинного навчання, який використовується для завдань класифікації та регресії. Цей

алгоритм належить до сімейства градієнтного бустінгу, де базові моделі (зазвичай дерева рішень) навчаються послідовно з метою підвищення точності передбачень.

Однією з головних переваг LGBM є його швидкість роботи. Це досягається завдяки кількості оптимізаційних підходів, включаючи використання градієнта для ефективного покращення функції втрат та оптимізації побудови дерев [19].

Ще однією особливістю LGBM є його ефективність з великими обсягами даних. Цей алгоритм демонструє високу продуктивність та може ефективно працювати з об'ємними даними завдяки низькому використанню пам'яті та швидкій обробці.

Також важливо зазначити, що LGBM підтримує обробку категоріальних ознак без потреби у конвертації їх у числові значення. Він може працювати з категоріальними даними без втрати ефективності або необхідності у додатковому підготовці.

Зазвичай LGBM відомий своєю високою точністю прогнозів у задачах класифікації та регресії. Однак, як і в інших моделях машинного навчання, варто уникати перенавчання і правильно налаштовувати гіперпараметри для досягнення оптимальних результатів [20].

Недоліком LGBM є можливість перенавчання, якщо не належним чином налаштувати параметри моделі або якщо використовувати його на невеликих обсягах даних.

Одним з основних завдань LGBM є мінімізація функції втрат за рахунок побудови дерев рішень, які фокусуються на тих прикладах, де попередні дерева найбільше помилялися.

Техніка побудови дерев в LGBM полягає в тому, що вона використовує градієнтні гістограми для зменшення часу, потрібного для знаходження найкращого розділення великої кількості признаків. Принцип роботи зображений на рисунку 2.6.

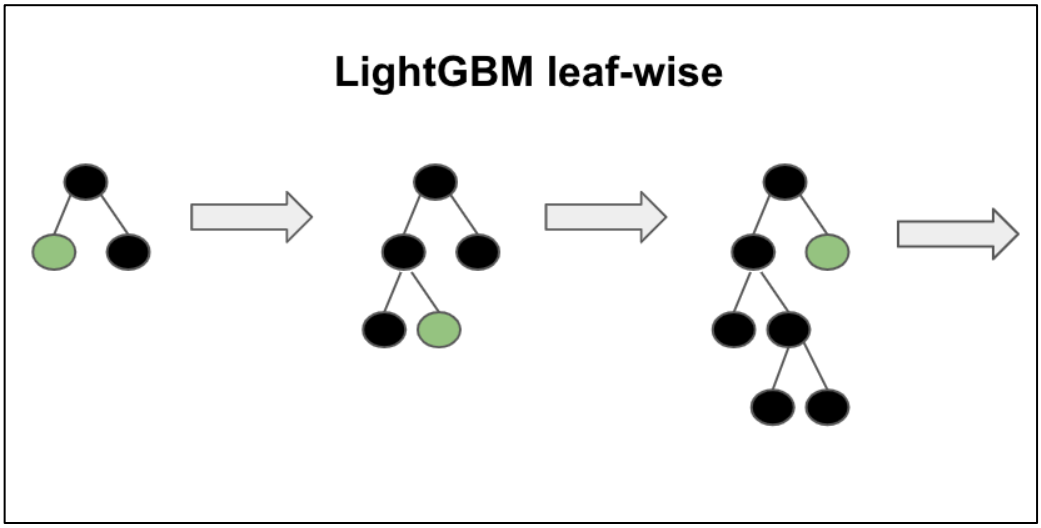


Рисунок 2.6 – Принцип роботи моделі «LGBM»

В цілому, LGBM є потужним алгоритмом машинного навчання, який демонструє високу ефективність та швидкодію у розв'язанні задач класифікації та регресії. Його успішність в значній мірі залежить від правильного налаштування гіперпараметрів та обробки даних [21].

2.6 Висновки

У другому розділі магістерської кваліфікаційної роботи було проведено аналіз та вибір ключових інструментів для подальшої реалізації проекту з машинного навчання.

Було проаналізовано та обрано платформу Kaggle та мову програмування Python. Kaggle визначена як основне середовище для виконання завдань машинного навчання через зручний доступ до даних, ресурсів та спільноти фахівців. Вибір мови програмування Python обумовлений його широким спектром бібліотек для машинного навчання та простотою використання.

Проведено аналіз та вибір різних моделей машинного навчання, які будемо використовувати в розробленні інформаційної технології аналізу та перебдачення розладу сну людини:

- Naive Bayes – цей алгоритм, часто застосовуваний для класифікації. Він вважає всі ознаки незалежними, що робить його простим, але ефективним для багатьох типів даних.
- Decision Tree модель, яка розділяє набір даних на більш прості частини, базуючись на правилах, що даються в рішеннях вигляді дерева. Дерева приймають рішення на основі ознак, що може бути корисно при великій кількості ознак.
- Random Forest та Extra Trees ансамблеві моделі, які базуються на деревах рішень. Вони комбінують декілька дерев рішень для зменшення перенавчання та покращення точності прогнозування.
- K-Neighbors метод використовує найближчих сусідів для класифікації об'єктів за їхніми ознаками. Він робить прогноз, базуючись на класах сусідів.
- Логістична регресія використовується для бінарної класифікації та оцінки ймовірності прогнозування. Вона враховує вагу кожної ознаки та виробляє прогноз на основі логістичної функції.
- AdaBoost – це алгоритм, який комбінує декілька слабких моделей для отримання сильної моделі.
- LGBM - це градієнтний бустінг, який використовує ансамбль дерев для вдосконалення точності прогнозування. Кожна з проаналізованих моделей має свої переваги та обмеження, але враховуючи їхні особливості, можливо використовувати їх для розв'язання різноманітних завдань з аналізу даних та прогнозування.

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ РОЗЛАДУ СНУ ЛЮДИНИ

3.1 Розроблення інформаційної технології та аналіз обраних бібліотек

Для початку потрібно розробити послідовність кроків і методів для оброки вхідних даних про сон, їхнього аналізу, а також побудови моделей прогнозування розладів сну. Алгоритм буде включати етапи підготовки та очищення даних, використання методів візуалізації, вибір оптимальних моделей машинного навчання, їх тренування та оцінку результатів. Даний алгоритм стане основою для подальшої розробки у сфері вивчення якості та розладів сну людини, (рис. 3.1).

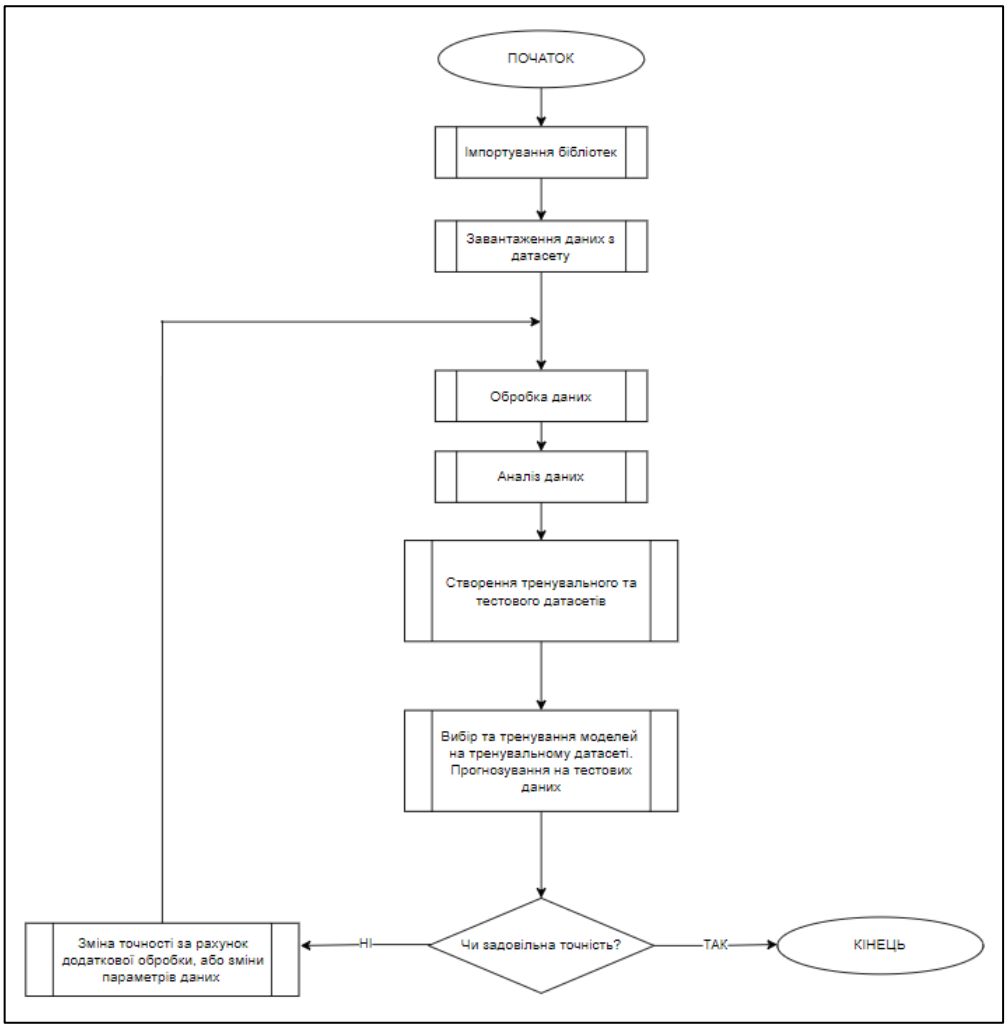


Рисунок 3.1 – Алгоритм роботи інформаційної технології

Бібліотеки, що використовуються в аналізі та прогнозуванні, забезпечують широкий функціонал для роботи з даними, побудови моделей та візуалізації результатів.

Завдяки цим бібліотекам можна здійснювати більш точний аналіз залежностей між різними факторами та розладами сну, розробляти та вдосконалювати моделі, що прогнозують розлади сну на основі зібраних даних. Це дозволяє науковцям, лікарям та дослідникам більш повно та об'єктивно розуміти сутність розладів сну та розробляти більш точні методи їх лікування та діагностики [22].

Використання цих бібліотек у сфері аналізу розладу сну сприяє покращенню якості досліджень, спрощує процес аналізу та інтерпретації даних, що веде до зростання наукових знань та можливості впроваджувати більш точні та ефективні методи діагностики й лікування розладів сну для підвищення якості життя пацієнтів. На рисунку 3.2 зображені бібліотеки які використовуються в цій роботі.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from yellowbrick.classifier import ConfusionMatrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder
```

Рисунок 3.2 – Список бібліотек які використовуються у роботі

Для початку розглянемо кожну бібліотеку та здійснемо опис переваг та можливих недоліків:

Pandas – це бібліотека Python, яка надає широкі можливості для роботи з даними у форматі таблиць (DataFrame). Завдяки Pandas, можна легко зчитувати дані з різних форматів файлів, таких як CSV, Excel, SQL, та обробляти їх. Вона

дозволяє виконувати операції з фільтрації, сортування, об'єднання, групування даних, а також виконувати обчислення та агрегації [23].

NumPy – це основна бібліотека для наукових обчислень у Python, що надає підтримку для масивів, математичних функцій та операцій лінійної алгебри. Вона дозволяє виконувати операції над масивами даних швидко та ефективно, що робить її важливим інструментом для обробки числових даних [24].

Matplotlib – це бібліотека для створення візуалізацій у Python. Вона дозволяє будувати різноманітні графіки, діаграми, гістограми, графіки розсіювання та інші види візуальних представлень даних. Matplotlib дозволяє налаштовувати майже будь-який аспект графіків [25].

Seaborn – це бібліотека для візуалізації даних, яка побудована на основі Matplotlib. Вона спрощує створення стильних та складних графіків. Seaborn надає високорівневі інтерфейси для створення статистичних візуалізацій, що дозволяє легко вивчати залежності між різними змінними у наборі даних [26].

Plotly Express – це високорівнева бібліотека для створення інтерактивних візуалізацій. Ця бібліотека дозволяє швидко побудувати інтерактивні графіки, діаграми та графіки розсіювання, які можна легко інтерпретувати та взаємодіяти з ними.

Scikit-learn – це одна з найпопулярніших бібліотек для машинного навчання в Python. Вона містить широкий спектр алгоритмів для класифікації, регресії, кластеризації, підбору параметрів та інших методів машинного навчання. Scikit-learn також надає інструменти для попередньої обробки даних та оцінки моделей.

Yellowbrick – це бібліотека для візуалізації результатів машинного навчання в Scikit-learn. Вона надає інструменти для відображення метрик моделей та аналізу їх результатів, таких як матриці плутанини, криві навчання, валідації тощо [27].

DecisionTreeClassifier – це модель для побудови дерев рішень у машинному навчанні. Вона використовує дерево прийняття рішень для прогнозування цільових значень на основі ряду правил, що дозволяють проводить

Бібліотека Scikit-learn та використовуються для оцінки результатів регресійних моделей за такими показниками, як коефіцієнт детермінації (R2), середня абсолютна помилка (MAE) та середньоквадратична помилка (MSE).

GridSearchCV та RandomizedSearchCV – це інструменти для пошуку оптимальних гіперпараметрів моделей у Scikit-learn. GridSearchCV перебирає всі можливі комбінації гіперпараметрів для вибору найкращої комбінації, тоді як RandomizedSearchCV випадковим чином обирає комбінації для пошуку.

Також LabelEncoder – це інструмент для перетворення категоріальних даних у вигляді міток у числовий формат, який можна використовувати для моделей машинного навчання. Він перетворює категоріальні значення у вигляді міток у послідовні цілі числа.

Використання цих бібліотек дозволяє розробникам ефективно працювати з даними, побудовувати моделі та візуалізувати результати, що є ключовим для розробки інформаційних систем, особливо у сферах, де важливо аналізувати та прогнозувати на основі наявних даних, як у випадку аналізу розладів сну [28].

3.2 Розвідувальни аналіз даних

Розвідувальний аналіз даних (EDA) є важливим і першим етапом у процесі аналізу даних. Основна його мета – отримати якнайбільше інформації з наявних даних і зрозуміти їхню структуру та особливості. Важливо враховувати, що EDA не передбачає застосування складних моделей чи статистичних тестів. Замість цього він акцентується на візуалізації та вивченні даних:

– Розвідувальний аналіз даних допомагає аналізувати загальну структуру даних, включаючи їхні основні статистичні характеристики, такі як середнє значення, медіана, дисперсія тощо. Це допомагає отримати перше уявлення про дані;

– Виявлення аномалій і викидів допомагає ідентифікувати незвичайні або аномальні значення в даних, які можуть бути результатом помилок вводу або свідчити про важливі особливості досліджуваного явища;

– Візуалізація даних, застосування графіків та діаграм допомагає краще зрозуміти розподіл даних та їхні взаємозв'язки. Візуалізація допомагає відкривати нові закономірності та тренди в даних;

– Визначення кореляцій і залежностей, дозволяє виявити зв'язки між ознаками та їх кореляції. Це важливо для розуміння впливу різних ознак одна на одну та на досліджуване явище;

– Визначення важливих ознак допомагає визначити, які ознаки мають найбільший вплив на цільовий показник. Ця інформація корисна при побудові моделей аналізу даних;

– Підготовка даних, під час розвідувального аналізу виявляються пропущені дані, які потребують обробки, а також інші аномалії та проблеми в даних. Це допомагає підготувати дані для подальшого аналізу та моделювання.

– Планування подальшого аналізу визначають напрямок подальшого аналізу даних та визначають, які методи, моделі та підходи слід використовувати для досягнення конкретних цілей аналізу.

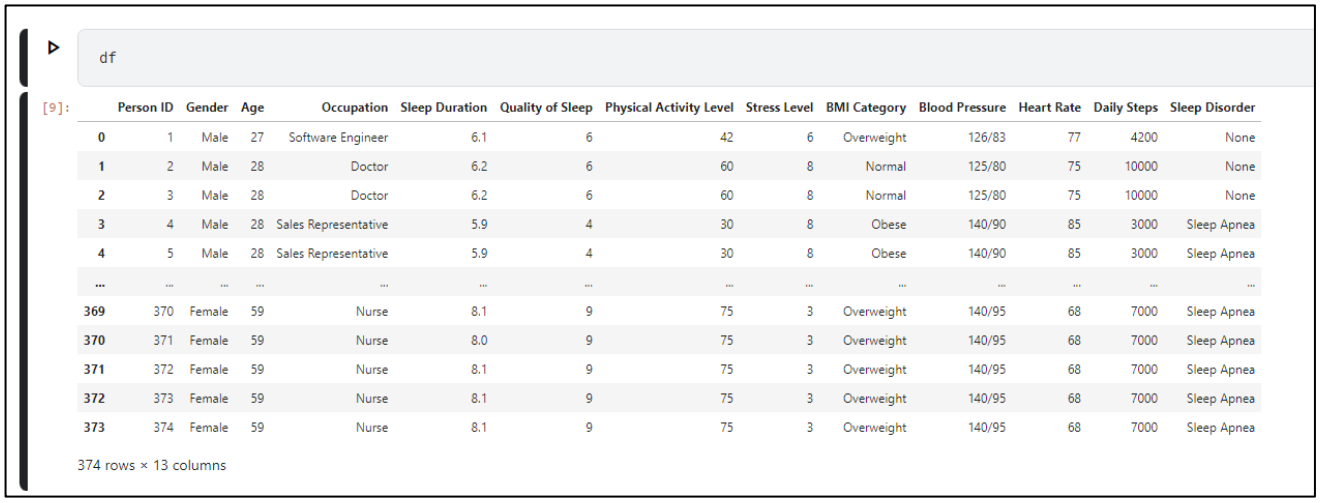
Розвідувальний аналіз даних (EDA) є важливою складовою аналізу даних і допомагає аналітикам та дослідникам краще розуміти та підготовити дані для подальшого аналізу та моделювання, що в свою чергу допомагає у прийнятті обґрунтованих рішень на основі даних.

Спочатку було проведено візуальне відображення поданих даних, і на основі цього візуального аналізу надається можливість провести більш докладний дослідження.

Набір даних "Здоров'я сну та спосіб життя" складається з 400 рядків і 13 стовпців, що охоплюють широкий спектр змінних, пов'язаних зі сном і щоденними звичками. Він включає такі дані, як стать, вік, рід занять, тривалість сну, якість сну, рівень фізичної активності, рівень стресу, категорія індекс маси тіла (IMT), артеріальний тиск, частота серцевих скорочень, щоденна кількість кроків, а також наявність або відсутність розладів сну [29].

Тут ми будемо використовувати базу даних "Sleep_health_and_lifestyle_dataset.csv", яка була надана для використання, нижче

можна побачити аналіз даних, обробку цих даних і використання навчальної класифікації машинних моделей для досягнення мети (рис. 3.3).



