Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» _____ 2024 р

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «Комп'ютерне опрацювання біосигналів на основі методів машинного навчання»

Захищено на засіданні
Атестаційної комісії №42
протокол № __ від __.06.2024 р.
Оцінка _____ /
_____
Голова Атестаційної комісії
_____ **СКОБ Ю.О.**

Виконав:
студент 4 курсу, групи КІ–41
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 – Комп’ютерна інженерія.
ВАРОСЯН Арсен Левонович

Керівник: к.т.н., доцент ЗВО кафедри теоретичної та прикладної системотехніки
БАКУМЕНКО Ніна Станіславовна

Рецензент: д.т.н., доцент, професор закладу вищої освіти кафедри теоретичної та прикладної інформатики факультету математики і інформатики
РУККАС Кирило Маркович

Харків – 2024

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і двох додатків. Загальний обсяг роботи складає 82 сторінок, із яких 50 сторінок основної частини з 12 рисунками, 12 найменуваннями списку використаних джерел та чотирьома додатками з двома таблицями.

Метою кваліфікаційної роботи є підвищення точності розпізнавання станів біомедичної системи за допомогою методів машинного навчання.

Об'єкт дослідження – процеси вдосконалення системи моніторингу фізичних вправ на основі біосигналів в умовах апріорної невизначеності вхідних даних.

Предмет дослідження – математичні моделі та обчислювальні методи обробки біосигналів, зокрема методи машинного навчання.

Проблема, яка вирішується в кваліфікаційній роботі, полягає в тому, щоб скориставшись існуючими методами машинного навчання та програмними засобами для обробки біосигналів, мінімізувати витрати часу на створення системи моніторингу фізичних вправ, її відладку та оптимізацію, забезпечити високу точність та надійність отриманих даних, а також ефективно управляти невизначеністю вхідних даних.

Область застосування – розробка систем моніторингу фізичних вправ на основі біосигналів. Розроблений програмний продукт може широко використовуватися в сфері спортивної науки, медицини, реабілітації та фітнесу.

Ключові слова: ЕКГ, ЕЕГ, ЕМГ, біосигнали, Random Forest

ABSTRACT

An explanatory note to a bachelor's thesis consists of an introduction, three sections, conclusions, a list of sources used, and two appendices. The total volume of the work is 82 pages, of which 50 pages are the main part with 12 figures, 12 names of the list of used sources and four appendices with two tables.

The purpose of the qualification work is computer processing of biosignals based on machine learning methods in the R programming language using the tidyverse and randomForest libraries.

The object of research is the process of improving the physical exercise monitoring system based on biosignals in conditions of a priori uncertainty of input data.

The subject of research is mathematical models and computational methods of processing biosignals, in particular, methods of machine learning.

The problem solved in the qualification work is to use existing machine learning methods and software tools for processing biosignals to minimize the time spent on creating an exercise monitoring system, its debugging and optimization, to ensure high accuracy and reliability of the data obtained, and to effectively manage the uncertainty of the input data.

The field of application is the development of exercise monitoring systems based on biosignals. The developed software product can be widely used in the field of sports science, medicine, rehabilitation and fitness.

Keywords: ECG, EEG, EMG, biosignals, Random Forest

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

ЕМК – Електронна медична картка;

ЕКГ – Електрокардіографія;

CNN – Convolutional Neural Networks;

RNN – Recurrent neural network;

SVM – Support vector machine;

OOB – Out-of-band;

RF – Random Forest;

RL – Reinforced learning;

МН – Машинне навчання.

ВСТУП

З поширенням цифрових технологій та зростанням обсягів медичних даних постає важлива проблема ефективного опрацювання біосигналів для точного діагностування та моніторингу стану пацієнтів. Біосигнали, такі як електрокардіограма (ЕКГ), електроенцефалограма (ЕЕГ), електроміограма (ЕМГ) та електроокулограма (ЕОГ), відіграють ключову роль у сучасній медичній практиці, дозволяючи своєчасно виявляти різноманітні патологічні стани та контролювати хід лікування. Однак, обробка та аналіз цих сигналів часто стикаються з численними викликами, зокрема, з шумом, артефактами та варіабельністю сигналів, що робить задачу діагностики складною і трудомісткою.

Актуальність роботи. В контексті сучасної медицини, проблема автоматизації обробки біосигналів стає все більш актуальною. Це пов'язано з необхідністю підвищення точності діагностики, зменшення кількості хибно позитивних та хибно негативних діагнозів, а також з потребою в режимі реального часу відстежувати стан пацієнтів, особливо у випадках критичних станів, таких як серцевий напад чи епілептичний припадок. Застосування методів МН дозволяє значно покращити процеси обробки та аналізу біосигналів, забезпечуючи високу швидкість та точність при виявленні аномалій і патологій.

Метою дослідження кваліфікаційної роботи є підвищення точності розпізнавання станів біомедичної системи за допомогою методів машинного навчання.

Об’єкт дослідження – процеси обробки та аналізу біосигналів з використанням сучасних методів машинного навчання.

Методи дослідження: методи машинного навчання, такі як глибоке навчання, методи з використанням рекурентних нейронних мереж, а також традиційні методи класифікації та регресії; методи обробки сигналів, включаючи

фільтрацію, нормалізацію та виявлення артефактів; методи оцінки ефективності алгоритмів класифікації та регресії.

Предмет дослідження – методи машинного навчання для комп'ютерного опрацювання біосигналів та їх застосування в медичній діагностиці.

Завдання дослідження:

1. Аналіз існуючих методів обробки біосигналів: виконання огляд сучасних методів та інструментів для аналізу біосигналів з використанням машинного навчання, таких як глибоке навчання та рекурентні нейронні мережі.

2. Модифікація існуючих методів: розробка або адаптування методів машинного навчання для обробки біосигналів, які забезпечують високу точність та ефективність аналізу.