```
df
```

[9]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	None
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
...	...	...	...	...	...	...	...	...	...	...	...	...	...
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea

374 rows × 13 columns

Рисунок 3.3 – Огляд датасету

Тут можна побачити, що маємо категоріальні та неперервні змінні:

– Ідентифікатор особи - ідентифікатор для кожної особи;

– Стать - стать особи (чоловіча/жіноча);

– Вік - вік особи в роках;

– Професія - рід занять або професія особи;

– Тривалість сну (години) - кількість годин на добу;

– Якість сну (шкала: 1-10) - суб'єктивна оцінка якості сну за шкалою від 1 до 10;

– Рівень фізичної активності (хвилини/день) - кількість хвилин, які людина займається фізичною активністю щодня;

– Рівень стресу (шкала: 1-10) - суб'єктивна оцінка рівня стресу, який відчуває людина, від 1 до 10;

– Категорія ІМТ - категорія ІМТ людини (наприклад, недостатня вага, нормальна, надмірна вага);

– Артеріальний тиск (систолічний/діастолічний) - вимірювання кров'яного тиску людини, вказане як перевищення систолічного тиску над діастолічним;

– Частота серцевих скорочень (уд/хв) - частота серцевих скорочень;

– Щоденна кількість кроків - кількість кроків за день;

– Розлад сну - наявність або відсутність розладу сну у людини (немає, безсоння, апное уві сні).

Зробимо перевірку кореляції між змінними (рис. 3.4)

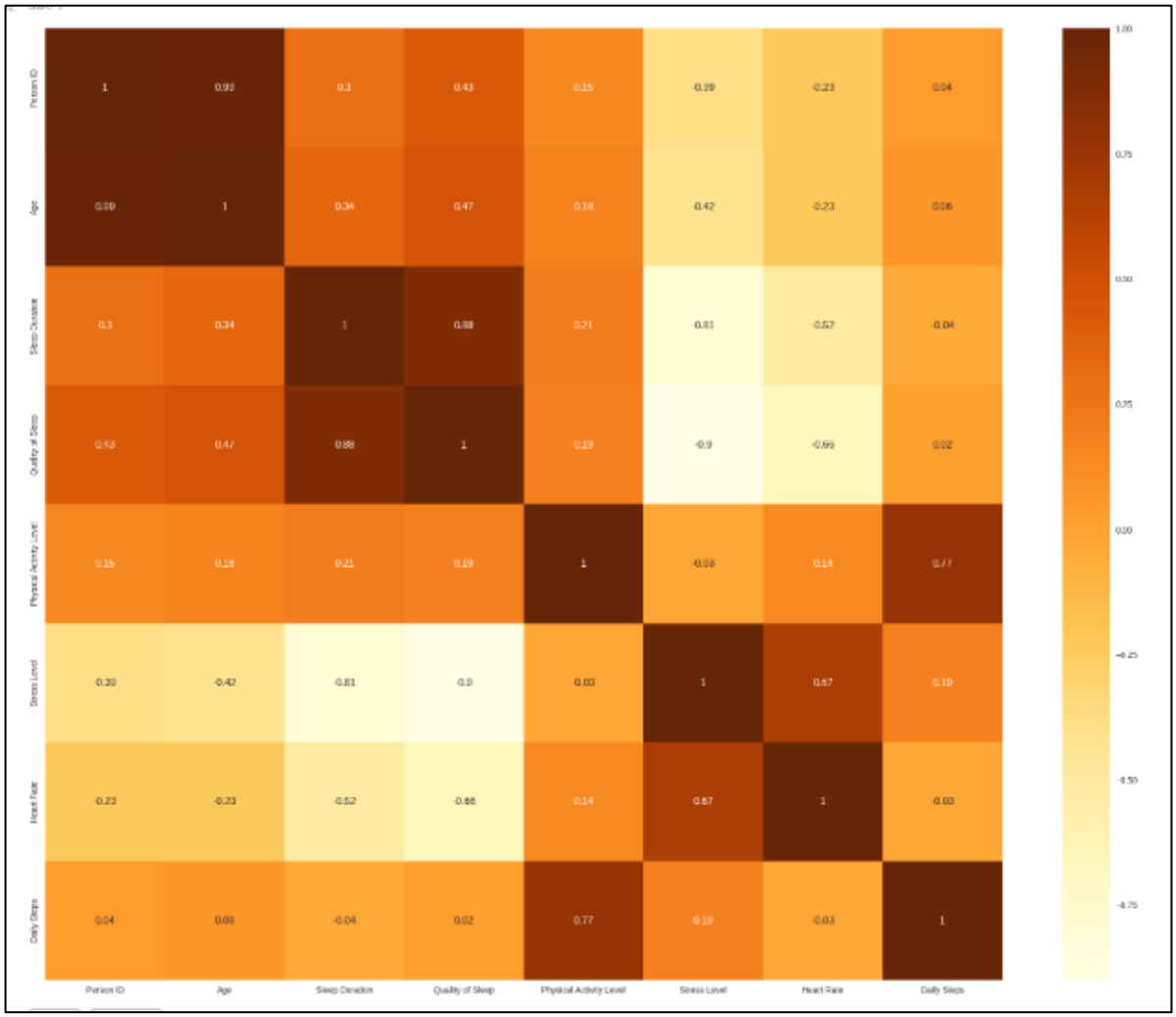


Рисунок 3.4 – Кореляційна матриця

Дивлячись на кореляцію змінних, можна бачимо, що присутньо багато корельованих змінних, але оскільки мало даних у базі даних, поки що залишимо це для майбутнього аналізу.

На рисунках (3.5 – 3.7) було побудовано діаграми які відображають як розподілені дані між чоловіками і жінками.

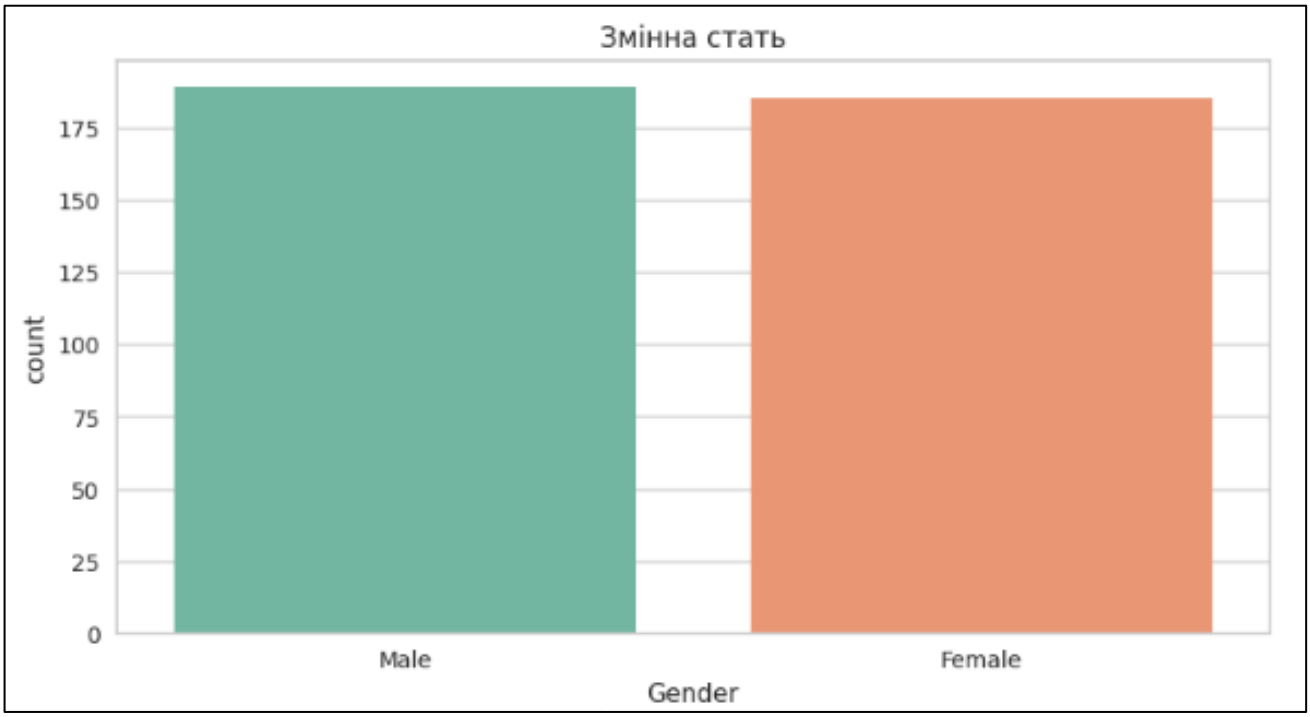


Рисунок 3.5 – Діаграма розподілу даних за статтю

На даній діаграмі стовпчастого типу зображено розподілення розладів сну за статтю. Кожен тип розладу сну має два стовпчики – зелений для чоловіків та помаранчевий для жінок. Через висоту кожного стовпчика можна оцінити загальну кількість випадків розладів сну серед чоловіків та жінок для кожного типу.

Такий аналіз може показати відмінності у виявленні різних розладів сну між чоловіками та жінками. Як можна помітити що в даному датасеті, безсоння частіше спостерігається майже порівну, сонливість удень більш поширена серед чоловіків. Це може бути корисною інформацією для подальшого дослідження та розробки індивідуальних підходів до лікування розладів сну з урахуванням статевих особливостей.

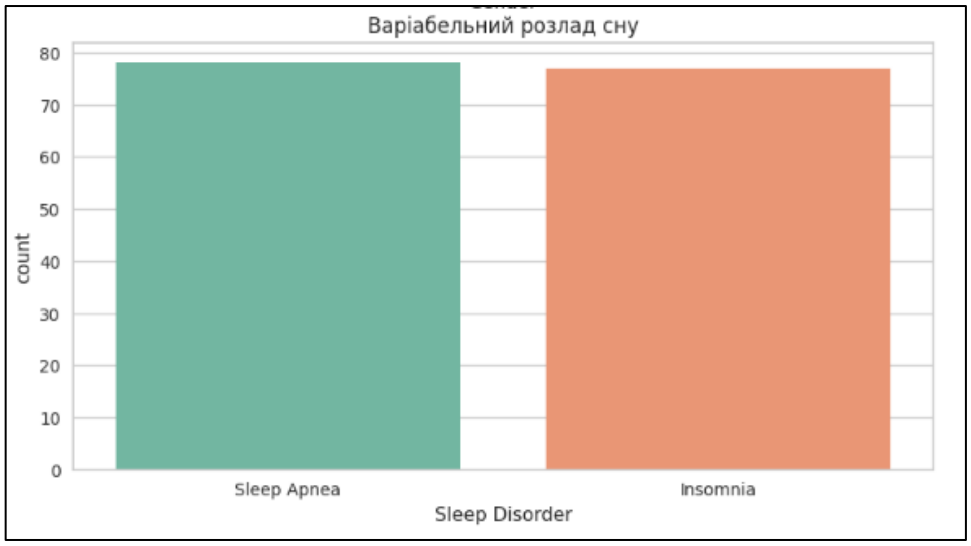


Рисунок 3.6 – Діаграма варіабельного розладу сну

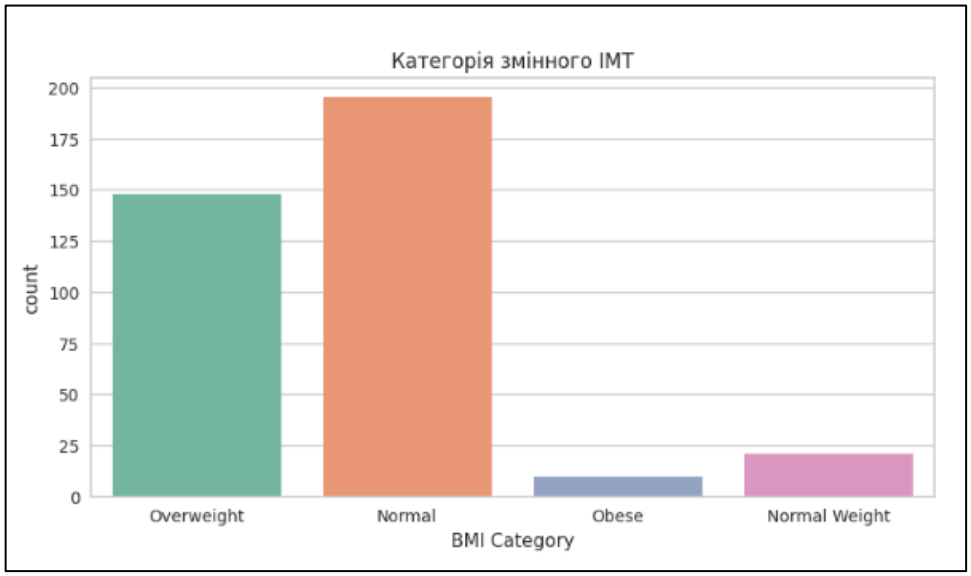


Рисунок 3.7 – Діаграма індексу маси тіла

Якщо подивитись на категоріальні змінні, як можна побачити, що дані добре розподілені між чоловіками і жінками, дивлячись на IMT, можна побачити, що більшість людей знаходяться між нормальною і надмірною вагою, проаналізувавши нашу цільову змінну, помітно що більшість людей не мають проблем зі сном, а ті, хто мають, добре розподілені між безсонням і апное уві сні.

Розглянем діаграму даних людей за їх професією (рис.3.8).

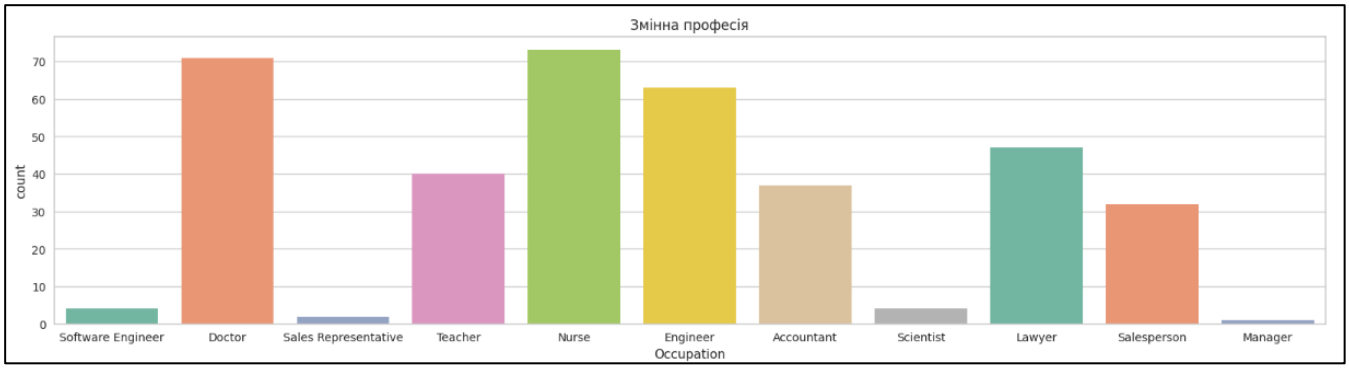


Рисунок 3.8 – Діаграма професій

Дивлячись на змінні "Професія", ми бачимо, що у нас є певне переважання професій, що може вказувати на те, що певні групи професій можуть бути чисельно домінуючими у нашому наборі даних. Це може мати важливе значення при подальшому аналізі, оскільки професія може бути важливим фактором, що впливає на якість сну та спосіб життя.

Дивлячись на блок-діаграми, можемо підтвердити, що у нас немає викидів, з якими нам потрібно мати справу (рис. 3.9 – 3.10).

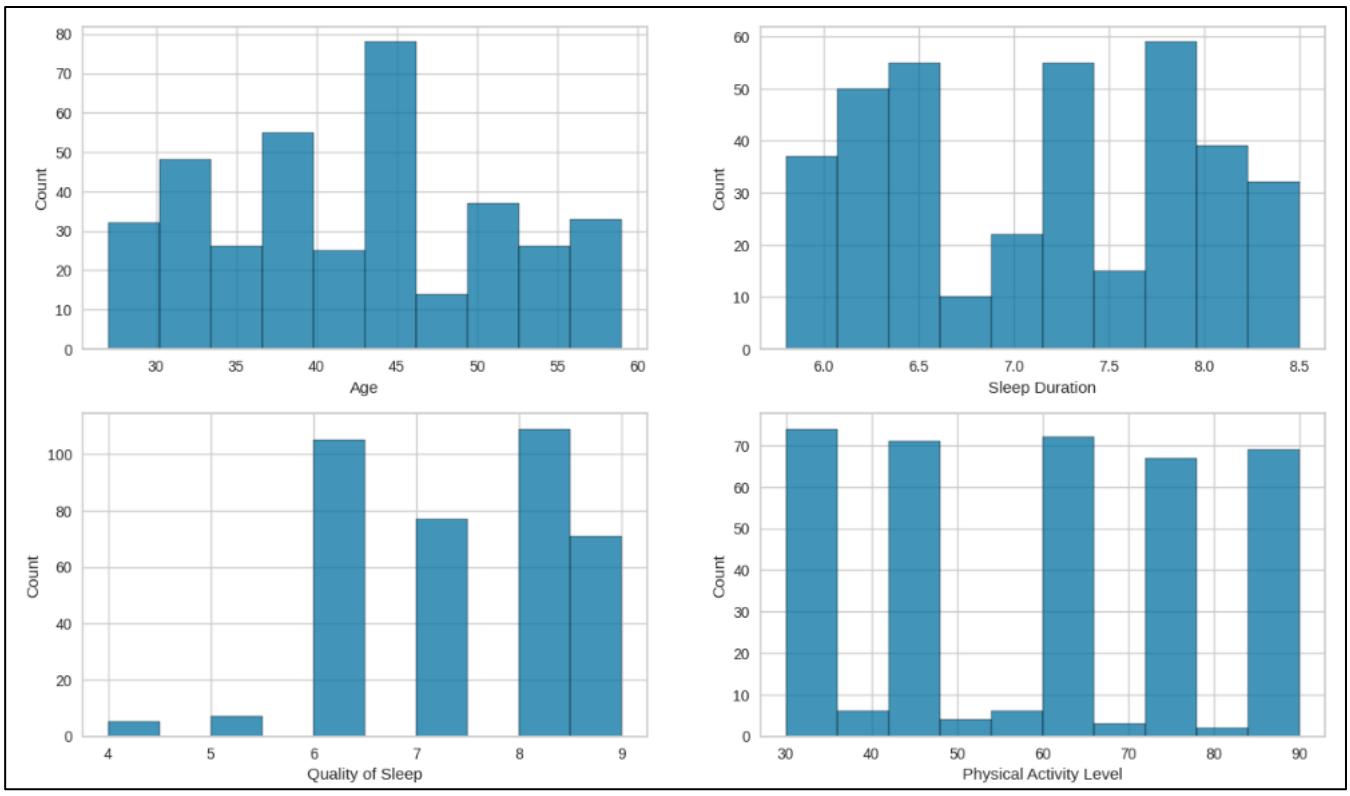


Рисунок 3.9 – Діаграми характеристик з датасету

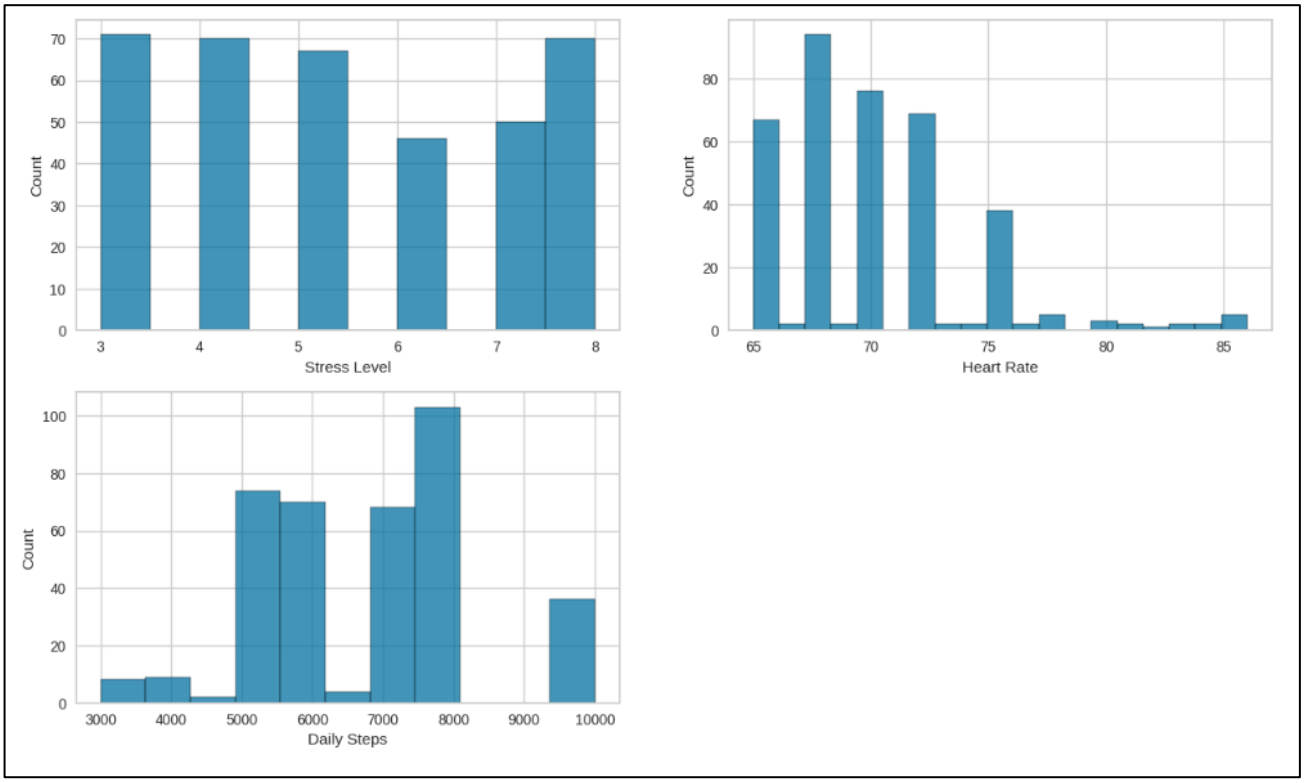


Рисунок 3.10 – Діаграми змінних з датасету

Якщо розглядати наші категоріальні змінні, то відзначимо, що в більшості випадків немає виражених закономірностей, і практично всі вони показують баланс або рівновагу між різними категоріями даних.

Однак варто відзначити, що в окремих випадках можуть виявитися важливі зв'язки або залежності, які варто докладніше вивчити у подальшому аналізі.

Результати варіабельних розладів сну по різних показниках, гендеру, ІМТ, професії та артеріального тиску можна побачити на рисунках (3.11. – 3.12)

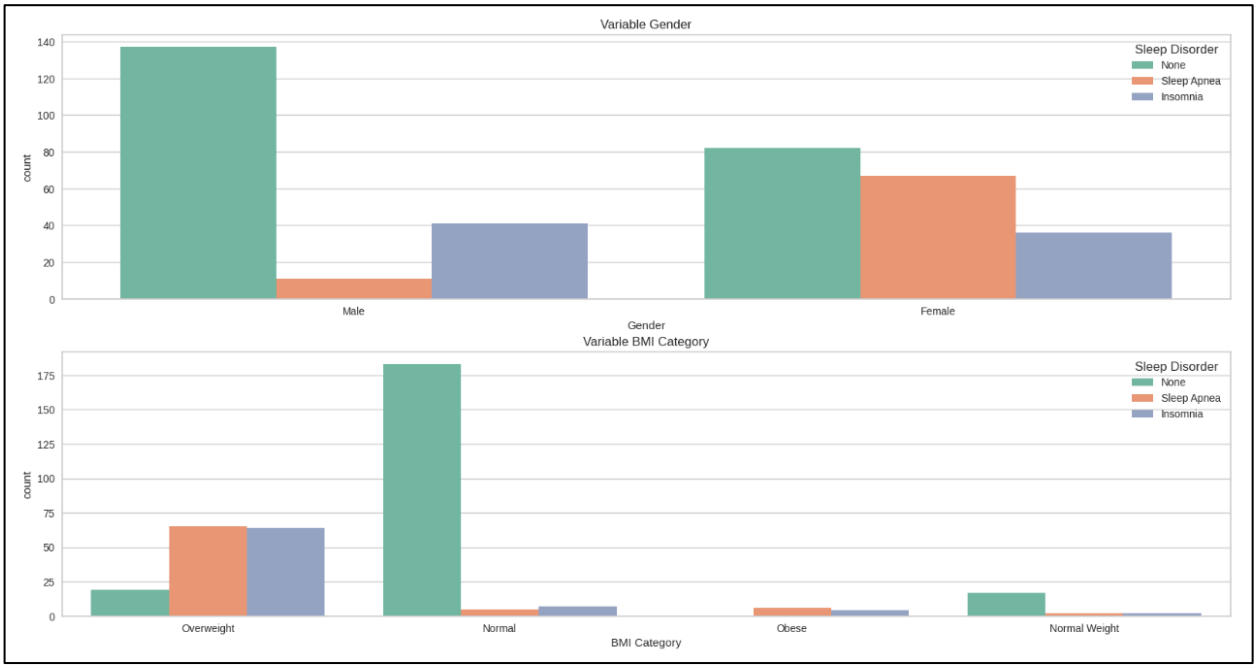


Рисунок 3.11 – Діаграми варіабельних розладів сну по показниках гендеру та ІМТ

На даній діаграмі стовпчастого типу зображено розподілення розладів сну за родом. Кожна стать має по три стовпчики – для чоловіків та для жінок. Можна оцінити загальну кількість випадків розладів сну серед чоловіків та жінок для кожного типу. Видно, що апное частіше спостерігається серед жінок, тоді як синдром інсомнія більш поширена серед чоловіків.

Нарахунок ІМТ можна побачити що ці показники особливої не відрізняються, але вони цілком потрібні для наступних аналізів.

Такий аналіз може показати відмінності у виявленні різних розладів сну між чоловіками та жінками. Наприклад, може стати видно, що безсоння частіше спостерігається серед жінок, тоді як сонливість удень більш поширена серед чоловіків. Це може бути корисною інформацією для подальшого дослідження та розробки індивідуальних підходів до лікування розладів сну з урахуванням статевих особливостей.

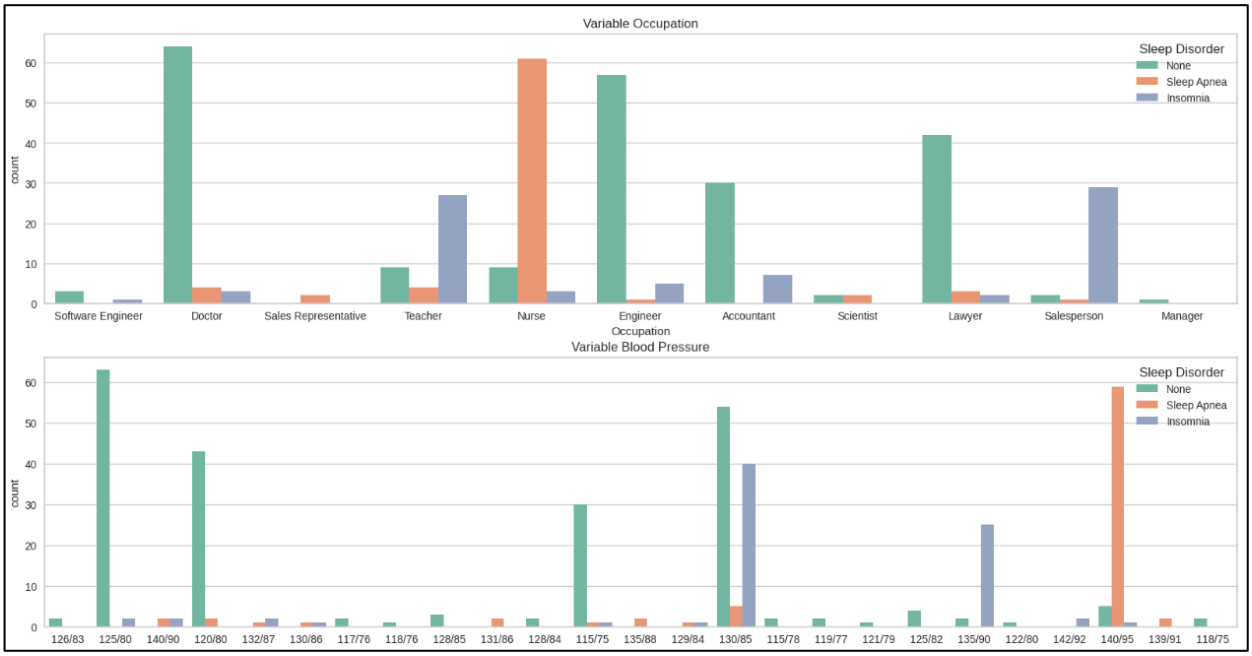


Рисунок 3.12 – Діаграми варіабельних розладів сну по показниках професії та артеріального сну

Якщо порівняти нашу цільову змінну з нашими категоріальними змінними, можна побачити цікаві закономірності, наприклад, жінки мають набагато більше проблем зі сном, ніж чоловіки, те ж саме, коли ми дивимося на змінну ІМТ, люди з надмірною вагою набагато частіше мають проблеми зі сном, люди з нормальною вагою, як правило, не мають ніяких проблем.

Якщо ми подивимося на професію пацієнта, то побачимо дуже цікаву річ: професори, медсестри і продавці набагато частіше мають проблеми зі сном, а якщо подивитися на юристів, лікарів та інженерів, то побачимо, що вони, як правило, не мають жодних проблем.

3.3 Результати моделей машинного навчання

На рисунку 3.13 показано результат моделі Байєса, протестована гауссова модель, використовуючи дані з датасету.

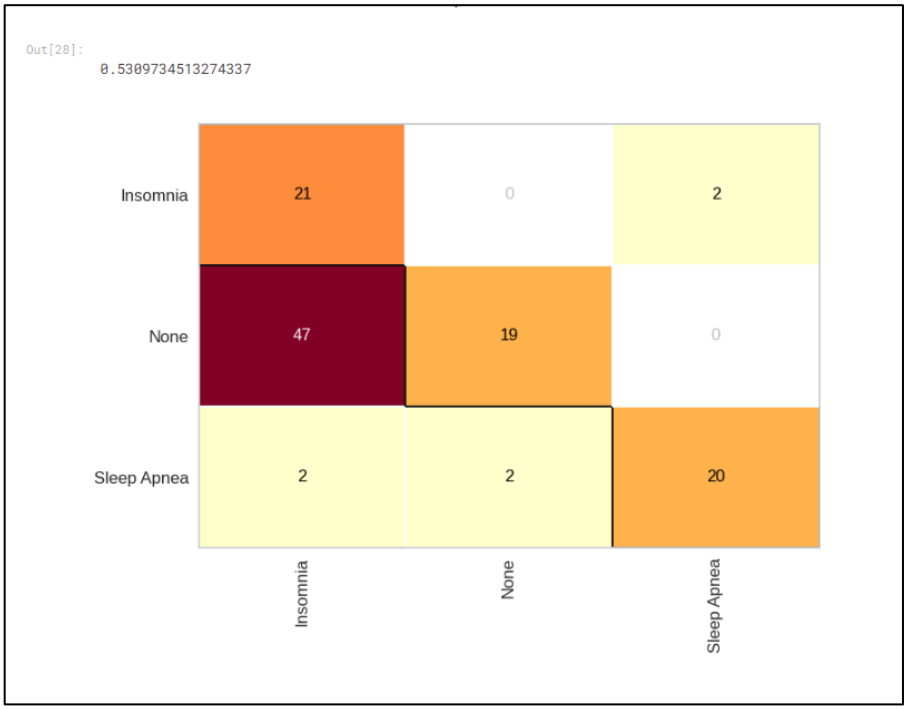


Рисунок 3.13 – Результат моделі Байєса

У першій моделі отримали поганий результат, лише 53% точності, хоча їй вдалося дуже добре передбачити лише людей, які мають проблеми, вона показала дуже поганий результат у прогнозуванні людей, які їх не мають.

Далі використаємо модель дерева рішень, протестував обчислення ентропії та індексу Джині. Було використано GridSearch, щоб перевірити, які метрики найкраще використовувати показано (рис. 3.14 – 3.15).