3. Виконання оцінки інформативності параметрів біосигналів: проведення оцінки інформативності різних параметрів біосигналів для визначення їх значущості у діагностиці та моніторингу стану користувачів.

4. Аналіз та оцінка отриманих результатів.

РОЗДІЛ 1.
ОГЛЯД ТА АКТУАЛЬНІСТЬ ТЕМИ КОМП'ЮТЕРНОГО ОПРАЦЮВАННЯ БІОСИГНАЛІВ.

Методи машинного навчання базуються на використанні статистичних моделей і алгоритмів для аналізу великих обсягів даних. У контексті медичних біосигналів, ці методи дозволяють виявляти приховані закономірності та кореляції, що може бути важко зробити за допомогою традиційних методів аналізу. Використання методів, таких як глибоке навчання, підтримкові векторні машини, рекурентні нейронні мережі та інші, забезпечує високу точність і швидкість обробки даних, що є критично важливим у медицині

Сучасні технології машинного навчання, зокрема, глибинного навчання, дають новий інструмент для аналізу складних біологічних даних. Застосовуючи біосигнали, людство може знайти складні залежності в даних, які раніше були абсолютно недосяжними для класичних методів.

1.1. Поточне використання.

- Медичний моніторинг. Безперервний аналіз біосигналів дозволяє режимі реального часу отримувати важливу інформацію для спостереження за пацієнтами в різних хірургічних станах, моніторингу їх загального самопочуття під час фізичних навантажень.

Технології реального часу. Сучасні системи медичного моніторингу на основі біосигналів, таких як ЕКГ, ЕЕГ, пульсоксиметрія, можуть надавати точні та миттєві дані про стан пацієнта. Машинне навчання може миттєво виявляти відхилення від норми і реагувати на них, що особливо важливо при таких станах, як серцеві напади або апное уві сні. Це широко використовується для фахівців, які працюють з хірургією та анестезією. Серед можливостей - підвищення контролю над станом пацієнта і реакція на різні зміни в анестезії;

- Діагностичний автоматизований аналіз біологічних сигналів. Автоматизований аналіз діагнозів дозволяє виявити проблеми, які вказують на різні захворювання, аритмії або епілепсію.

Рання діагностика. Машинне навчання здатне аналізувати складні біосигнали та визначати закономірності, які сигналізують про існування хвороби задовго до того, як почнуть проявлятися очевидні симптоми. Цей процес допомагає виявити і почати лікування на ранній стадії хвороби, що покращує прогноз пацієнта. Точність діагностики. Завдяки високій продуктивності та здатності до ретельного аналізу даних, системи машинного навчання розпізнають складні патологічні стани, які діагностика може не помітити за допомогою оптичного аналізу.

- Контроль реабілітаційних пристроїв, управління пристроями. Біосигнали подаються в пристрої для трансляційної реабілітації, засновані на виявленні нервово-м'язової тканини, які призначені для управління протезами або екзоскелетами в процесі реабілітації.

Стосовно інтерфейсів між мозком і комп'ютером, то реабілітація дозволила людям контролювати свої протези або екзоскелети силою свого розуму. Машинне навчання є важливим у цьому аспекті, оскільки саме воно інтерпретує нейронні сигнали і перетворює їх на команди управління пристроєм. Крім того, пристрої стали більш зручними у використанні для людини, в основному зосереджуючись на циклі реабілітації. Завдяки машинному навчанню системи сьогодні підлаштовуються під потреби конкретної людини, що дозволяє їм створювати більш економічне та цілеспрямоване відновлення.

- Контроль фізичного стану та спортивна наука. Широко використовуються у спортивній науці для оптимізації тренувальних та відновлювальних процесів, покращення та інтенсифікації спортивних результатів. Такі технології дозволяють тренерам і спортсменам отримати важливі знання про те, як організм реагує на різні види стресу. Вимірювання параметрів - під час тренувань

спортсмени за допомогою датчиків контролюють такі показники, як частота серцевих скорочень, частота дихання, рівень кисню в крові, м'язова активність та інші. Отримані дані дають можливість оцінити інтенсивність навантаження, ступінь втоми. Такий аналіз біосигналів дає можливість використовувати індивідуальний підхід для складання плану тренувань, зосередитися на оптимізації процесу відновлення, знизити ризики отримання травм [1].

Біосигнали також можна використовувати для аналізу та вдосконалення техніки. За допомогою систем машинного навчання аналізуються рухи, виявляються неефективні або шкідливі прийоми, а потім техніка коригується тренером, який працює зі спортсменом на полі.

1.2. Актуальність.

Тема обробки біосигналів є дуже актуальною в наш час, і ось кілька причин:

1. Досягнення в області сенсорних технологій та збору даних.

Розвиток сенсорних технологій та даних; сучасні сенсорні технології стрімко розвиваються, створюючи можливості для більш чутливих і точних датчиків. Збір даних став більш детальним, а нові типи датчиків, такі як носимі пристрої, розумний текстиль та імплантовані біосенсори, допомагають збирати дані про широкий спектр параметрів, включаючи частоту серцевих скорочень, рівень кисню в крові, мозкову активність та багато іншого. Збір даних дозволив створити детальну картину пацієнта, а отже, підвищити якість діагностики та моніторингу стану пацієнта. Крім того, використання технології Інтернету речей дає можливість збирати дані в режимі реального часу та здійснювати безперервний моніторинг пацієнта увесь час, навіть у віддалених місцях.

2. Великі дані та машинне навчання.