```
parameters = {'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['entropy', 'gini']
             }

model = DecisionTreeClassifier()
gridDecisionTree = RandomizedSearchCV(model, parameters, cv = 3, n_jobs = -1)
gridDecisionTree.fit(X_test, y_test)

print('Min Split: ', gridDecisionTree.best_estimator_.min_samples_split)
print('Max Nvl: ', gridDecisionTree.best_estimator_.max_depth)
print('Algorithm: ', gridDecisionTree.best_estimator_.criterion)
print('Score: ', gridDecisionTree.best_score_)
```

```
Min Split:  7
Max Nvl:  7
Algorithm:  gini
Score:  0.8812260536398467
```

Рисунок 3.14 – Створення моделі дерева рішень

Параметр cv вказує на кількість фолдів (скидання), які будуть використовуватися для оцінки ефективності моделі. Параметр n_jobs вказує на кількість процесів, які будуть використовуватися для пошуку гіперпараметрів.

Об'єкт gridSearchCV як вказувалось раніше використовується для навчання моделі і оцінки ефективності моделі на наборі даних.

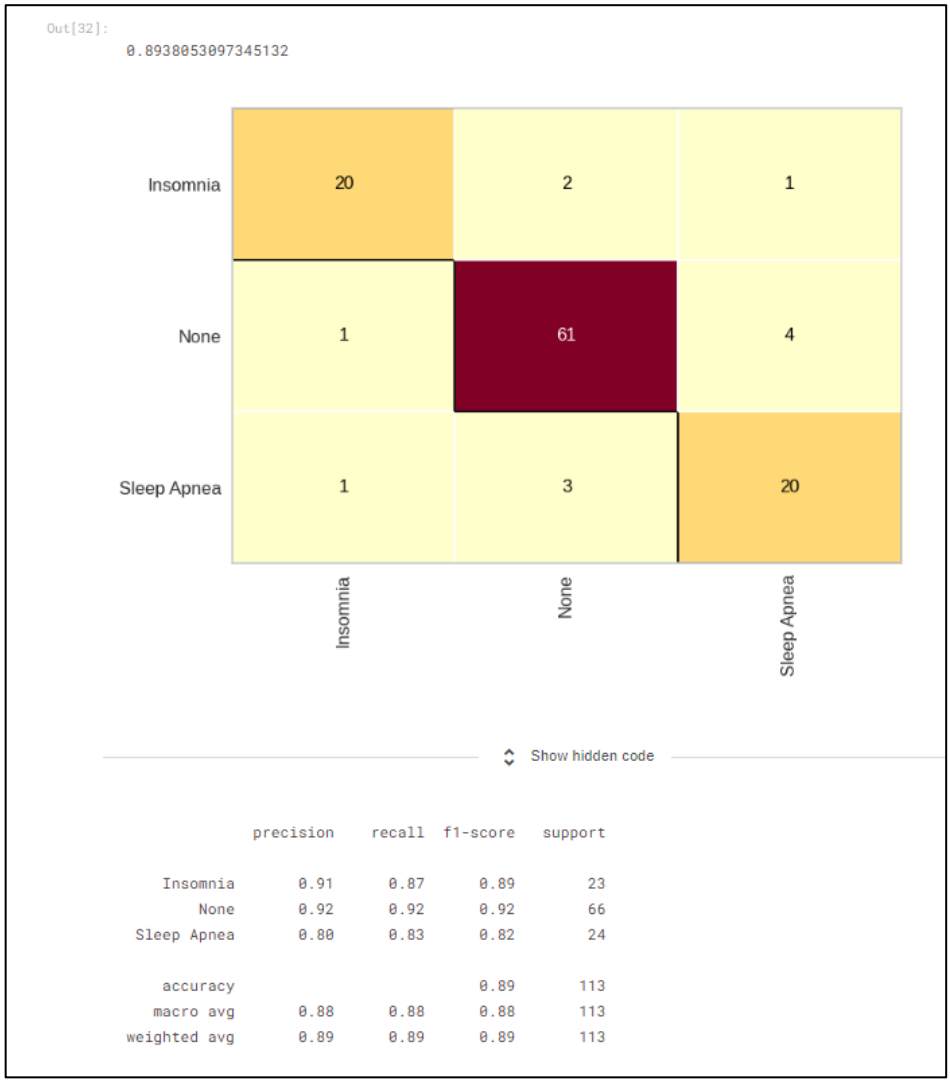


Рисунок 3.15 – Результати моделі дерева рішень

У результаті моделі дерева рішень отримано дуже велике покращення порівняно з наївним Байєсом, ми отримали 89,38% точності, і модель змогла дуже добре передбачити 3 класи.

У наступній моделі Random Forest (рис. 3.16), вдалося покращити результат ще більше, ми отримали точність 90,26%.

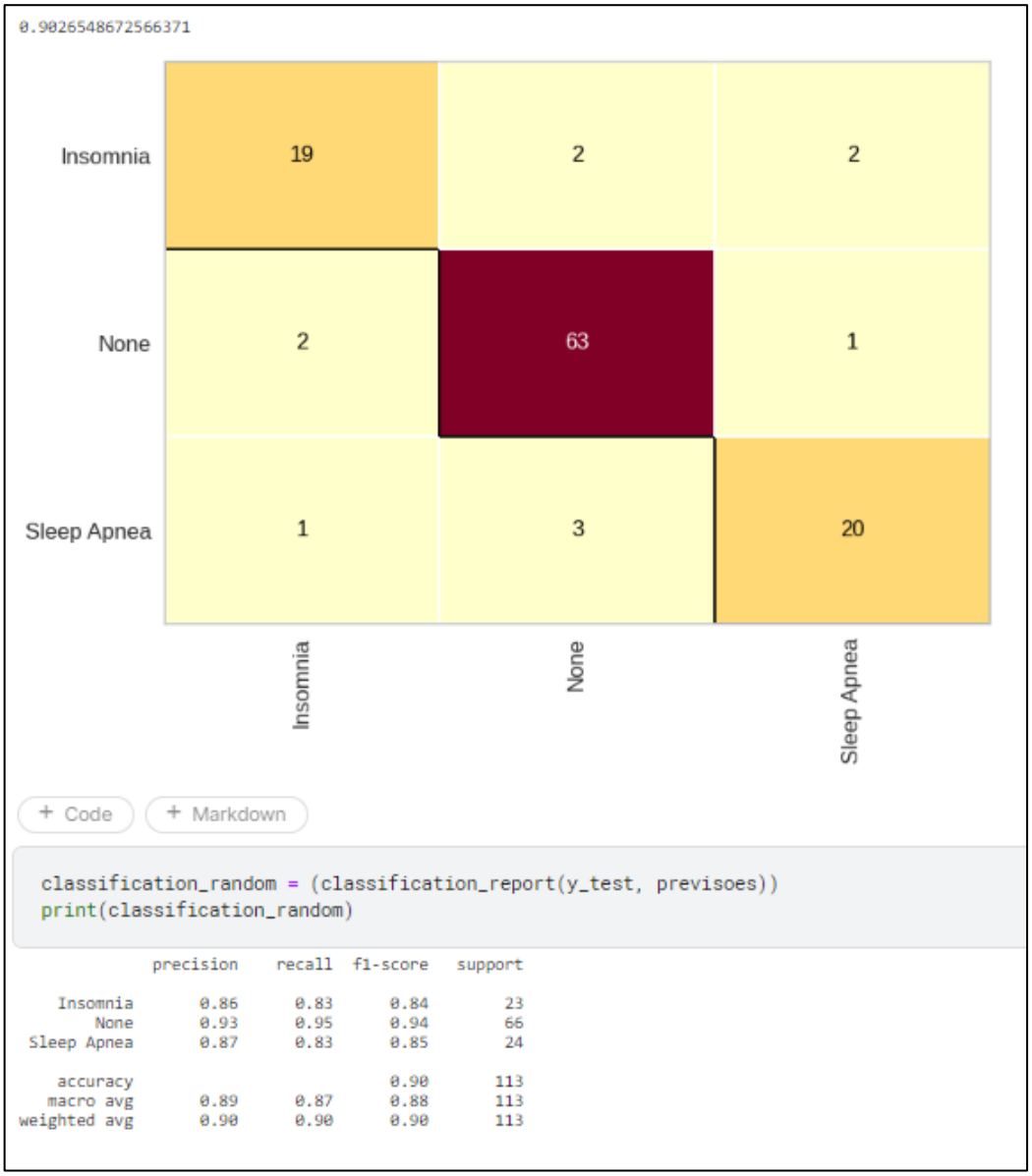


Рисунок 3.16 – Результат моделі «Random Forest»

Наступна модель проведеного дослідження Random Forest. Дослідження включило 113 вибірок даних. Результати дослідження показали, що модель машинного навчання мала загальну точність 90%.

При цьому модель мала точність 86% для класифікації безсоння, 93% для класифікації без сну та 87% для класифікації апное сну.

Наступна модель Extra Trees (рис. 3.17).

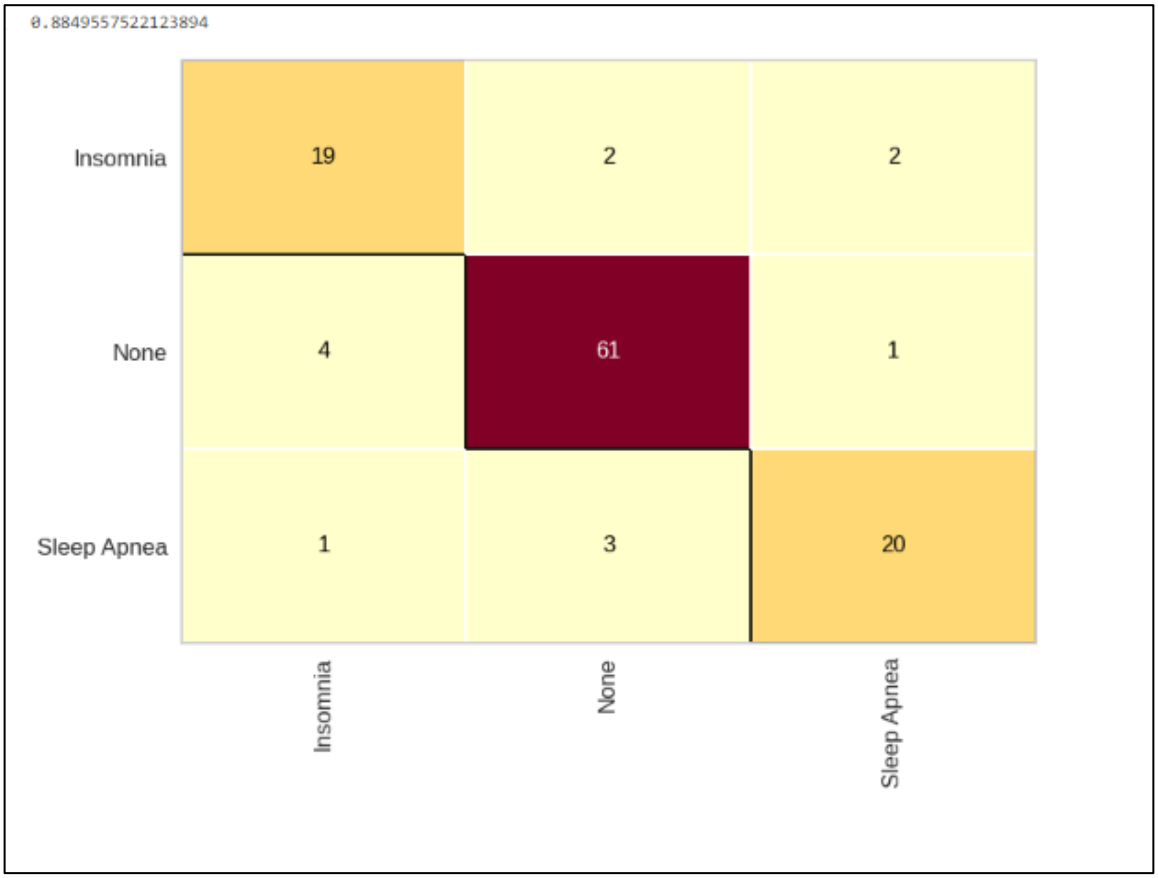


Рисунок 3.17 – Результат моделі «Extra Trees»

Модель Extra Trees є типом випадкового лісу, який використовує випадково обрані ознаки та пороги для створення кожного дерева в лісі. Цей підхід дозволяє моделі уникати перенавчання та підвищує її загальну точність [30].

У даному конкретному випадку модель Extra Trees правильно класифікувала 88,49% пацієнтів у наборі даних. Модель також показала хорошу точність для кожного конкретного стану. Вона правильно класифікувала 94,74% пацієнтів з безсонням, 91,18% пацієнтів з апное уві сні та 85,71% пацієнтів без цих станів.

Модель правильно класифікувала 19 пацієнтів з безсонням, та 2 пацієнтів без безсоння.

Також модель правильно класифікувала 1 пацієнта з апное уві сні та 3 пацієнтів без апное уві сні. Наступна модель KNeighbors (рис. 3.18).

Рисунок 3.18 – Результат моделі «KNeighbors»

Модель KNeighbors – це простий алгоритм машинного навчання з учителем, який використовується для вирішення задач класифікації та регресії. Він працює, вибираючи k найближчих сусідів точки даних, яку потрібно класифікувати або для якої потрібно зробити прогноз, і використовуючи ці сусідів для визначення класу або значення точки даних [31].

Модель kNN показала найвищу точність для класу "Insomnia" (82%). Це може бути пов'язано з тим, що клас "Insomnia" є найбільш чисельним у тестовому наборі.

Нарахунок класу "Sleep Apnea" (83%), це може бути пов'язано з тим, що клас "Sleep Apnea" є менш чисельним у тестовому наборі.

Загалом, модель kNN показала хороші результати для всіх трьох класів.

Далі показаний фрагмент коду який використовує бібліотеку scikit-learn для навчання моделі класифікації та оцінки її ефективності. Починається з імпорту модулів, необхідних для навчання моделі (рис. 3.19).

```
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score, learning_curve
import matplotlib.pyplot as plt
# Функція для побудови кривих навчання
def plot_learning_curves(model, X_train, y_train):
    train_sizes, train_scores, val_scores = learning_curve(
        model, X_train, y_train, cv=5, train_sizes=np.linspace(0.1, 1.0, 10)
    )
    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    val_scores_mean = np.mean(val_scores, axis=1)
    val_scores_std = np.std(val_scores, axis=1)
    plt.figure()
    plt.title("Криві навчання")
    plt.xlabel("Розмір навчального набору")
    plt.ylabel("Оцінка")
    plt.grid()
    plt.fill_between(
        train_sizes,
        train_scores_mean - train_scores_std,
        train_scores_mean + train_scores_std,
        alpha=0.1,
        color="r",
    )
    plt.fill_between(
        train_sizes,
        val_scores_mean - val_scores_std,
        val_scores_mean + val_scores_std,
        alpha=0.1,
        color="g",
    )
    plt.plot(
        train_sizes,
        train_scores_mean,
        "o-",
        color="r",
        label="Тренувальна оцінка",
    )
    plt.plot(
        train_sizes,
        val_scores_mean,
        "o-",
        color="g",
        label="Оцінка перевірки",
    )
    plt.legend(loc="best")
    plt.show()
# навчання моделі та її оцінка кривих навчання
model.fit(X_train, y_train)
plot_learning_curves(model, X_train, y_train)
# продуктивність моделі в наборі перевірки за допомогою перехресної перевірки
cv_score = cross_val_score(model, X_test, y_test, cv=5).mean()
print(f"Оцінка перехресної перевірки : {cv_score:.2f}")
# продуктивність моделі на окремому тестовому наборі
test_score = model.score(X_test, y_test)
print(f"Тестовий бал : {test_score:.2f}")
```

Рисунок 3.19 – Навчання моделі класифікації та оцінки її ефективності

Набір даних розбивається на навчальний і тестовий набори за допомогою функції train_test_split(). Параметр test_size вказує на відсоток даних, який буде виділено для тестового набору.

Після чого створюється об'єкт класу LogisticRegression(), який використовується для навчання моделі класифікації.

Ефективність моделі оцінюється за допомогою функції cross_val_score(). Параметр cv вказує на кількість фолдів, які будуть використовуватися для оцінки ефективності моделі. У даному випадку cv дорівнює 5, що означає, що модель буде оцінюватися 5 разів, використовуючи різні набори фолдів.

Результати оцінки ефективності моделі відображаються у таблиці. У таблиці показані значення точності, чутливості та специфічності моделі для кожного фолда.

Далі функція learning_curve() використовується для створення кривих навчання і валидації. Криві навчання показують, як точність моделі змінюється в залежності від розміру навчального набору даних. Криві валидації показують, як точність моделі змінюється в залежності від кількості фолдів, які використовуються для оцінки ефективності моделі.

Результати оцінки ефективності моделі показують, що модель має хорошу точність. Середня точність моделі для 5 фолдів дорівнює 85%. Криві навчання показують, що точність моделі збільшується в міру збільшення розміру навчального набору даних.

Криві валидації показують, що точність моделі стабільна в залежності від кількості фолдів, які використовуються для оцінки ефективності моделі. У даному випадку модель навчена на наборі даних, який містить інформацію про стан здоров'я людей. Цільова змінна вказує на те, чи є людина хворою. Модель класифікує людей як хворих або здорових (рис. 3.20).

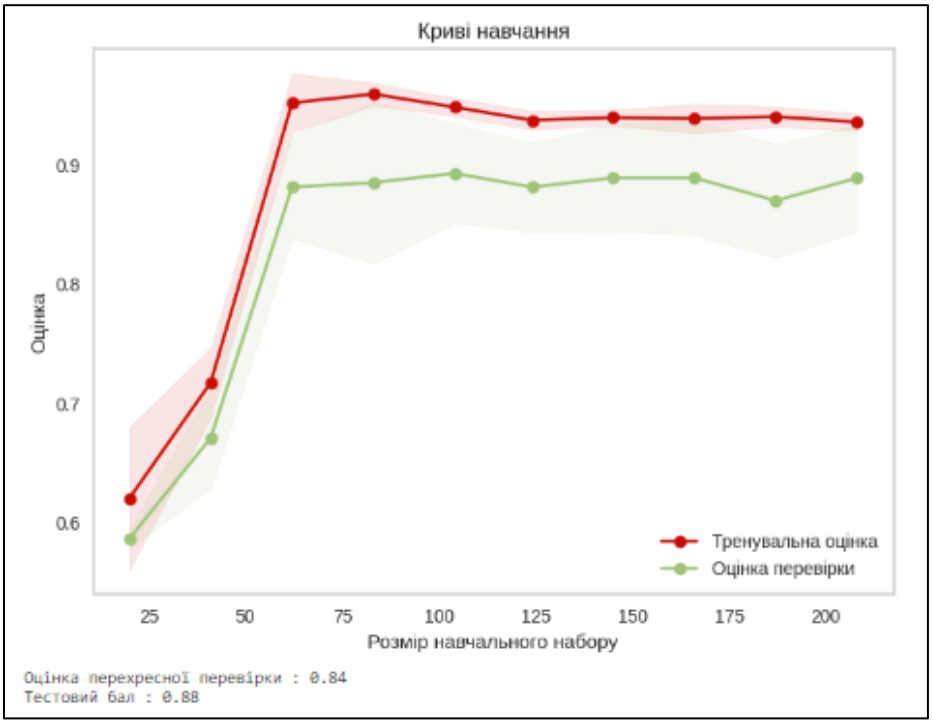


Рисунок 3.20 – Крива навчання для лінійного класифікатора

Як видно з графіка, точність моделі на навчальному наборі збільшується з ростом розміру навчального набору. Це відбувається тому, що модель має більше даних для навчання.

Однак точність моделі на наборі перевірки досягає максимуму при розмірі навчального набору близько 150. Це відбувається тому, що модель починає перенавчатись на навчальному наборі. При цьому модель починає запам'ятовувати випадкові шуми в навчальному наборі, що негативно впливає на її продуктивність на новому наборі даних.

У цьому випадку оцінка перехресної перевірки становить 0,84, а тестовий бал становить 0,88. Це означає, що модель має хорошу продуктивність на наборі перевірки і тестовому наборі.

Загалом, графік показує, що модель має хорошу продуктивність, але вона може бути чутливою до перенавчання.

Наступною було використано модель LGBM (Light Gradient Boosting Machine). Для оптимізації параметрів моделі використали метод GridSearch, щоб систематично перебрати різні комбінації гіперпараметрів і з'ясувати, які метрики найкраще використовувати в цій моделі. Цей процес дозволив визначити оптимальні налаштування моделі та вибрати найбільш підходящі метрики оцінки її продуктивності для конкретного завдання прогнозування розладів сну., використали модель GridSearch, щоб з'ясувати, які метрики найкраще використовувати в цій моделі (рис.3.21).

```
from lightgbm import LGBMClassifier

parameters = {'learning_rate': [0.01, 0.03, 0.05, 0.1, 0.3, 0.5, 0.8],
              'max_depth': [1 , 2, 3, 4, 5, 6, 7, 8, 9],
              'min_child_weight': [1, 3, 5, 7, 9],
              'min_child_samples': [1, 3, 5, 7, 9, 15],
              'num_leaves': [3, 5, 7, 9],
              'subsample': [0.1, 0.3, 0.5, 0.7, 0.9],
              'colsample_bytree': [0.1, 0.3, 0.5, 0.7, 0.9],
              'n_estimators': [500],
              'reg_alpha': [0.1, 0.3, 0.5, 0.7, 0.9],
              'reg_lambda': [0.1, 0.3, 0.5, 0.7, 0.9]
             }

model = LGBMClassifier()
lgbm_grid = RandomizedSearchCV(model,parameters, cv = 3, n_jobs = -1)
lgbm_grid.fit(  X_test, y_test.ravel())

print('Score: ', lgbm_grid.best_score_)
print('Params: ', lgbm_grid.best_params_)
```

```
Score:  0.89272030651341
Params:  {'subsample': 0.7, 'reg_lambda': 0.9, 'reg_alpha': 0.9, 'num_leaves': 9, 'n_estimator
s': 500, 'min_child_weight': 3, 'min_child_samples': 3, 'max_depth': 9, 'learning_rate': 0.03,
'colsample_bytree': 0.9}
```

Рисунок 3.21 – Результат моделі «LGBM»

У моделі LGBM, отримано трохи гірший результат, ніж інші, це все ще чудова модель з точністю 89,27%

Наступне використав модель градієнтного підсилення, тут скористались моделю GridSearch, щоб з'ясувати, які метрики найкраще використовувати в цій моделі (рис. 3.22).