З появою технологій великих даних і методів машинного навчання з'явилася можливість аналізувати складні біологічні сигнали з максимальною ефективністю. Це полегшує постановку точного діагнозу, прогнозування

перебігу хвороби та розробку персоналізованих методів «таргетованої» терапії. Зокрема, методи машинного навчання, що включають нейронні мережі, алгоритми глибинного навчання, кластеризації та класифікації, здатні виявляти приховані закономірності, що містяться в біологічних сигналах, які не піддаються класичним методам [2]. Таким чином, завдяки штучному інтелекту аналіз біосигналів дозволяє автоматизувати процеси аналізу і, відповідно, підвищити ефективність роботи медичних працівників, виключаючи помилки.

3. Онлайн-медицина та телемедичні послуги.

Пандемія COVID-19 значно прискорила розвиток і впровадження телемедичних послуг. Пацієнти часто не могли відвідувати медичні заклади через високий ризик інфікування. У цьому випадку віддалена обробка біосигналів є одним з ключових інструментів. Вона дає можливість лікарям дистанційно контролювати стан пацієнта, консультувати людину і навіть коригувати лікування в режимі реального часу. Потенціал застосування цієї інноваційної технології є високим, враховуючи, що кількість людей з деякими хронічними захворюваннями є значною. Також така технологія потрібна мешканцям віддалених регіонів, адже вони мають обмежену кількість медичних послуг. Крім того, це може бути корисним методом запобігання переповненню медичних закладів, оскільки багато людей просто фізично не прийдуть до них.

4. Актуальність реабілітації поранених на війні.

Через повномасштабне вторгнення Росії у лютому 2022 р. Україна стала найбільш замінованою країною у світі. За оцінками ДСУ (Державної служби України), приблизно 30% території країни є замінованими, та деяка частина з цього відсотку є боєприпаси, що не змогли вибухнути. Задля допомоги пораненим багато лікарень зазнали завеликий наплив пацієнтів, найчастіше у східних і південних областях. У цих лікарнях надають первинну невідкладену допомогу, стабілізують стан пацієнтів, а за необхідності в дію йде хірургічне втручання. Для зменшення навантажень лікарень згодом пацієнтів переводять в

інші «відносно» безпечні регіони (Наприклад, Київ або Вінниця), для подальшого постачання допомоги.

Тобто, можна сказати відверто: потреба надання медичної допомоги різко зросло. За статистикою та даними Міністерства соціальної політики, з моменту ескалації лютого 2022 р. кількість людей з інвалідністю зросла на 300 тис., а попит на фізичних терапевтів з досвідом лікування тяжких травм збільшився вдвічі [3].

5. Персоналізована медицина.

Одним з найважливіших досягнень сучасної медицини є можливість персоналізованого підходу до лікування, що дозволяє сфокусуватися на індивідуальних особливостях пацієнта, який лікується. Аналіз біосигналів і великих даних за допомогою обробки та машинного навчання дає можливість виявлення персональних закономірностей розвитку захворювання та індивідуального планування терапії. Це дає змогу не лише покращити результат лікування за рахунок зменшення побічних ефектів та усунення ризику непередбачуваної реакції організму на ліки, а й врахувати генетичні, екологічні та поведінкові фактори, що забезпечує комплексну підтримку здоров'я пацієнта. Іншим аспектом персоналізованої медицини є адаптація терапії до різних етнічних і культурних груп населення, що дозволяє проводити більш точне і релевантне медичне лікування. Аналіз біосигналів за допомогою машинного навчання дозволяє визначити реакцію пацієнта на терапію та коригувати її в режимі реального часу. Цей метод дуже актуальний для лікування важких захворювань, таких як рак, оскільки "цільовий підхід", заснований на генетичних змінах в організмі пацієнта, є кінцевою метою для кожного конкретного випадку. Крім того, закономірності біомаркерів пацієнта дозволяють розробляти профілактичні заходи, спрямовані на врахування факторів ризику та запобігання захворюванням.

6. Раннє виявлення захворювань.

Ще один важливий момент полягає в тому, що багато захворювань, зокрема, можна вилікувати лише за умови їх виявлення на ранній стадії. За допомогою сучасних методів обробки біосигналів можна виявити навіть перші ознаки патологій задовго до їх виникнення. Так, ЕКГ або електроенцефалограма відносно здорової людини вже дає можливість визначити попередні ознаки серцевого або нервового захворювання. Це стосується і такої гострої проблеми, як рак, де рання діагностика відкриває шлях до успішного лікування. Раннє виявлення хвороби економить не лише кошти на лікування, а й життя пацієнтів. Наприклад, за допомогою аналізу біосигналів можна виявити перші ознаки діабету або гіпертонії, що, в свою чергу, дозволить вжити превентивних заходів, таких як зміна способу життя або проведення лікування, що було б неможливо після інсульту. Виявлення інфекційних захворювань, таких як COVID-19, дозволяє охолодити ланцюги передачі інфекції та вчасно надати допомогу. Таким чином, це також зменшує навантаження на медичну систему та робить значний прогрес у превентивній медицині [4].

7. Розвиток технологій носимих пристроїв.

Ринок пристроїв, що носяться, таких як фітнес-трекери, смарт-годинники та медичні монітори, стрімко зростає. Ці пристрої здатні регулярно відстежувати фізичні параметри людини: фізичну активність, сон, стрес тощо. Якщо отриману інформацію збирати та аналізувати за допомогою інструментів машинного навчання, то можна виявити ознаки змін у стані здоров'я та вчасно на них відреагувати. Як наслідок, покращується якість життя користувачів, а також розвивається превентивна медицина, оскільки можна виявити ранні ознаки та симптоми. Носимі пристрої також необхідні для моніторингу стану здоров'я в режимі реального часу, що є критично важливим для пацієнтів з хронічними захворюваннями. Наприклад, люди з серцево-судинними проблемами можуть безперервно відстежувати свій серцевий ритм за допомогою смарт-годинника і отримувати сповіщення про аритмію або інші проблеми. Отримані дані можуть бути надіслані лікарям для аналізу та контролю. Носимі пристрої часто просто

синхронізуються з мобільними додатками для здоров'я, які пропонують рекомендації щодо фізичних вправ, планів харчування та інших змін у способі життя.