```
from sklearn.ensemble import GradientBoostingClassifier

parameters = {'learning_rate': [0.01, 0.02, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5, 0.005],
              'n_estimators': [300, 500]
             }


model = GradientBoostingClassifier()
gridGradient = RandomizedSearchCV(model, parameters, cv = 5, n_jobs = -1)
gridGradient.fit(X_test, y_test)

print('Learning Rate: ', gridGradient.best_estimator_.learning_rate)
print('Score: ', gridGradient.best_score_)
```

```
Learning Rate:  0.01
Score:  0.9002177068214804
```

Рисунок 3.22 – Результат моделі «Gradient Boosting»

Об'єкт моделі GradientBoostingClassifier створюється з використанням заданого словника параметрів. Після цього створюється об'єкт RandomizedSearchCV, який використовується для перебору різних значень параметрів learning_rate та n_estimators. Об'єкт RandomizedSearchCV навчає модель на наборі даних X_train з різними значеннями параметрів та оцінює точність кожної моделі на наборі даних X_test.

Загальні результати для порівняння моделей на рисунку 3.23 – 3.24.

	Naive Bayes	Decision Tree	Random Forest	Extra Trees	KNN	Logistic Regression	AdaBoost	GradientBoosting	LGBM
Model	Naive Bayes	Decision Tree	Random Forest	Extra Trees	KNN	Logistic Regression	AdaBoost	GradientBoost	LGBM
Scaling	Normal Data	Normal Data	Normal Data	Normal Data	Normal	Normal Data	StandardScaler	StandardScaler	StandardScaler
Type	Gaussian	Gini	Gini	Gini	-	-	-	-	-
Precision	0.530973	0.893805	0.902655	0.884956	0.884956	0.911504	0.911504	0.920354	0.893805

Рисунок 3.23 – Результат усіх моделей

72]:

	Model	Train Accuracy	Test Accuracy
0	NaiveBayes	0.536398	0.530973
3	ExtraTrees	0.927203	0.884956
4	KNN	0.896552	0.884956
1	DecisionTree	0.927203	0.893805
8	LGBM	0.919540	0.893805
2	RandomForest	0.927203	0.902655
5	LogisticRegression	0.927203	0.911504
6	AdaBoost	0.900383	0.911504
7	GradientBoosting	0.927203	0.920354

Рисунок 3.24 – Кінцевий результат навчальних та тестових даних

Загалом, результати показують, що моделі машинного навчання можуть бути ефективними для прогнозування розладів сну. Однак, існують деякі важливі відмінності між моделями.

У результаті найкраща модель по точності виявилось у моделі Gradient Boosting, з точки зору точності - 92,03%.

Інші моделі, такі як логістична регресія, AdaBoost та інші, також показали непогані результати навчання, але їхня точність тестування нижча.

Модель Naive Bayes показала найнижчу точність навчання та тестування. Це може бути пов'язано з тим, що ця модель робить припущення про незалежність змінних, яке може не бути справедливим у даному випадку.

Наступні кроки, які можна зробити для покращення точності прогнозування, включають:

– Збільшення розміру набору даних. Більший розмір набору даних може забезпечити більш репрезентативне представлення різних типів розладів сну.

– Використання більш складних моделей машинного навчання. Більш складні моделі можуть краще враховувати складність взаємозв'язків між різними факторами, пов'язаними зі сном.

– Використання додаткових даних. Крім інформації про сон, можна використовувати додаткові дані, такі як медична історія, генетичні дані та інше.

3.4 Висновки

У третьому розділі магістерської кваліфікаційної роботи було проведено вивчення та оцінка використаних бібліотек у розробці інформаційної технології для аналізу та прогнозування розладу сну. Кожна з обраних бібліотек виявила високий рівень важливості та відповідність власним функціональним потребам в процесі обробки, аналізу та візуалізації даних, а також побудови моделей машинного навчання.

Бібліотека Pandas, як одна з ключових бібліотек, надала мощні інструменти для завантаження та обробки даних у вигляді таблиць, що спростило їхню маніпуляцію. NumPy забезпечив можливості для числових обчислень, що було необхідним для статистичного аналізу даних. Matplotlib та Seaborn надали інструменти для створення графічних візуалізацій, що дозволило краще розуміти взаємозв'язки між різними аспектами даних.

Scikit-learn виявився незамінним у реалізації моделей машинного навчання, надаючи широкий спектр алгоритмів для побудови моделей прогнозування. Yellowbrick та GridSearchCV/RandomizedSearchCV допомогли візуалізувати метрики та оцінювати продуктивність моделей, а також вибирати оптимальні параметри для моделей. LabelEncoder спростило обробку категоріальних даних.

Загальний аналіз бібліотек підтвердив їхню важливість у розробці системи аналізу та передбачення розладу сну. Використання цих бібліотек стало ключовим етапом у забезпеченні ефективного та точного аналізу даних, візуалізації результатів та побудові моделей, що важливо для розробки інформаційних систем у сферах, де необхідно аналізувати та передбачати на основі наявних даних, так як у випадку аналізу розладів сну.

Проведено докладний огляд та аналіз датасету, який містить інформацію про різні аспекти життя та здоров'я людини, пов'язані зі сном. Описані категоріальні та

числові змінні дозволили зрозуміти структуру та характеристики даних, що є ключовим етапом у будь-якому аналізі даних.

Категоріальні змінні, такі як стать, професія, категорія ІМТ та наявність розладу сну, надали унікальну інформацію про особливості популяції та можливі впливи на якість сну. Це стало важливою базою для подальшого аналізу впливу цих факторів на розлад сну. З числових змінних, таких як вік, тривалість сну, якість сну, рівень фізичної активності, рівень стресу, артеріальний тиск, частота серцевих скорочень та щоденна кількість кроків, отримано значимі числові дані, що дозволили виявити можливі кореляції та взаємозв'язки між ними.

Цей аналіз є ключовим для визначення факторів, які можуть впливати на якість сну, і він надав можливість відібрати найбільш важливі змінні для подальшої моделювання та передбачення розладу сну. Наприклад, з'ясувати, як тривалість сну, рівень стресу або фізична активність можуть корелювати з наявністю розладу сну у людини.

Важливим кроком у розумінні структури даних та ідентифікації ключових параметрів, які можуть впливати на розлад сну. Це забезпечило підґрунтя для подальших модельних розвідок та допомогло відібрати найбільш інформативні змінні для побудови моделей машинного навчання.

Проведено роботу із застосуванням різноманітних моделей для прогнозування розладу сну. Результати кожної моделі показали великий рівень значущості та впливу обраних алгоритмів на точність передбачення.

Модель Random Forest показала точність на рівні 90,26%. Модель LGBM, хоч і показала трохи меншу точність (89,27%), але теж може вважатись ефективною для передбачення розладу сну.

За результатами оцінки, модель Gradient Boosting виявилася найбільш точною серед усіх випробуваних моделей машинного навчання, з точністю прогнозування на рівні 92,03%. Це свідчить про її високу ефективність у роботі з наданими даними та здатність зробити більш точні прогнози щодо розладів сну у людей.

Отже, результати дослідження показали, що різні моделі машинного навчання можуть забезпечити значний рівень точності при передбаченні розладів сну. Однак, модель Gradient Boosting виявилася найефективнішою серед усіх, забезпечуючи найвищу точність у прогнозуванні розладів сну з використанням наданих у роботі даних.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія аналізу та передбачення розладу сну людини» 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 [32].

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	3	3	4
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	3	2	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5

Продовження таблиці 4.1

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
11. Практична здійсненність (термін реалізації)	3	4	5
12. Практична здійсненність (розробка документів)	4	5	4
Сума балів	37	40	41
Середньоарифметична сума балів $СБ_c$	39,3		

За результатами розрахунків, наведених в таблиці 4.1, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в [32].

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та передбачення розладу сну людини» становить 39,3 бала, що, відповідно до [32], свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [33] :

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

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

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

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

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
Кількість використаних моделей машинного навчання	одиниць	5	9	1,8	0,35
Попередня обробка та очистка даних	бал	6	9	1,5	0,1
Точність прогнозу	%	80	92	1,15	0,15
Кількість графіків розвідувального аналізу	одиниць	10	14	1,4	0,25
Алгоритми нормалізації даних	бал	7	9	1,28	0,15

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

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1{,}8 \cdot 0{,}35 + 1{,}5 \cdot 0{,}1 + 1{,}15 \cdot 0{,}15 + 1{,}4 \cdot 0{,}25 + 1{,}28 \cdot 0{,}15 = 1{,}49.$$

Отже, за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,49 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія аналізу та передбачення розладу сну людини», 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 [32]:

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

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.

$$З_о = 16420{,}00 \cdot 42 / 21 = 32840{,}00 \text{ (грн)}.$$

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 НДР з розробки іформаційної технології аналізу та передбачення розладу сну людини	16420,00	781,90	42	32840,00

Продовження таблиці 4.3

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Інженер-розробник програмного забезпечення	16090,00	766,19	38	29115,24
Лікар-терапевт (сомнолог) вищої категорії	14000,00	666,67	5	3333,33
Провідний фахівець	7850,00	373,81	38	14204,76
Інженер-аналітик (аналіз)	16100,00	766,67	7	5366,67
Всього				84860,00

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія аналізу та передбачення розладу сну людини» розраховуємо за формулою:

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

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:

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

де M_M – розмір мінімальної місячної заробітної плати, приймемо $Мм$=6700,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення [32];

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.

$$C_1 = 6700{,}00 \cdot 1{,}10 \cdot 1{,}35 / (21 \cdot 8) = 59{,}22 \text{ (грн)}.$$

$$З_{р1} = 59{,}22 \cdot 6{,}53 = 386{,}73 \text{ (грн)}.$$

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
Монтаж обчислювального обладнання для забезпечення розробки і проведення аналізу	6,53	2	1,10	59,22	386,73
Підготовка робочого місця дослідника-розробника інформаційної технології	7,57	2	1,35	72,68	550,21
Інсталяція програмного забезпечення розробки (моделювання) інформаційної технології аналізу	5,55	4	1,70	91,53	507,97

Продовження таблиці 4.4

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Введення кодів блоку формування бази даних системи аналізу	7,25	5	1,70	91,53	663,57
Введення кодів блоку передбачення розладів	4,58	5	1,10	59,22	271,24
Монтаж дослідного обладнання	16,25	5	1,70	91,53	1487,31
Налагодження системної взаємодії компонентів	8,03	4	1,50	80,76	648,49
Інтеграція модулів аналізу та передбачення розладів сну	9,55	5	1,70	91,53	874,08
Всього					5389,61

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 11%.

$$З_{дод} = (84860{,}00 + 5389{,}61) \cdot 11 / 100\% = 9927{,}46 \text{ (грн)}.$$

4.3.2 Відрахування на соціальні заходи

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

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

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

Зн = (84860,00 + 5389,61 + 9927,46) · 22 / 100% = 22038,95 (грн).

4.3.3 Сировина та матеріали

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

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

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.

$М_1$ = 15,000 · 32,00 · 1,01 - 0,0 · 0,0 = 484,80 (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
Папір для плотера А1	32,00	15,000	0,0	0,0	484,80
FLASH-пам’ять Traveler 128 GB	329,00	1,000	0,0	0,0	332,29
Канцелярські товари	210,00	4,000	0,0	0,0	848,40
Картридж змінний Canon laser	1032,00	2,000	0,0	0,0	2084,64
Органайзер офісний Skiper DATA	196,00	4,000	0,0	0,0	791,84
Папір офісний Skiper 80 (500) А4	184,00	4,000	0,0	0,0	743,36
Папка для паперів Skiper F4-500	85,00	4,000	0,0	0,0	343,40
Всього					5628,73

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія аналізу та передбачення розладу сну людини» відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \quad (4.8)$$

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