8. Інтеграція з електронними медичними записами (EMR).

Обробка біосигналу інтегрована з електронною медичною картою. Поєднання цих двох систем допоможе створити цілісну картину здоров'я пацієнта. Унікальна база даних може зберігати та аналізувати великі обсяги даних з різних джерел, а це означає, що діагноз лікаря стає більш точним, а лікування - більш ефективним. Безперервно оновлюючи карту пацієнта, без ручного втручання можна буде отримувати актуальні дані з біосенсорів та клінічну інформацію. Інтеграція інформаційних систем різних спеціалістів та аптек також сприяє забезпеченню безперервності лікування. Ще один аспект - зв'язок біосигналу з електронними медичними картами. Вивчаючи обсяг анамнестичних даних, кількість попередніх візитів і проведених досліджень, лікар може більш точно зрозуміти стан пацієнта і прийняти рішення про лікування. В системі ЕМК є можливість відстежувати тенденції та загальні закономірності в перебігу захворювання. Завдяки інтеграції ЕД ми зменшимо кількість помилок, пов'язаних з ручним введенням даних.

9. Розвиток реабілітаційних технологій.

Обробка біосигналів та машинне навчання також можуть бути застосовані в реабілітаційних технологіях. Наприклад, електроміограма може бути використана для аналізу стану м'язів людини під час реабілітаційних процедур, в результаті чого можна розробити план реабілітації для кожного пацієнта і відстежувати прогрес в режимі реального часу. Наразі використання біологічного зворотного зв'язку в реабілітації може допомогти у створенні адаптивних тренажерів та пристроїв, які допоможуть підлаштовувати пристрої під потреби пацієнта, а отже, полегшити одужання та ще швидше відновити активний спосіб життя. Крім того, у поєднанні з технологіями віртуальної та доповненої

реальності біосигнали можуть бути використані для створення персональних тренувальних середовищ, які привертають увагу пацієнтів та залучають їх до активного відновлення. Так, віртуальний тренажер може відтворювати реальні життєві сценарії, які допоможуть пацієнту відновити рухові функції та інші навички. Біосигнали слугуватимуть для вимірювання та автоматичного коригування тренувального процесу, таким чином залучаючи пацієнтів до одужання. Такі тренувальні технології можна використовувати дистанційно, щоб допомогти у відновленні навіть тим пацієнтам, які перебувають вдома.

10. Психічне здоров'я.

Ще одна перспективна сфера застосування біосигнального моніторингу - психічне здоров'я. Наприклад, аналіз змін частоти серцебиття, електропровідності шкіри та інших показників може допомогти діагностувати стрес, тривогу та депресію. Машинне навчання, що використовується для подальшого аналізу даних, формує індивідуальні «патерни», які можна легко впровадити в мобільні додатки або носимі пристрої для моніторингу даних у режимі реального часу. Таким чином, це дозволяє проводити постійну діагностику та вчасно реагувати на зміни. Обробка біосигналів у психічному здоров'ї також може бути використана для створення сценаріїв біо- та нейрофідбеку. Останній дозволяє пацієнту отримувати зворотній зв'язок від поточного стану його мозку. Пацієнт в цьому випадку отримує комплексне втілення власного мозку, що дозволяє йому бачити свій стан в режимі реального часу і розвивати необхідні навички контролю стресу [5]. Наприклад, датчики серцебиття і дихання можуть контролювати його в розслабленій позі під час медитації, що поступово дозволяє пацієнтові контролювати свій рівень стресу. Ефективність психотерапії також можна виміряти за допомогою моніторингу біосигналів, що робить психотерапію більш об'єктивною та керованою даними.

11. Спортивна медицина та оптимізація тренувань.

Також біосигнали використовують у спортивній медицині, щоб допомогти спортсменам ефективніше проводити тренування. Аналізуючи інформацію під час занять, можна визначити оптимальні режими, запобігти перевантаженням і травмам спортсмена, а також стежити за його відновленням. Застосування машинного навчання біосигналів у спорті дає можливість створити персональний режим тренувань на основі індивідуальних фізіологічних особливостей спортсмена, що якісно підвищує ефективність тренувань і допомагає досягти високих результатів. Дані з датчиків дадуть можливість спортивним командам і тренерам відстежувати динаміку здоров'я під час змагань або навіть тренувань. Ця інформація дозволить оцінити готовність спортсменів і, за необхідності, скоригувати їхні тренувальні режими для забезпечення максимально ефективної роботи та відновлення. Крім того, обробка сигналів може допомогти оцінити ефективність плану вправ та інших методів, які використовують тренери. Це сприяє більш глибокому науковому обґрунтуванню тренувального процесу спортсменів. Отже, використання такої технології є критично важливим для безпеки спортсменів, оскільки вона може виявити та запобігти надзвичайним ситуаціям медичного характеру.

1.3. Важливість використання методів машинного навчання для аналізу медичних даних.

Сьогодні методи, засновані на машинному навчанні в галузі медицини, сприяють успішній діагностиці, лікуванню та профілактиці різних захворювань. У сучасній медицині машинне навчання включається в методи аналізу та обробки медичних даних, серед яких є біосигнали. Біосигнали - це електричні сигнали, що генеруються біологічними системами: серцем, мозком, м'язами, необхідні для діагностики та моніторингу стану пацієнта [6]. Наука машинного навчання в аналізі біосигналів дозволяє точно ідентифікувати, прогнозувати, лікувати захворювання.

По-перше, підвищення точності діагностики. Методи машинного навчання дають можливість аналізувати велику кількість даних і виявляти складні

закономірності, які можуть бути недостатньо помітними для лікаря. Це, наприклад, аналіз електрокардіограм для діагностики захворювань серця або електроенцефалограм з метою виявлення неврологічних розладів. Методи машинного навчання, зокрема, на основі глибоких нейронних мереж, дозволяють автоматично помічати відхилення в біосигналах, що значно підвищує точність діагностики. Електрокардіограма є найважливішим діагностичним інструментом для виявлення серцевих патологій. Існуючі методи аналізу ЕКГ засновані на візуальній оцінці лікаря, яка може бути суб'єктивною і, крім того, передбачає високий професіоналізм фахівця, який зробив таку оцінку. Методи машинного навчання з використанням машини опорних векторів і рекурентної нейронної мережі RNN дозволяють автоматизувати цей процес і досягти високої точності для виявлення таких патологій, як аритмія, ішемічна хвороба серця і так далі.

По-друге, оптимізація лікувальних процесів. Розуміючи результати лікування на основі попередньої інформації, алгоритм машинного навчання може допомогти передбачити найкращий метод лікування для конкретного пацієнта. Для пацієнтів з тривалими захворюваннями, такими як рецидивуючі, хронічні або інші важкі випадки, машинне навчання може допомогти розрахувати кількість ліків, які потрібно ввести, або процес реабілітації, який потрібно запустити. Таким чином, використання машинного навчання допомагає підбирати індивідуальне лікування для конкретного пацієнта, застосовуючи несприятливі методи лікування з меншою кількістю побічних ефектів або взагалі без них. Наприклад, пацієнту з діабетом можна порадити максимальну кількість інсуліну відповідно до його рівня інсуліну, споживання вуглеводів і фізичних навантажень.

По-третє, покращення клінічних випробувань та прискорення пошуку ефективності. Логічно, що машинне навчання також може допомогти поліпшити дизайн і процес проведення клінічних випробувань. Алгоритми можуть легко визначити, які люди будуть найкращими кандидатами для окремих випробувань, а також як найкраще проаналізувати дані, зібрані під час випробувань, і оцінити

ефективність нових ліків і методів лікування. Це безпосередньо призводить до скорочення часу, витраченого на розробку нових медичних технологій, і, як наслідок, до підвищення їхньої якості. Для нових лікарських засобів аналіз даних, отриманих в ході клінічних випробувань, про побічні ефекти, ефективність лікування та інших подібних і важливих для індустрії показників, дозволяє в кінцевому підсумку дати точне визначення безпечності та ефективності нових ліків, що дозволяє заощадити час і знизити витрати на їх розробку. При цьому алгоритми машинного навчання можуть обробляти ці дані з усіх етапів випробувань.

Нарешті, автоматизація рутинної роботи. Алгоритми машинного навчання можуть перетворити багато повторюваних процесів, таких як обробка медичних зображень або збір даних про пацієнтів, на тривіальні, дозволяючи персоналу, якого не вистачає, зосередитися на більш серйозних питаннях. Багато рутинних процесів у медицині можна автоматизувати за допомогою алгоритмів машинного навчання, наприклад, обробку медичних зображень або збір даних про пацієнтів. Це може бути автоматичне розпізнавання патологій за зображеннями або моніторинг стану пацієнта за допомогою натільних пристроїв. Така автоматизація також може допомогти медичним працівникам виконувати більш важливу і складну роботу, що підвищить рівень доступності медичної допомоги для населення. Обробка медичних зображень - важливий етап у діагностиці. Використання алгоритмів глибокого навчання, або CNN, дає змогу машині самостійно аналізувати медичні зображення. Йдеться про рентгенівські знімки, МРТ, КТ та інші. Штучна нейронна мережа може ідентифікувати ознаки патологій, як пухлинних, так і переломних та інфекційних. Це дозволяє не тільки підвищити точність діагностики, а й заощадити час на проведення аналізу.

Підсумовуючи, машинне навчання є цінною палітрою для обробки та аналізу даних біосигналів. У майбутньому використання машинного навчання в аналізі біосигналів зробить значний внесок. З часом, з розвитком технологій і зростанням обсягів медичних даних, збільшиться і кількість шансів розробити

правильні способи діагностики, лікування та профілактики захворювань. Рутинна робота буде автоматизована, з'являться нові методи індивідуальної роботи з пацієнтами, що зробить медицину більш ефективною та здоровою. Це відкриває нові перспективи для виявлення, лікування та прогнозування захворювань. Інтеграція методів машинного навчання в медицину сприяє досягненню належної точності та ретельності діагностики, прискоренню та наочності процесу лікування, отриманню кращих результатів та створенню персоналізованої політики, а також створенню можливостей для ручних та рутинних завдань, що підвищує рівень медичного обслуговування та благополуччя людей.

Висновки за розділом 1

Демонструється значна актуальність і перспективність використання методів машинного навчання для аналізу біосигналів у медичних дослідженнях та практиці. Сучасні алгоритми машинного навчання, такі як глибинне навчання та рекурентні нейронні мережі, забезпечують високу точність та швидкість обробки великих обсягів медичних даних, що сприяє ранній діагностиці та ефективному моніторингу пацієнтів у режимі реального часу. Це має особливе значення для персоналізованої медицини, де індивідуальні особливості пацієнтів враховуються для підвищення ефективності лікування та зменшення побічних ефектів. Впровадження машинного навчання в медичну практику також відкриває нові можливості для розвитку реабілітаційних технологій, покращення спортивних тренувальних програм, оптимізації управління протезами та екзоскелетами, а також вдосконалення психічного та фізичного здоров'я. Завдяки цьому підходу, медичні працівники отримують потужні інструменти для більш точного діагностування, прогнозування перебігу захворювань і розробки персоналізованих методів лікування, що в кінцевому підсумку покращує якість життя пацієнтів та підвищує загальну ефективність медичних послуг.