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

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

k – кількість найменувань устаткування.

$$В_{спец} = 46599,00 \cdot 1 \cdot 1,01 = 47064,99 \text{ (грн)}.$$

Отримані результати зведемо до таблиці 4.6.

Таблиця 4.6 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Серверне обладнання обробки та збереження DATA BASE	1	46599,00	47064,99
Маршрутизатор	1	3259,00	3291,59
Всього			50356,58

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \quad (4.9)$$

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

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

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

k – кількість найменувань програмних засобів.

$B_{прг} = 6820,00 \cdot 1 \cdot 1,01 = 6888,20$ (грн).

Отримані результати зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Середовище програмування та бібліотеки	1	6820,00	6888,20
Kaggle Dataset „Sleep health and lifestyle”	1	9860,00	9958,60
Доступ до мережі Internet (високошвидкісний) грн/місяць	2	229,00	462,58
Всього			17309,38

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:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.10)$$

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.

$$А_{обл} = (24899{,}00 \cdot 2) / (3 \cdot 12) = 1383{,}28 \ (грн).$$

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника інформаційної технології аналізу	24899,00	3	2	1383,28
Електронно-обчислювальний центр системи аналізу (Системний блок 2E Complex Gaming (2E-10204)	31699,00	3	2	1761,06

Продовження таблиці 4.8

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Робоче місце розробника інформаційної технології	8599,00	5	2	286,63
Пристрої передачі даних	7499,00	4	2	312,46
Пристрої виведення інформації	6599,00	5	2	219,97
Блоки зовнішньої памяті (зберігання бази даних)	7250,00	4	2	302,08
Програмне забезпечення Microsoft Windows, Offise	9469,00	3	2	526,06
Лабораторія досліджень	425000,00	20	2	3541,67
Всього				8333,20

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (4.11)$$

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; приймемо Ц$_e$ = 7,50 (грн);

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

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

$$B_e = 0,32 \cdot 320,0 \cdot 7,50 \cdot 0,95 / 0,97 = 768,00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.9.

Таблиця 4.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника ІТ аналізу	0,32	320,0	768,00
Електронно-обчислювальний центр системи аналізу (Системний блок 2E Complex Gaming (2E-10204)	0,42	320,0	1008,00
Робоче місце розробника інформаційної технології	0,10	320,0	240,00
Пристрої передачі даних	0,04	320,0	96,00
Пристрої виведення інформації	0,16	8,0	9,60

Продовження таблиці 4.9

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Блоки зовнішньої памяті (зберігання бази даних)	0,03	320,0	72,00
Серверне обладнання обробки та збереження DATA BASE	0,36	320,0	864,00
Маршрутизатор	0,04	200,0	60,00
Всього			3117,60

4.3.9 Службові відрядження

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

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (4.12)$$

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

$$В_{св} = (84860{,}00 + 5389{,}61) \cdot 20 / 100\% = 18049{,}92 \text{ (грн)}.$$

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:

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

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

$$В_{сп} = (84860,00 + 5389,61) \cdot 30 / 100\% = 27074,88 \text{ (грн)}.$$

4.3.11 Інші витрати

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

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.14)$$

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

$$І_в = (84860,00 + 5389,61) \cdot 55 / 100\% = 49637,28 \text{ (грн)}.$$

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:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.15)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв} = 110\%$.

$$В_{нзв} = (84860,00 + 5389,61) \cdot 110 / 100\% = 99274,57 \text{ (грн)}.$$

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

$$В_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (4.16)$$

$В_{заг}$= 84860,00 +5389,61 +9927,46 +22038,95 +5628,73 +0,00 +50356,58 + 17309,38 + 8333,20 +3117,60 +18049,92 +27074,88 +49637,28 +99274,57 = 400998,17 (грн).

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

$$ЗВ = \frac{В_{заг}}{\eta}, \quad (4.17)$$

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

ЗВ = 400998,17 / 0,9 = 445553,52 (грн).

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

Результати дослідження проведені за темою «Інформаційна технологія аналізу та передбачення розладу сну людини» 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;

Показники за 4 роки зі збільшення кількості споживачів або осіб має наступні цифри. За перший рік – 750 споживачів, за другий рік – 900 споживачів, за третій рік – 900, і за четвертий рік – 500 осіб.

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 7250 осіб;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 12500,00 (грн);

$\pm\Delta Ц_о$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 681,25 (грн).

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

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.18)$$

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

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

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

Збільшення чистого прибутку 1-го року:

$$\Delta П_1 = (681{,}25 \cdot 7250{,}00 + 13181{,}25 \cdot 750) \cdot 0{,}83 \cdot 0{,}4 \cdot (1 - 0{,}18/100\%) = 4035958{,}00 \text{ (грн)}.$$

Збільшення чистого прибутку 2-го року:

$$\Delta П_2 = (681{,}25 \cdot 7250{,}00 + 13181{,}25 \cdot 1650) \cdot 0{,}83 \cdot 0{,}4 \cdot (1 - 0{,}18/100\%) = 7265575{,}15 \text{ (грн)}.$$

Збільшення чистого прибутку 3-го року:

$$\Delta П_3 = (681{,}25 \cdot 7250{,}00 + 13181{,}25 \cdot 2550) \cdot 0{,}83 \cdot 0{,}4 \cdot (1 - 0{,}18/100\%) = 10495192{,}30 \text{ (грн)}.$$

Збільшення чистого прибутку 4-го року:

$$\Delta П_4 = (681{,}25 \cdot 7250{,}00 + 13181{,}25 \cdot 3050) \cdot 0{,}83 \cdot 0{,}4 \cdot (1 - 0{,}18/100\%) = 12289424{,}05 \text{ (грн)}.$$

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

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.19)$$

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

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

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

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

$$ПП= 4035958,00/(1+0,2)^1+7265575,15/(1+0,2)^2+10495192,30/(1+0,2)^3+$$
$$+12289424,05/(1+0,2)^4=3363298,33+5045538,30+6073606,66+5926612,68=$$
$$=20409055,97 \text{ (грн)}.$$

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

$$PV = k_{інв} \cdot ЗВ, \quad (4.20)$$

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 445553,52 (грн).

$$PV = k_{інв} \cdot ЗВ = 2 \cdot 445553,52 = 891107,03 \text{ (грн)}.$$

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

$$E_{абс} = ПП - PV \quad (4.21)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 20409055,97 (грн);

PV – теперішня вартість початкових інвестицій, 891107,03 (грн).

$$E_{абс} = ПП - PV = 20409055,97 - 891107,03 = 19517948,93 \text{ (грн)}.$$

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

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.22)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 19517948,93 (грн);

PV – теперішня вартість початкових інвестицій, 891107,03 (грн);

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

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+19517948,93/891107,03)^{1/4} = 1,19.$$

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

$$\tau_{мін} = d + f, \quad (4.23)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,32.

$\tau_{мін}$= 0,1+0,32 = 0,42 < 1,19 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія аналізу та передбачення розладу сну людини» доцільно.

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

$$T_{ок} = \frac{1}{E_в}, \quad (4.24)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$= 1 / 1,19 = 0,84 року.

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

4.5 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та передбачення розладу сну людини» становить 39,3 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 1,49 рази.

Також термін окупності становить 0,84 року, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія аналізу та передбачення розладу сну людини».

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробленню інформаційної технології аналізу та передбаченю розладу сну людини, що є важливим аспектом у вивченні фізіологічних та психологічних процесів людського організму. Використання технологій машинного навчання та аналізу даних відкриває нові можливості для точного прогнозування розладів сну, що в свою чергу дозволяє удосконалити методи лікування та підтримки здоров'я у сфері сну та психофізіології людини.

Виявлено, що інтеграція різних технологій може значно підвищити ефективність виявлення та передбачення розладів сну. Порівняно та оглято існуючі аналоги продемонстрували різноманітність підходів та технологій для аналізу сну, підкресливши необхідність подальших досліджень для вдосконалення методів аналізу та підвищення доступності інструментів для прогнозування розладів сну.

Також сконцентровано увагу на виборі ключових інструментів машинного навчання, зокрема платформи Kaggle та мови програмування Python. Ці інструменти обрані через їхню доступність, розширені можливості та спрощення у використанні. Сам аналіз різних моделей машинного навчання виявив переваги та обмеження кожної з них, але підкреслив, що при правильному використанні вони можуть бути використані для різних завдань аналізу даних та передбачення.

Вивчено та оцінено ключові бібліотеки для аналізу та передбачення розладу сну, які надали необхідні інструменти для обробки даних, візуалізації результатів та побудови моделей машинного навчання. Детальний аналіз датасету різних аспектів життя та здоров'я людини, пов'язаних із сном, дозволив виділити важливі змінні, що впливають на розлад сну, та вибрати найбільш інформативні для подальшого моделювання.

У цілому, застосування різних моделей машинного навчання продемонструвало високий рівень точності в передбаченні розладу сну. Зокрема, модель Gradient Boosting здобула найвищу точність на рівні 92,03%, виявивши

найбільшу ефективність у прогнозуванні розладу сну на основі наданих даних порівняно з іншими моделями.

Також виконано дослідження економічної частини роботи, виявлено високий комерційний потенціал розробки за темою "Інформаційна технологія аналізу та передбачення розладу сну людини" на рівні 39,3 бала, що перевищує середній рівень та свідчить про важливість проведення подальших досліджень. Розрахований термін окупності становить 0,84 роки, що менше за 3 роки, підкреслюючи комерційну привабливість розробки та її можливу здатність привернути інвесторів для подальшої фінансової підтримки та можливе виведення на ринок.

Підтверджено, що застосування технологій машинного навчання та аналізу даних може забезпечити високий рівень точності в передбаченні розладів сну. Проте, модель Gradient Boosting виявилася найефективнішою серед усіх випробуваних моделей, забезпечивши найвищу точність у прогнозуванні розладів сну на основі наданих даних. Ці результати дозволяють використовувати цю інформаційну систмему та є ключовим елементом у розробці ефективних систем аналізу та прогнозування розладів сну, що має важливе значення для медичної діагностики та поліпшення якості життя людей.

Отже, можна зробити висновок про успішне розроблення інформаційної технології аналізу та передбачення розладу сну людини.

За даною темою дослідження зроблено доповідь на тему «Інформаційна технологія аналізу та передбачення розладу сну людини» на LIII Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації ВНТУ з публікацією тез (м. Вінниця, 2023-2024 рр.) [1].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іщук Д.В., Жуков С.О. Інформаційна технологія аналізу та передбачення розладу сну людини. *LIII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2023-2024)*. Вінниця, 2023-2024. URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/19789/16383

2. Faramawy, S., Zaki, T., & Sakr, A. E. Sleep prediction, and processing: A review. *Multiscale entropy analysis of single lead ECG derived for AI based prediction of sleep apnea events*. 2018. Т. 34. С. 34-54. URL: https://www.sciencedirect.com/science (дата звернення 12.09.2023)