РОЗДІЛ 2.
ОГЛЯД МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ БІОСИГНАЛІВ

2.1. Визначення біосигналів та їх класифікація

Біосигнали - це сигнали, які генеруються живими організмами, вимірюються та аналізуються для отримання інформації про фізіологічний стан, функціонування та поведінку цих організмів. Ці сигнали пов'язані з різними фізіологічними процесами і можуть бути електричними, хімічними, механічними та тепловими. Насправді, обробка та вивчення біосигналів широко використовуються в багатьох напрямках сучасної медицини, в більшості випадків для діагностики та моніторингу стану організму [7].

2.1.1. Основні типи біосигналів.

1. Електрокардіограма – ЕКГ зазвичай застосовується для оцінки порушень роботи серця, наприклад, аритмії, інфаркту міокарда та інших.

2. Електроенцефалограма, а саме електрична активність головного мозку та перцептору. ЕЕГ дає глибоке розуміння нейронної активності, а її аналіз може бути використаний для побудови перцепторів нейронних мереж. Крім того, ЕЕГ використовується для діагностики неврологічних розладів, наприклад, епілепсії, черепно-мозкових травм, порушень сну та багатьох інших.

3. Електроміограма – показує стан м'язів-розтяжок і скелетних м'язів. ЕМГ допомагає діагностувати патологію нервів і визначає різні типи м'язових проблем.

4. Електроокулограма – пояснює, який потенціал виникає між рогівкою та сітківкою ока.

2.1.2. Класифікація біосигналів.

Типи біосигналів можуть бути різними, залежно від кількох критеріїв:

1. За джерелом сигналу - нервові, серцеві, м'язові тощо.

2. За методом вимірювання - інвазійні та неінвазійні.

3. За характером сигналу: постійні та змінні.

2.1.3. Існуючі комп'ютерні додатки та інформаційні системи.

1. Bioconductor - відкрита платформа для аналізу біологічних даних, яка включає численні пакети для аналізу даних біосигналів. Основні функції: Аналіз геномних даних, аналіз експресії генів, інтеграція з RStudio.

2. Kubios HRV - програмне забезпечення для аналізу варіабельності серцевого ритму (HRV) з ЕКГ записів. Основні функції: Аналіз HRV, обробка сигналів, графічний інтерфейс користувача.

3. BioSig - відкрите програмне забезпечення для аналізу біомедичних сигналів, розроблене для використання з MATLAB та Octave. Основні функції: Обробка та аналіз ЕКГ, ЕЕГ, ЕМГ сигналів, підтримка різних форматів даних.

4. MNE-Python - бібліотека для аналізу та візуалізації даних MEG та EEG з відкритим кодом, написана на Python. Основні функції: Обробка сигналів, аналіз джерел, візуалізація.

5. EEGLAB - Інтерактивний інструмент MATLAB для обробки даних ЕЕГ. Основні функції: Передпроцесинг ЕЕГ, аналіз незалежних компонентів (ICA), візуалізація.

Загалом, також треба відзначити інтеграцію з RStudio: можна використовувати наступні пакети та бібліотеки, що дозволяють аналізувати біосигнали:

- **Signal:** Пакет для роботи з цифровою обробкою сигналів у R. Основні функції: Фільтрація сигналів, аналіз спектрів, обробка часових рядів.
- **Eegkit:** Пакет для аналізу ЕЕГ даних. Основні функції: Передпроцесинг даних ЕЕГ, аналіз, візуалізація.

- **HRV:** Пакет для аналізу варіабельності серцевого ритму. Основні функції: Аналіз часових та частотних доменів HRV, інтеграція з іншими біосигналами.

2.2. Огляд основних методів машинного навчання в медичному застосуванні.

Загалом можна описати мінімальні семантичні вимоги до всіх методів машинного навчання, що застосовуються в медичній практиці. Разом з тим, характеристики цього гарантованого відтворення повинні якнайкраще акцентувати увагу на їхній суті та особливостях наступних методів:

2.2.1. Навчання з вчителем (Supervised Learning)

1. Логістична регресія (Logistic Regression): це статистичний метод аналізу набору даних, в якому одна або кілька незалежних змінних фіксують ймовірність настання певної події або стану [8]. У медичних дослідженнях логістична регресія найчастіше використовується для прогнозування розвитку досліджуваного захворювання на основі декількох клінічних і біологічних показників пацієнта. Як приклад, можна навести випадок, коли лікарі загальної практики використовують логістичну регресію для розрахунку ймовірності серцевого нападу, враховуючи кров'яний тиск, рівень холестерину, вік і вагу. Перевага логістичної регресії полягає в тому, що вона пропонує кількісну оцінку ймовірностей, що є корисним для прийняття клінічних рішень. Крім того, вона може включати велику кількість предикторів і може обробляти нелінійні взаємозв'язки між змінними за допомогою логістичної кривої.

Однією з найпоширеніших методик у медичній статистиці є логістична регресія. Це найпоширеніший метод аналізу бінарних або дихотомічних відповідей. Цей метод використовується для оцінки ступеня зв'язку незалежних факторів, які називаються предикторами, з дихотомічною залежною змінною. Застосування: Медичні експерти можуть використовувати логістичну регресію для оцінки ризику серцево-судинних захворювань. Такі фактори, як куріння, високий кров'яний тиск, діабет, рівень холестерину та багато інших, часто

використовуються як коваріанти в дослідженнях. Це дозволяє лікарям оцінити конкретний потенційний ризик пацієнта і розробити план профілактики або терапії на основі об'єктивних параметрів.

Логістична регресія оцінює ймовірність події P(Y=1) як:

$P(Y = 1 \mid X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_k X_k)}}$, де:

- Y - залежна змінна (цільова змінна), яка приймає значення 0 або 1.
- $X_1, X_2, \ldots, X_k$ - незалежні змінні (предиктори).
- β_0 - вільний член (константа).
- $\beta_1, \beta_2, \ldots, \beta_k$ - коефіцієнти моделі.
- e - основа натурального логарифму.