3. Devine, J. K., Nichols, M. G., Schwartz, L. P., Choynowski, J., & Hursh, S. R. (2023). Biomathematical modeling for the prediction of sleep behavior and comparison against cognitive performance in firefighters. *Safety science*, *163*. URL: https://www.sciencedirect.com/science (дата звернення 12.09.2023)

4. Zhai, Qian, et al. "Machine Learning-Enabled Noncontact Sleep Structure Prediction." *Advanced Intelligent Systems* 4.5 (2022): 2100227.Verhun V. R. Характеристика методів розв'язання задачі класифікації в інтелектуальному аналізі даних навчальних програм. Scientific Bulletin of UNFU. 2019. Т. 29, № 6. С. 136–139.

5. Cabaneros S. M., Calautit J. K., Hughes B. R. A review of artificial neural network models for ambient air pollution prediction. *Environmental Modelling & Software*. 2019. Т. 119. С. 285-304. URL: https://doi.org/10.1016/j.envsoft.2019.06.014 (дата звернення 14.09.2023)

6. Sarker I. H. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*. 2021. Т. 2. №. 3. С. 160. URL: https://link.springer.com/article/10.1007/s42979-021-00592-x (дата звернення 15.09.2023)

7. Hopkins E. et al. Machine learning tools, algorithms, and techniques. *Journal of Self-Governance and Management Economics*. 2022. Т. 10. №. 1. С. 43-55.

URL: https://www.ceeol.com/search/article-detail?id=1030033 (дата звернення 15.09.2023)

8. Yin, Jiaju, Jiandong Xu, and Tian-Ling Ren. "Recent Progress in Long-Term Sleep Monitoring Technology." *Biosensors* 13.3 (2023): 395. URL: https://www.mdpi.com/2079-6374/13/3/395

9. Pappalardo, G., Cafiso, S., Di Graziano, A., & Severino, A. (2021). Decision tree method to analyze the performance of lane support systems. *Sustainability*, *13*(2), 846. URL: https://www.mdpi.com/2071-1050/13/2/846

10. Шостя С. П. Машинне навчання. *Матеріали VII Всеукраїнської науково-практичної Інтернет-конференції молодих учених та студентів Полтава, 24-25 листопада 2021 р.* Полтава С. 146. URL: http://elcat.pnpu.edu.ua/docs/IICTE_2021.pdf#page=148 (дата звернення: 01.10.2023)

11. Sultonov S. IMPORTANCE OF PYTHON PROGRAMMING LANGUAGE IN MACHINE LEARNING. *International Bulletin of Engineering and Technology*. 2023. Т. 3. №. 9. С. 28-30. URL: https://internationalbulletins.com/intjour/index.php/ibet/article/view/1020/658# (дата звернення 02.10.2023)

12. Siegmund N., Ruckel N., Siegmund J. Dimensions of software configuration: on the configuration context in modern software development. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2020. С. 338-349. URL: https://doi.org/10.1145/3368089.3409675 (дата звернення 02.10.2023)

13. Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20-28. URL: https://jastt.org/index.php/jasttpath/article/view/65/24

14. Janiesch C., Zschech P., Heinrich K. Machine learning and deep learning. *Electronic Markets*. 2021. Т. 31. №. 3. С. 685-695. URL: https://link.springer.com/article/10.1007/s12525-021-00475-2 (дата звернення 04.10.2023)

15. Sagheer A., Kotb M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*. 2019. Т. 323. С. 203-213. URL: https://doi.org/10.1016/j.neucom.2018.09.082 (дата звернення 08.10.2023)

16. Hao J., Ho T. K. Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*. 2019. Т. 44. №. 3. С. 348-361. URL: https://doi.org/10.3102/1076998619832248 (дата звернення 09.10.2023)

17. Morgenthaler S. Exploratory data analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2009. Т. 1. №. 1. С. 33-44. URL: https://doi.org/10.1002/wics.2 (дата звернення 15.10.2023)

18. Goloborodko A. A. et al. Pyteomics—a Python framework for exploratory data analysis and rapid software prototyping in proteomics. *Journal of The American Society for Mass Spectrometry*. 2013. Т. 24. №. 2. С. 301-304. URL: https://doi.org/10.1007/s13361-012-0516-6 (дата звернення 18.10.2023)

19. Raschka S., Patterson J., Nolet C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*. 2020. Т. 11. №. 4. С. 193. URL: https://doi.org/10.3390/info11040193 (дата звернення 02.11.2023)

20. Yamak P. T., Yujian L., Gadosey P. K. A comparison between arima, lstm, and gru for time series forecasting. *Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence*. 2019. С. 49-55. URL: https://doi.org/10.1145/3377713.3377722 (дата звернення 03.11.2023)

21. DiPietro R., Hager G. D. Deep learning: RNNs and LSTM. *Handbook of medical image computing and computer assisted intervention.* Academic Press. 2020.

С. 503-519. URL: https://doi.org/10.1016/B978-0-12-816176-0.00026-0 (дата звернення 03.11.2023)

22. Janiesch C., Zschech P., Heinrich K. Machine learning and deep learning. *Electronic Markets*. 2021. Т. 31. №. 3. С. 685-695. URL: https://link.springer.com/article/10.1007/s12525-021-00475-2 (дата звернення 04.10.2023)

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

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

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

26. Introduction to XGBoost Algorithm in Machine Learning, 2023. URL: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/ (дата звернення 24.09.2023)

27. GLOB Module: File Searching In Python, 2022. URL: https://pythonology.eu/glob-module-file-searching-in-python/ (дата звернення 24.09.2023)

28. Working with warnings in Python, 2020. URL: https://lerner.co.il/2020/04/27/working-with-warnings-in-python/ (дата звернення 25.09.2023)

29. Сайт Kaggle. URL: https://www.kaggle.com/ (дата звернення 03.10.2023)

30. Pedro Domingos. "The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World." Basic Books, 2015.

31. Sebastian Raschka and Vahid Mirjalili. "Python Machine Learning." Packt Publishing, 2017.

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

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

Додаток А

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

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
____ д.т.н., проф. Віталій МОКІН
«__» _____2023 року

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ РОЗЛАДУ СНУ ЛЮДИНИ
08-34.МКР.002.00.000 ТЗ

Керівник: к.т.н., доц. каф. САІТ
_____ Сергій ЖУКОВ
«__» _____ 2023 р.
Розробив ст. гр. 2ІСТ-22м
_____Дмитро ІЩУК
«__» _____ 2023 р.

Вінниця 2023

1. Підстава для проведення робіт.
Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2023р., та індивідуальне завдання на МКР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2023р.
2. Джерела розробки:
1) 1) Kaggle Dataset «Sleep health and lifestyle 2023». URL: https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset;
2) Sarker I. H. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*. 2021. Т. 2. №. 3. С. 160.
3. Мета і призначення роботи.
Метою дослідження є аналіз та підвищення точності передбачення розладу сну людини.
4. Вихідні дані для проведення робіт.
Ishchuk Dmytro Kaggle Dataset « Sleep health and lifestyle 2023». URL: https://www.kaggle.com/dimaishchuk/sleep-health-eda-and-machine-learning
5. Методи дослідження.
В даній роботі використовуються методи машинного навчання для передбачення розладу сну людини.
6. Етапи роботи і терміни їх виконання:
а) Аналіз предметної області _____ – _____
б) Вибір оптимальних моделей _____ – _____
в) Розроблення інформаційної технології аналізу та передбачення розладу сну у людини. _____ – _____
г) Економічна частина _____ – _____
д) Оформлення матеріалів до захисту МКР _____ – _____
7. Очікувані результати та порядок реалізації.
Отримати інформаційну технологію аналізу та передбачення розладу сну людини.
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 «___» _____ 2023 р.
Початок розробки «___» _____ 2023 р.
Граничні терміни виконання МКР «___» _____ 2023 р.

Розробив студент групи 2ІСТ-22м _____ Дмитро ІЩУК

Додаток Б

Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія аналізу та передбачення розладу сну людини»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 96% Схожість 4%

Аналіз звіту подібності (відмітити потрібне)

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

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

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

Автор роботи _____ (підпис) Дмитро ЩУК

Керівник роботи _____ (підпис) Сергій ЖУКОВ

Додаток В

Лістинг програми

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from yellowbrick.classifier import ConfusionMatrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('/kaggle/input/sleep-health-and-lifestyle-dataset/Sleep_health_and_lifestyle_dataset.csv', sep = ',')
df.head(5)
df.info()
df.describe()
df.dtypes
corr = df.corr().round(2)
plt.figure(figsize = (25,20))
sns.heatmap(corr, annot = True, cmap = 'YlOrBr')
plt.figure(figsize = (20, 10))
plt.subplot(2, 2, 1)
plt.gca().set_title('Variable Gender')
sns.countplot(x = 'Gender', palette = 'Set2', data = df)
plt.subplot(2, 2, 2)
plt.gca().set_title('Variable BMI Category')
sns.countplot(x = 'BMI Category', palette = 'Set2', data = df)
plt.subplot(2, 2, 3)
plt.gca().set_title('Variable Sleep Disorder')
sns.countplot(x = 'Sleep Disorder', palette = 'Set2', data = df)
plt.figure(figsize = (20, 10))
plt.subplot(2, 1, 1)
plt.gca().set_title('Variable Occupation')
sns.countplot(x = 'Occupation', palette = 'Set2', data = df)
plt.subplot(2, 1, 2)
plt.gca().set_title('Variable Blood Pressure')
sns.countplot(x = 'Blood Pressure', palette = 'Set2', data = df)
```

```
plt.figure(figsize = (15, 18))
plt.subplot(4, 2, 1)
sns.histplot(x = df['Age'], kde = False)
plt.subplot(4, 2, 2)
sns.histplot(x = df['Sleep Duration'], kde = False)
plt.subplot(4, 2, 3)
sns.histplot(x = df['Quality of Sleep'], kde = False)
plt.subplot(4, 2, 4)
sns.histplot(x = df['Physical Activity Level'], kde = False)
plt.subplot(4, 2, 5)
sns.histplot(x = df['Stress Level'], kde = False)
plt.subplot(4, 2, 6)
sns.histplot(x = df['Heart Rate'], kde = False)
plt.subplot(4, 2, 7)
sns.histplot(x = df['Daily Steps'], kde = False)
plt.title("Boxplot Age", fontdict = {'fontsize': 20})
sns.boxplot(x=df["Age"])
plt.title("Boxplot Sleep Duration", fontdict = {'fontsize': 20})
sns.boxplot(x=df["Sleep Duration"])
plt.title("Boxplot Quality of Sleep", fontdict = {'fontsize': 20})
sns.boxplot(x=df["Quality of Sleep"])
plt.title("Boxplot Physical Activity Level", fontdict = {'fontsize': 20})
sns.boxplot(x=df["Physical Activity Level"])
<Axes: title={'center': 'Boxplot Stress Level'}, xlabel='Stress Level'>
plt.title("Boxplot Heart Rate", fontdict = {'fontsize': 20})
sns.boxplot(x=df["Heart Rate"])
plt.title("Boxplot Daily Steps", fontdict = {'fontsize': 20})
sns.boxplot(x=df["Daily Steps"])
plt.figure(figsize = (20, 10))
plt.suptitle("Analysis Of Variable Sleep Disorder",fontweight="bold", fontsize=20)
plt.subplot(2, 1, 1)
plt.gca().set_title('Variable Gender')
sns.countplot(x = 'Gender', hue = 'Sleep Disorder', palette = 'Set2', data = df)
plt.subplot(2, 1, 2)
plt.gca().set_title('Variable BMI Category')
sns.countplot(x = 'BMI Category', hue = 'Sleep Disorder', palette = 'Set2', data = df)
plt.figure(figsize = (20, 10))
plt.subplot(2, 1, 1)
plt.gca().set_title('Variable Occupation')
sns.countplot(x = 'Occupation', hue = 'Sleep Disorder', palette = 'Set2', data = df)
plt.subplot(2, 1, 2)
plt.gca().set_title('Variable Blood Pressure')
```

```
sns.countplot(x = 'Blood Pressure', hue = 'Sleep Disorder', palette = 'Set2', data = df)
plt.figure(figsize = (25, 20))
plt.suptitle("Analysis Of Variable Sleep Disorder",fontweight="bold", fontsize=20)
plt.subplot(4,2,1)
sns.boxplot(x="Sleep Disorder", y="Age", data=df)
plt.subplot(4,2,2)
sns.boxplot(x="Sleep Disorder", y="Sleep Duration", data=df)
plt.subplot(4,2,3)
sns.boxplot(x="Sleep Disorder", y="Quality of Sleep", data=df)
plt.subplot(4,2,4)
sns.boxplot(x="Sleep Disorder", y="Physical Activity Level", data=df)
plt.subplot(4,2,5)
sns.boxplot(x="Sleep Disorder", y="Stress Level", data=df)
plt.subplot(4,2,6)
sns.boxplot(x="Sleep Disorder", y="Heart Rate", data=df)
plt.subplot(4,2,7)
sns.boxplot(x="Sleep Disorder", y="Daily Steps", data=df)
df = df.drop('Person ID', axis = 1)
hot = pd.get_dummies(df[['Gender', 'Occupation', 'BMI Category', 'Blood Pressure']])
df = pd.concat([df, hot], axis = 1)
df = df.drop(['Gender', 'Occupation', 'BMI Category', 'Blood Pressure'], axis = 1)
X = df.drop('Sleep Disorder', axis = 1)
X = X.values
y = df['Sleep Disorder']
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_standard = scaler.fit_transform(X)
from sklearn.model_selection import train_test_split
X_train,X_test, y_train, y_test = train_test_split(X_standard, y, test_size = 0.3, random_state = 0)
from sklearn.naive_bayes import GaussianNB
naive_bayes = GaussianNB()
naive_bayes.fit(X_train, y_train)
previsoes = naive_bayes.predict(X_test)
cm = ConfusionMatrix(naive_bayes)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
classification_naive_gaussian = (classification_report(y_test, previsoes))
print(classification_naive_gaussian)
score_naive_gaussian = 0.5309734513274337
parameters = {'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['entropy', 'gini']
```

```
            }
model = DecisionTreeClassifier()
gridDecisionTree = RandomizedSearchCV(model, parameters, cv = 3, n_jobs = -1)
gridDecisionTree.fit(X_train, y_train)
print('Mín Split: ', gridDecisionTree.best_estimator_.min_samples_split)
print('Max Nvl: ', gridDecisionTree.best_estimator_.max_depth)
print('Algorithm: ', gridDecisionTree.best_estimator_.criterion)
print('Score: ', gridDecisionTree.best_score_)
decision_tree = DecisionTreeClassifier(criterion = 'entropy', min_samples_split = 6, max_depth= 6, random_state=0)
decision_tree.fit(X_train, y_train)
previsoes = decision_tree.predict(X_test)
cm = ConfusionMatrix(decision_tree)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
classification_decision = (classification_report(y_test, previsoes))
print(classification_decision)
score_tree = 0.8938053097345132
columns = df.drop('Sleep Disorder', axis = 1).columns
feature_imp = pd.Series(decision_tree.feature_importances_, index = columns).sort_values(ascending = False)
feature_imp
from sklearn.ensemble import RandomForestClassifier
parameters = {'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['entropy', 'gini']
             }
model = RandomForestClassifier()
gridRandomForest = RandomizedSearchCV(model, parameters, cv = 5, n_jobs = -1)
gridRandomForest.fit(X_train, y_train)
print('Algorithm: ', gridRandomForest.best_estimator_.criterion)
print('Score: ', gridRandomForest.best_score_)
print('Mín Split: ', gridRandomForest.best_estimator_.min_samples_split)
print('Max Nvl: ', gridRandomForest.best_estimator_.max_depth)
random_forest = RandomForestClassifier(n_estimators = 100, min_samples_split = 5, max_depth= 5,  criterion = 'gini',
random_state = 0)
random_forest.fit(X_train, y_train)
previsoes = random_forest.predict(X_test)
cm = ConfusionMatrix(random_forest)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
classification_random = (classification_report(y_test, previsoes))
print(classification_random)
score_random = 0.9026548672566371
```

```
feature_imp_random = pd.Series(random_forest.feature_importances_, index = columns).sort_values(ascending = False)
feature_imp_random
from sklearn.ensemble import ExtraTreesClassifier
parameters = {'max_depth': [3, 4, 5, 6, 7, 9, 11],
              'min_samples_split': [2, 3, 4, 5, 6, 7],
              'criterion': ['entropy', 'gini']
             }
model = ExtraTreesClassifier()
gridExtraTrees = RandomizedSearchCV(model, parameters, cv = 3, n_jobs = -1)
gridExtraTrees.fit(X_train, y_train)
print('Algorithm: ', gridExtraTrees.best_estimator_.criterion)
print('Score: ', gridExtraTrees.best_score_)
print('Mín Split: ', gridExtraTrees.best_estimator_.min_samples_split)
print('Max Nvl: ', gridExtraTrees.best_estimator_.max_depth)
extra_trees = ExtraTreesClassifier(n_estimators = 100, min_samples_split = 6, max_depth= 7, criterion = 'gini', random_state
= 0)
extra_trees.fit(X_train, y_train)
previsoes = extra_trees.predict(X_test)
cm = ConfusionMatrix(extra_trees)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
classification_extra = (classification_report(y_test, previsoes))
print(classification_extra)
score_extra = 0.8849557522123894
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
k_list = list(range(1,10))
k_values = dict(n_neighbors = k_list)
grid = GridSearchCV(knn, k_values, cv = 2, scoring = 'accuracy', n_jobs = -1)
grid.fit(X_train, y_train)
grid.best_params_, grid.best_score_
knn = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
knn.fit(X_train, y_train)
previsoes = knn.predict(X_test)
cm = ConfusionMatrix(knn)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
classification_knn = (classification_report(y_test, previsoes))
print(classification_knn)
score_knn = 0.8849557522123894
import joblib
joblib.dump(classification_knn, 'sleep_health.pkl')
```

```
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score, learning_curve
import matplotlib.pyplot as plt
def plot_learning_curves(model, X_train, y_train):
    train_sizes, train_scores, val_scores = learning_curve(
        model, X_train, y_train, cv=5, train_sizes=np.linspace(0.1, 1.0, 10)
    )
    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    val_scores_mean = np.mean(val_scores, axis=1)
    val_scores_std = np.std(val_scores, axis=1)
    plt.figure()
    plt.title("Courbes d'apprentissage")
    plt.xlabel("Taille de l'ensemble d'entraînement")
    plt.ylabel("Score")
    plt.grid()
    plt.fill_between(
        train_sizes,
        train_scores_mean - train_scores_std,
        train_scores_mean + train_scores_std,
        alpha=0.1,
        color="r",
    )
    plt.fill_between(
        train_sizes,
        val_scores_mean - val_scores_std,
        val_scores_mean + val_scores_std,
        alpha=0.1,
        color="g",
    )
    plt.plot(
        train_sizes,
        train_scores_mean,
        "o-",
        color="r",
        label="Score d'entraînement",
    )
    plt.plot(
        train_sizes,
        val_scores_mean,
        "o-",
        color="g",
```

```
        label="Score de validation",
    )
    plt.legend(loc="best")
    plt.show()
model.fit(X_train, y_train)
plot_learning_curves(model, X_train, y_train)
cv_score = cross_val_score(model, X_test, y_test, cv=5).mean()
print(f"Score de validation croisée : {cv_score:.2f}")
test_score = model.score(X_test, y_test)
print(f"Score de test : {test_score:.2f}")
from sklearn.linear_model import LogisticRegression
logistic = LogisticRegression(random_state = 1, max_iter=10000)
logistic.fit(X_train, y_train)
previsoes = logistic.predict(X_test)
cm = ConfusionMatrix(logistic)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
logistic_normal = (classification_report(y_test, previsoes))
print(logistic_normal)
logistic_normal = 0.911504424778761
from sklearn.ensemble import AdaBoostClassifier
parameters = {'learning_rate': [0.01, 0.02, 0.05, 0.07, 0.09, 0.1, 0.3, 0.001, 0.005],
              'n_estimators': [300, 500]
             }
model = AdaBoostClassifier()
gridAdaBoost = RandomizedSearchCV(model, parameters, cv = 2, n_jobs = -1)
gridAdaBoost.fit(X_train, y_train)
print('Learning Rate: ', gridAdaBoost.best_estimator_.learning_rate)
print('Score: ', gridAdaBoost.best_score_)
ada_boost = AdaBoostClassifier(n_estimators = 500, learning_rate =  0.02, random_state = 0)
ada_boost.fit(X_train, y_train)
previsoes = ada_boost.predict(X_test)
cm = ConfusionMatrix(ada_boost)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
classification_ada_scaler = (classification_report(y_test, previsoes))
print(classification_ada_scaler)
score_ada_scaler = 0.911504424778761
from sklearn.ensemble import GradientBoostingClassifier
parameters = {'learning_rate': [0.01, 0.02, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5, 0.005],
              'n_estimators': [300, 500]
             }
```

```
model = GradientBoostingClassifier()
gridGradient = RandomizedSearchCV(model, parameters, cv = 5, n_jobs = -1)
gridGradient.fit(X_train, y_train)
print('Learning Rate: ', gridGradient.best_estimator_.learning_rate)
print('Score: ', gridGradient.best_score_)
grad_boost = GradientBoostingClassifier(n_estimators = 300, learning_rate = 0.01, random_state = 0)
grad_boost.fit(X_test, y_test)
previsoes = grad_boost.predict(X_test)
cm = ConfusionMatrix(grad_boost)
cm.fit(X_test, y_test)
cm.score(X_test, y_test)
classification_grad_standard = (classification_report(y_test, previsoes))
print(classification_grad_standard)
from lightgbm import LGBMClassifier
parameters = {'learning_rate': [0.01, 0.03, 0.05, 0.1, 0.3, 0.5, 0.8],
              'max_depth': [1 , 2, 3, 4, 5, 6, 7, 8, 9],
              'min_child_weight': [1, 3, 5, 7, 9],
              'min_child_samples': [1, 3, 5, 7, 9, 15],
              'num_leaves': [3, 5, 7, 9],
              'subsample': [0.1, 0.3, 0.5, 0.7, 0.9],
              'colsample_bytree': [0.1, 0.3, 0.5, 0.7, 0.9],
              'n_estimators': [500],
              'reg_alpha': [0.1, 0.3, 0.5, 0.7, 0.9],
              'reg_lambda': [0.1, 0.3, 0.5, 0.7, 0.9]
             }
model = LGBMClassifier()
lgbm_grid = RandomizedSearchCV(model,parameters, cv = 3, n_jobs = -1)
lgbm_grid.fit(X_train, y_train.ravel())
print('Score: ', lgbm_grid.best_score_)
print('Params: ', lgbm_grid.best_params_)
lgbm = LGBMClassifier(subsample = 0.7, reg_lambda = 0.9, reg_alpha = 0.9, num_leaves = 9, n_estimators = 500,
min_child_weight = 5, min_child_samples = 7, max_depth = 7, learning_rate = 0.5, colsample_bytree = 0.1, random_state =
0)
lgbm.fit(X_train, y_train)
previsoes = lgbm.predict(X_test)
cm = ConfusionMatrix(lgbm)
cm.fit(X_train, y_train)
cm.score(X_test, y_test)
classification_lgbm_standard = (classification_report(y_test, previsoes))
print(classification_lgbm_standard)
score_lgbm_standard = 0.8938053097345132
from sklearn.feature_selection import SelectKBest
```

```
from sklearn.feature_selection import chi2
features = X
target = y
best_features = SelectKBest(score_func = chi2,k = 'all')
fit = best_features.fit(features,target)
featureScores = pd.DataFrame(data = fit.scores_,index = list(columns),columns = ['Chi Squared Score'])
featureScores.sort_values(by = 'Chi Squared Score', ascending = False).round(2)
feature_imp
feature_imp_random
Naive_dict_v1 = {'Model':'Naive Bayes',
        'Scaling':'Normal Data',
        'Type':'Gaussian',
        'Precision':score_naive_gaussian}
Decision_dict = {'Model':'Decision Tree',
        'Scaling':'Normal Data',
        'Type': 'Gini',
        'Precision':score_tree}
Random_dict = {'Model':'Random Forest',
        'Scaling':'Normal Data',
        'Type': 'Gini',
        'Precision':score_random}
Extra_dict = {'Model':'Extra Trees',
        'Scaling':'Normal Data',
        'Type': 'Gini',
        'Precision':score_extra}
KNN_dict_v2 = {'Model':'KNN',
        'Scaling':'Normal',
        'Type':'-',
        'Precision':score_knn}
Logistic_dict_v1 = {'Model':'Logistic Regression',
        'Scaling':'Normal Data',
        'Type':'-',
        'Precision':logistic_normal}
ada_dict_v1 = {'Model':'AdaBoost',
        'Scaling':'StandardScaler',
        'Type':'-',
        'Precision':score_ada_scaler}
grad_dict_v1 = {'Model':'GradientBoost',
        'Scaling':'StandardScaler',
        'Type':'-',
        'Precision':score_grad_standard}
lgbm_dict_v1 = {'Model':'LGBM',
```

```
            'Scaling':'StandardScaler',
            'Type':'-',
            'Precision':score_lgbm_standard}
resume = pd.DataFrame({'Naive Bayes':pd.Series(Naive_dict_v1),
                       'Decision Tree':pd.Series(Decision_dict),
                       'Random Forest':pd.Series(Random_dict),
                       'Extra Trees':pd.Series(Extra_dict),
                       'KNN':pd.Series(KNN_dict_v2),
                       'Logistic Regression':pd.Series(Logistic_dict_v1),
                       'AdaBoost':pd.Series(ada_dict_v1),
                       'GradientBoosting':pd.Series(grad_dict_v1),
                       'LGBM':pd.Series(lgbm_dict_v1)
                      })
resume
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ РОЗЛАДУ СНУ ЛЮДИНИ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2023 р.

Вінниця 2023

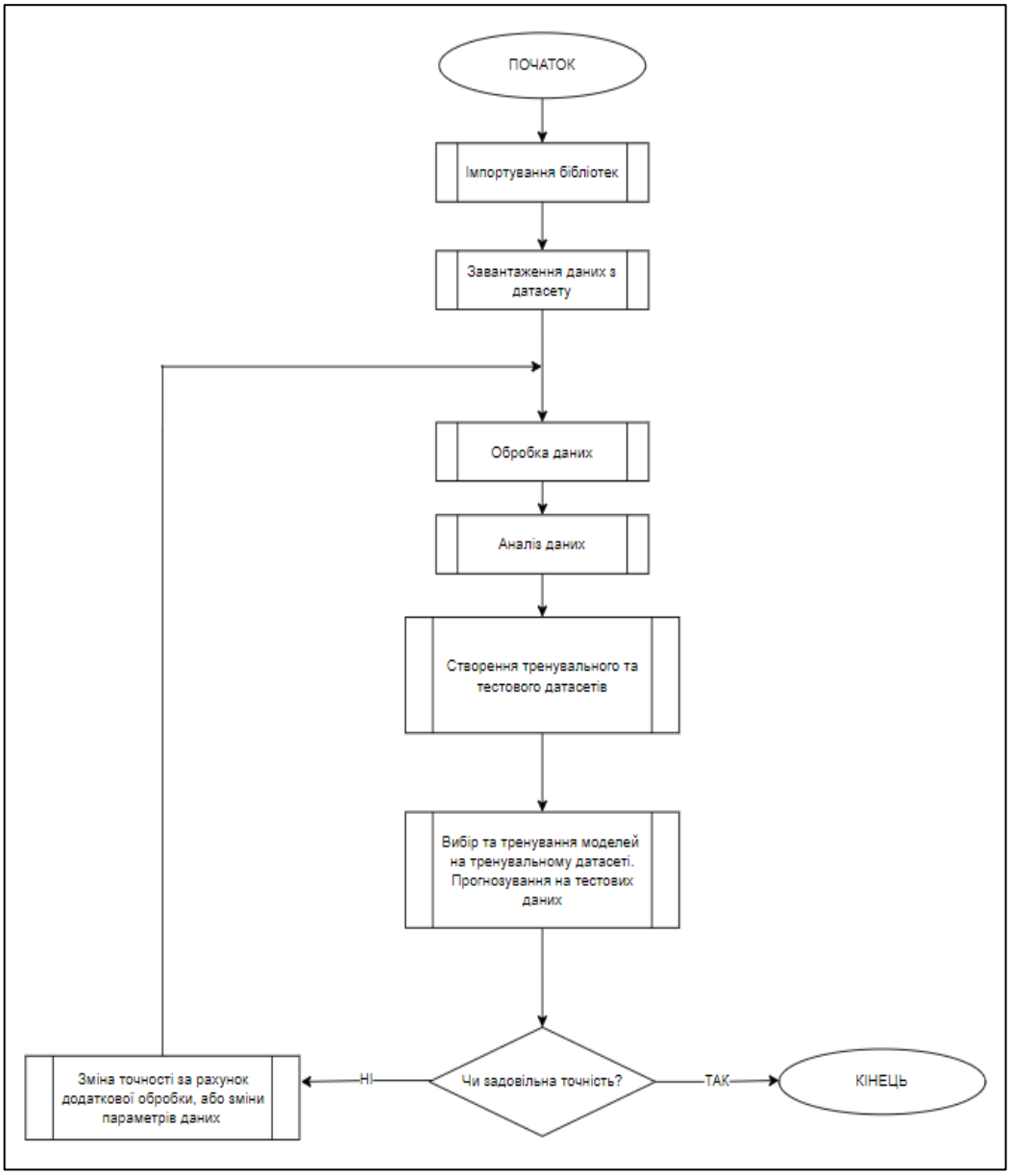


Рисунок Г.1 – Алгоритм роботи інформаційної технології

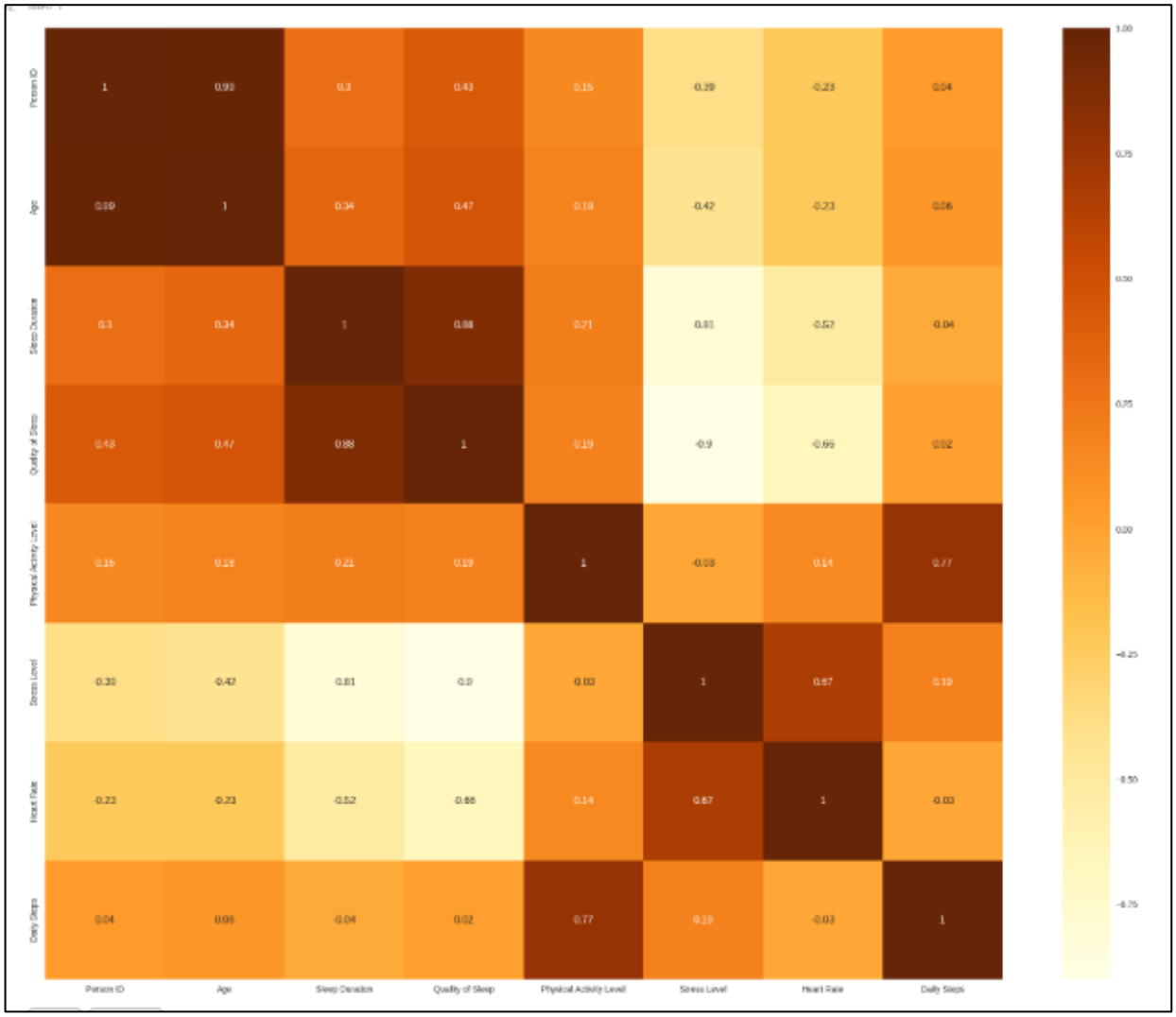


Рисунок Г.2 – Кореляційна матриця

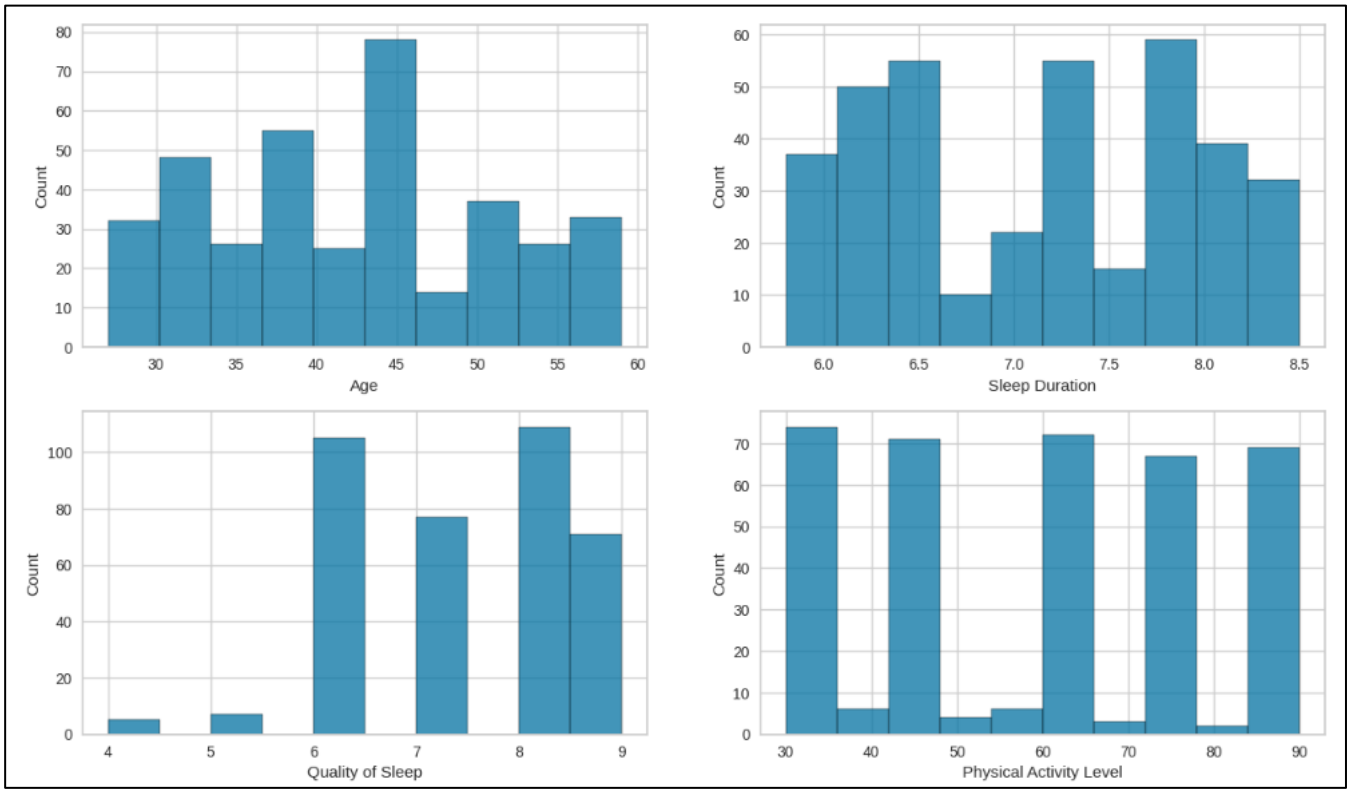


Рисунок Г.3 – Діаграми характеристик з датасету

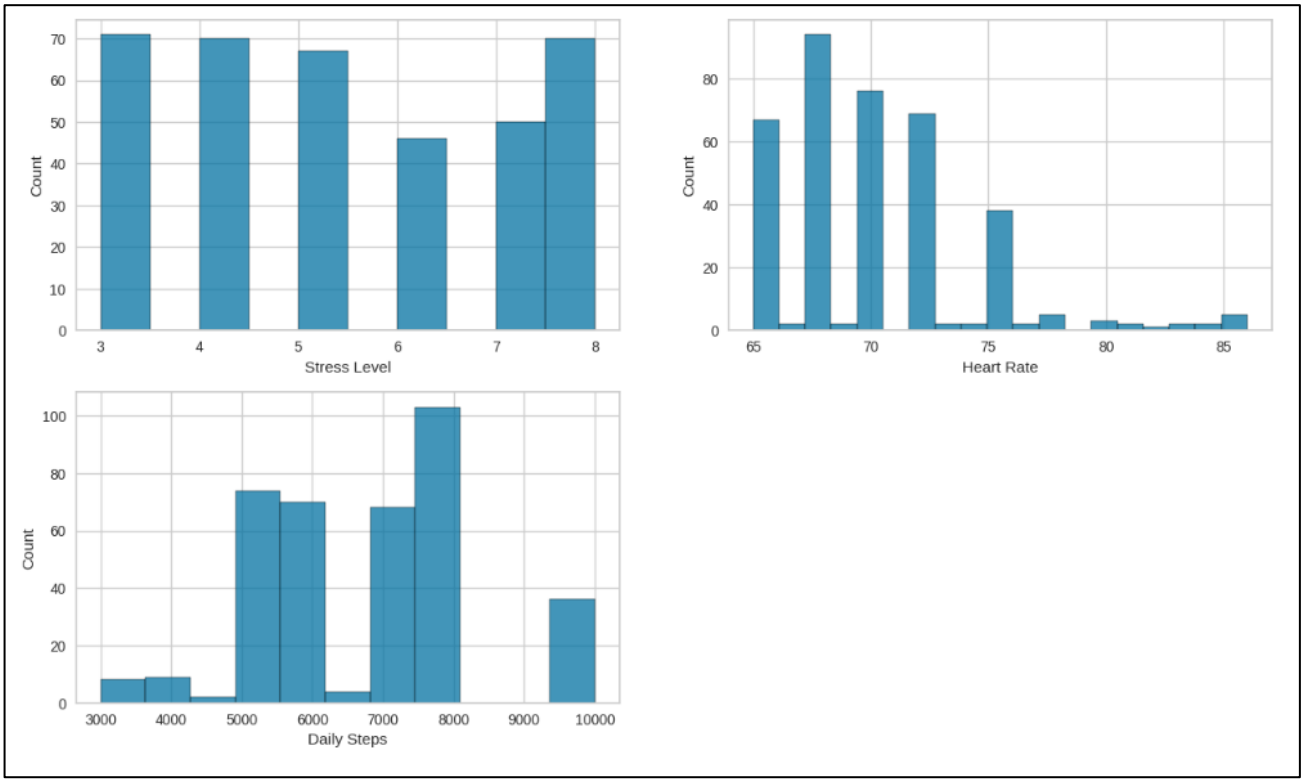


Рисунок Г.4 – Діаграми змінних з датасету

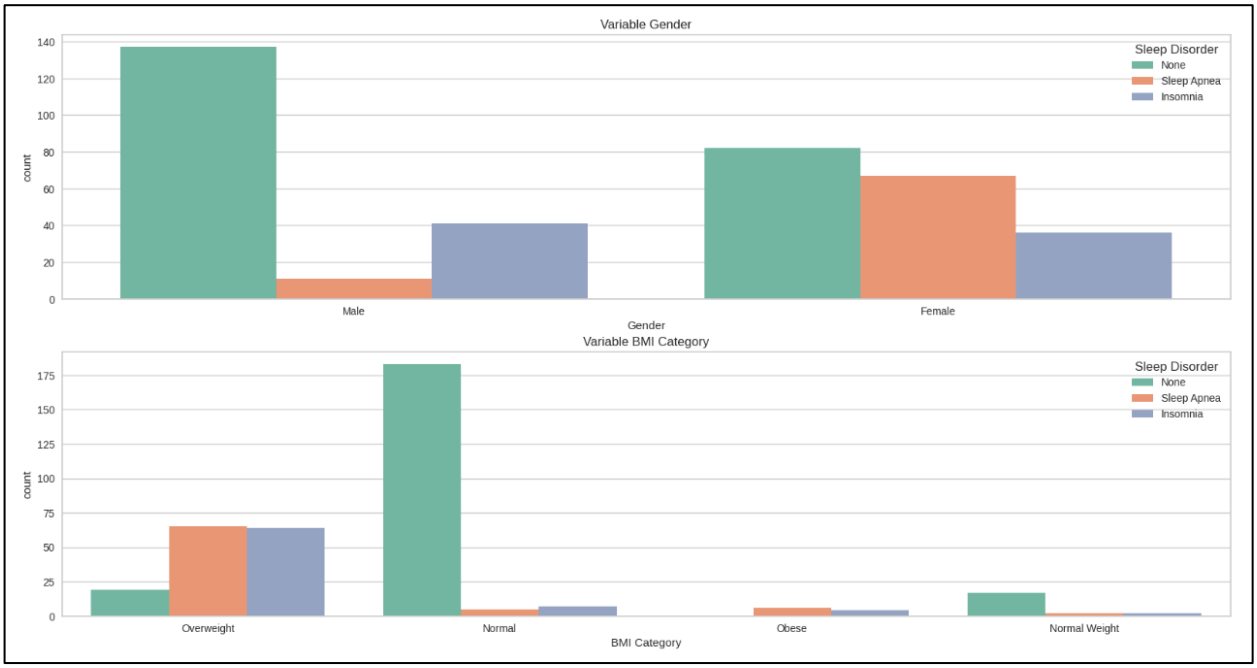


Рисунок Г.5 – Діаграми варіабельних розладів сну по показниках гендеру та ІМТ

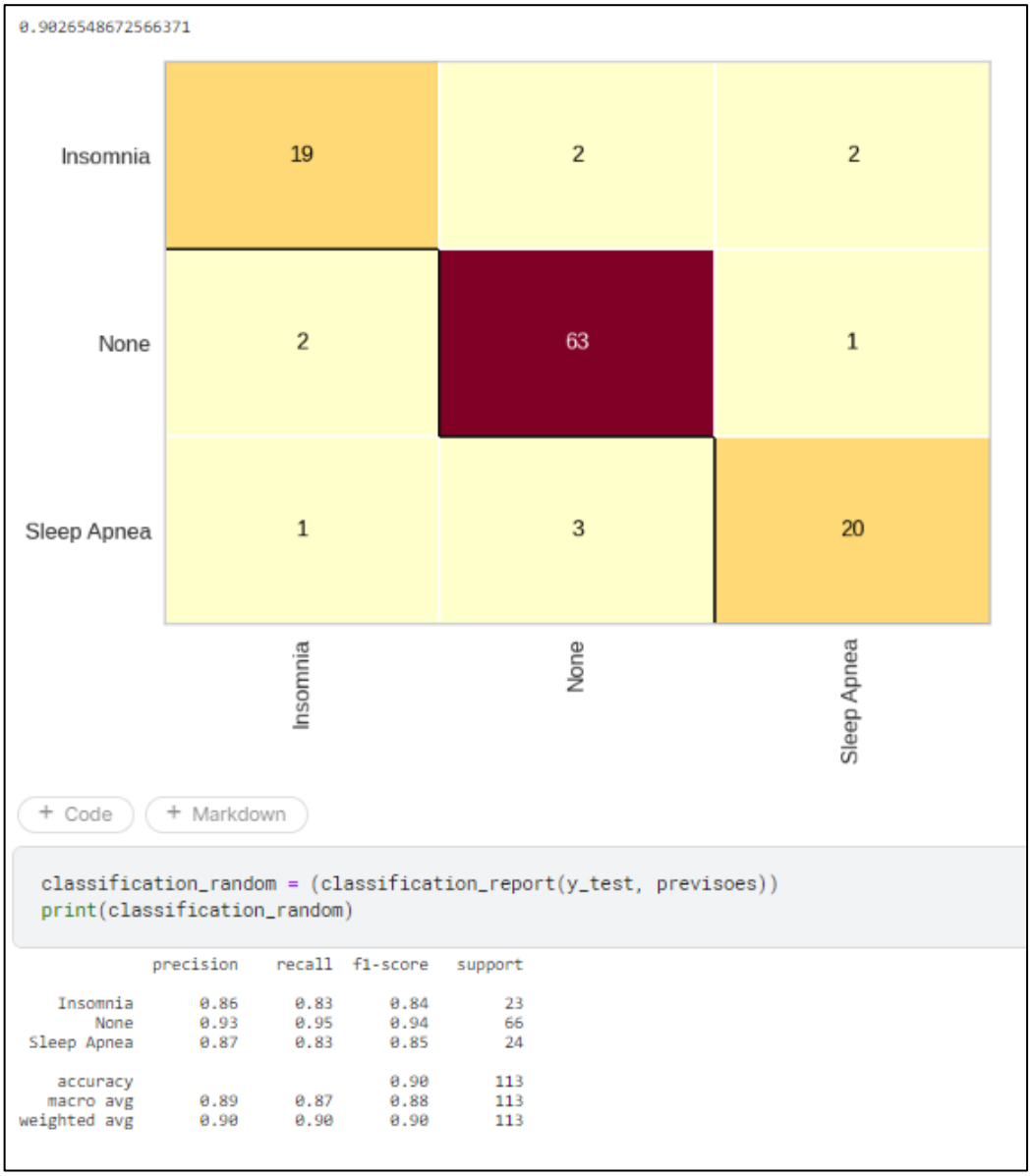


Рисунок Г.6 – Результат моделі Random Forest

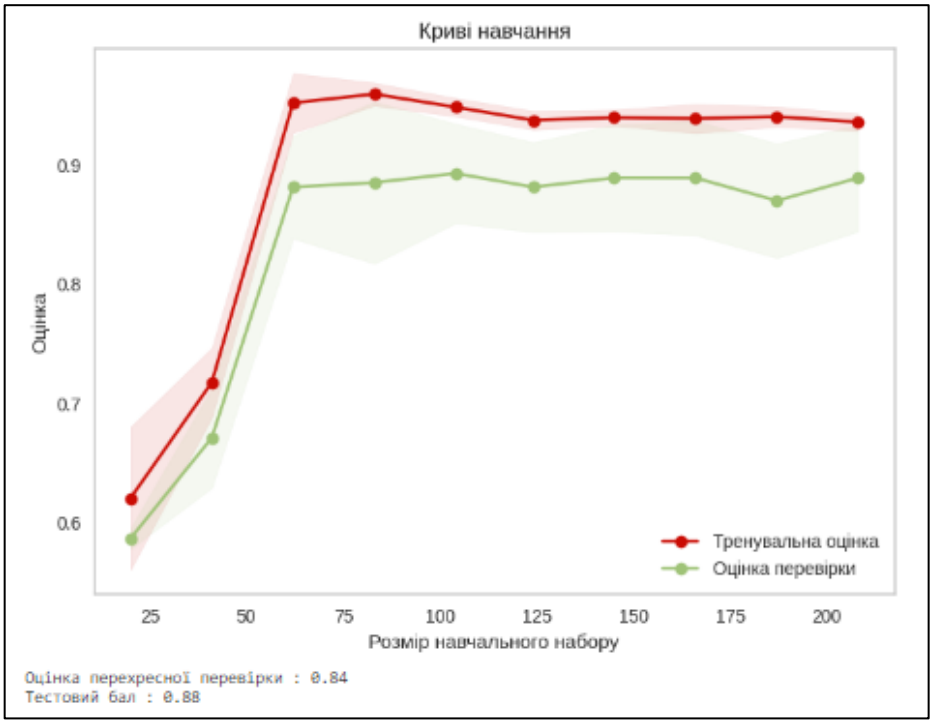


Рисунок Г.7 – Крива навчання для лінійного класифікатора

72]:

	Model	Train Accuracy	Test Accuracy
0	NaiveBayes	0.536398	0.530973
3	ExtraTrees	0.927203	0.884956
4	KNN	0.896552	0.884956
1	DecisionTree	0.927203	0.893805
8	LGBM	0.919540	0.893805
2	RandomForest	0.927203	0.902655
5	LogisticRegression	0.927203	0.911504
6	AdaBoost	0.900383	0.911504
7	GradientBoosting	0.927203	0.920354

Рисунок Г.8 – Кінцевий результат навчальних та тестових даних