Логістична функція (Сігмоїда): логістична функція (або функція Сігмоїда) перетворює лінійну комбінацію предикторів в значення від 0 до 1:

$$\sigma(z) = \frac{1}{1+e^{-z}}, \text{де: } z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

Відношення шансів:

В логістичній регресії часто використовується відношення шансів (odds ratio): $Odds = \frac{P(Y=1 \mid X)}{1-P(Y=1 \mid X)}$ і логарифм відношення шансів (log-odds): $\log(Odds) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$

2. Метод опорних векторів (Support Vector Machines, SVM): як найбільш прийнятний метод класифікації в області складних медичних описів і відхилень, враховуючи ефективність "розумного" поділу класів даних. Це поширений метод машинного навчання, який використовується для класифікації, регресії та інших цілей [9]. SVM корисний для медичної класифікації складних даних, таких як МРТ-зображення, генетичні дані або біомаркери, серед іншого. Його можна використовувати, коли класи не можуть бути ефективно розділені лінійно на графіку автентичної функції. Для цього метод проектує дані на вісь більшої

розмірності, де поділ на класи є більш лінійним. Він працює шляхом знаходження бічного поділу між класами, коли всі інші вирівнювання є неупередженою перспективою. Він є стійким до шуму та надмірного припасування, оскільки зіставлення може бути вирішене шляхом латерального розділення обмеженої кількості складних медичних даних Кенії.

SVM - це ефективний і високоточний інструмент класифікації, який також важливий для медицини, як однієї з найбільш життєво важливих галузей. Приклади: медичні зображення, такі як рентгенівські знімки, МРТ-сканування, аналіз клітинних зображень для виявлення ракових клітин тощо, є життєво важливими для виявлення інформаційних патернів. Сила SVM полягає у виявленні складних взаємозв'язків між даними, які не піддаються лінійному розділенню, і мінімізації першої та другої послідовних помилок; ці деталі мають вирішальне значення для створення медичних інструментів для діагностики.

Мета SVM - знайти таку гіперплощину $w * x + b = 0$, яка максимально відокремлює два класи даних. Для цього необхідно максимізувати відстань між цією гіперплощиною і найближчими точками кожного класу (опорними векторами).

Відстань до гіперплощини: відстань від точки x_i до гіперлощини визначається як: $d = \frac{|w*x_i+b|}{||w||}$

Оптимізація задачі: задача оптимізації формулюється таким чином, щоб максимізувати відстань між гіперплощиною і опорними векторами, що еквівалентно мінімізації ‖w‖. Це можна записати як задачу квадратичного програмування: $min_{w,b}\frac{1}{2}||w||^2$ при умовах: $y_i(w * x_i + b) \geq 1, \forall i$

Метод Лагранжа: для вирішення цієї задачі використовують множники Лагранжа. Функція Лагранжа для задачі SVM має вигляд: $L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n}\alpha_i[y_i(w * x_i + b) - 1]$, де α_i – множники Лагранжа.

Двоїста задача: $max_a \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i * x_j)$. При умовах: $\sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0$

Вирішення задачі SVM у випадку нелінійних даних: для нелінійних даних використовують ядрові функції (Kernel functions), які проектують дані у простір вищої розмірності, де вони стають лінійно розділеними. Найбільш популярні ядра:

- Лінійне ядро: $K(x_i, x_j) = x_i * x_j$
- Поліноміальне ядро: $K(x_i, x_j) = (x_i * x_j + c)^d$
- Радіально-базисне ядро (RBF): $K(x_i, x_j) = \exp(-\gamma \left|\left|x_i - x_j\right|\right|^2)$

Прогнозування: Після навчання моделі SVM, прогнозування класу нової точки x здійснюється за допомогою функції: $f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b)$.

3. Дерева рішень (Decision Trees) і Випадкові ліси (Random Forests): як методи аналізу та інтерпретації складних даних із занадто великою кількістю вхідних змінних. Випадкові ліси для обходу надмірної підгонки для великих обсягів даних.

Дерево рішень - це модель прогнозування, що складається з правил, згідно з якими приймаються рішення на основі визначених атрибутів [10]. Коли йдеться про медичні дослідження, дерева рішень підходять для аналізу більш складних даних, таких як медичні записи пацієнтів, щоб знайти діагностичний шлях і розробити можливу стратегію лікування. Основною перевагою дерев рішень є їхня інтерпретованість, що є життєво важливим у сфері медицини, оскільки медичний працівник повинен розуміти, чому модель дала ту чи іншу рекомендацію.

Випадкові ліси - це ансамбль дерев рішень, розроблених для регулювання високої точності та перенавчання. Загалом, цей метод можна охарактеризувати багатофакторною точністю і обмеженою тенденцією до перенавчання. Він підходить для аналізу медичних даних, що містять багато змінних.

Особливе місце в цьому списку займають ті методи, які дозволяють створювати прогнози або класифікаційні моделі на основі історичних даних. Ці моделі мають можливість працювати з великими обсягами даних і багатьма параметрами. Прикладом використання таких алгоритмів може бути наступне: за допомогою дерев рішень аналізуються клінічні дані, а далі, за допомогою випадкових лісів, можна підвищити точність і запобігти перенавчання, яке часто спостерігається з індивідуальними деревами рішень. Це може показати і покаже більш глибокі взаємозв'язки в даних, хоча вони можуть бути пропущені будь-яким іншим дослідником. Викладені методи і різні види їх розробки та оптимізації даних відіграють важливу роль у поліпшенні діагностики захворювань, в організації планування лікування та профілактики. Такі методи дозволяють медичному персоналу приймати рішення, що ґрунтуються на фактах.

Формула дерева рішень: Для будь-якого атрибута A з набором значень V можна визначити інформаційну ентропію E(S) для вибірки S: $E(S) = -\sum_{i=1}^{c} p_i log_2(p_i)$, де p_i - ймовірність класу i у вибірці S, а c - кількість класів.

Інформаційний приріст IG при розщепленні вибірки S по атрибуту A визначається як: $IG(S,A) = E(S) - \sum_{v \in V} \frac{|S_v|}{|S|} E(S_v)$, де S_v – підмножина S, в якій атрибут A має значення v.

2.2.2. Навчання без вчителя (Unsupervised Learning)

1. Кластеризація (Clustering): здійснюється для пошуку природних угруповань або закономірностей у даних, і може бути використана для пошуку нових кореляцій між характеристиками або категоріями пацієнтів на основі їхніх медичних записів.

$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} ||x_j - \mu_i||^2$, де C_i – множина об’єктів, що належать до кластера i.

2. Аналіз головних компонент (Principal Component Analysis, PCA): використовується для зменшення розмірності даних без знищення значної

частини інформації, що міститься в даних, щоб було легше аналізувати та візуалізувати велику кількість рядків і стовпців у медичних базах даних.

PCA використовується для зменшення розмірності даних. Вона проектує дані на новий простір з меншим числом вимірів, максимально зберігаючи варіацію в даних.

Коваріаційна матриця: $\sum = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$, де x_i – і-й зразок даних, $\bar{x}$ – середнє значення всіх зразків, n – кількість зразків.

Собственні вектори та власні значення: $\Sigma v = \lambda v$, де Σ – ковариаційна матриця, v – власний вектор, λ – власне значення.

Перетворення даних: Z=XW, де X – матриця даних, W – матриця власних векторів, Z – перетворені дані.

2.2.3. Навчання з підкріпленням (Reinforcement Learning).

RL - метод машинного навчання, в якому агент вивчає, як діяти в певному середовищі, щоб максимізувати певну довгострокову винагороду. У медичних застосуваннях RL можна використовувати для створення систем, які самоадаптуються до змін обставин, наприклад, для автоматичного регулювання дозування ліків у реальному часі. Далі наведено огляд основних понять RL разом із відповідними формулами.

Принаймні в медицині - його можна використовувати для розробки систем, які самоадаптуються до змін обставин об'єкта, наприклад, для регулювання дозування ліків у реальному часі.

2.2.4. Глибинне навчання (Deep Learning)

1. Конволюційні нейронні мережі (Convolutional Neural Networks, CNNs): один з найпоширеніших способів діагностики захворювань. При вимірюванні медичними визначеннями - рентгенівські промені, МРТ або комп'ютерна

томографія формують медичний опис, завдяки якому такі мережі можуть ідентифікувати найважливіші характеристики з високим рівнем достовірності.

2. Рекурентні нейронні мережі (Recurrent Neural Networks, RNNs): відмінною рисою є те, що вони обробляють послідовні дані. Репрезентативним прикладом цього типу моделей є часовий ряд ЕКГ або ЕЕГ - дані залежать від часу.

2.3. Огляд методу Random Forest

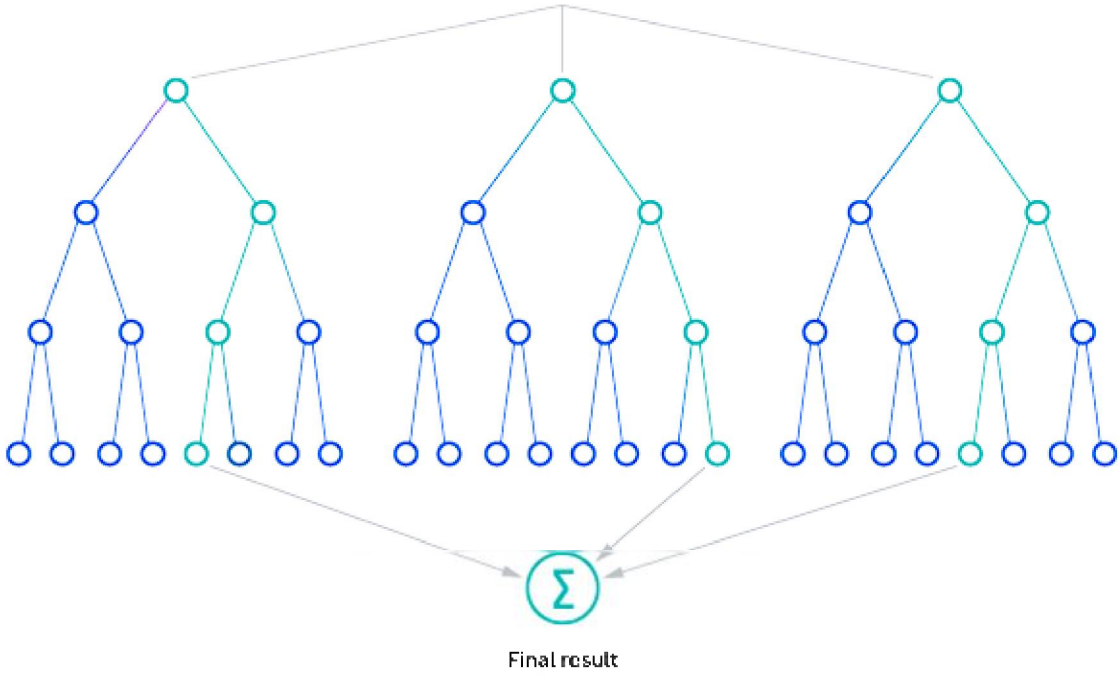


Рисунок 2.1 – Візуальне зображення застосування випадкового лісу

2.3.1. Основні принципи роботи Random Forest.

Random Forest (RF) - Потужна і універсальна модель завдяки своїй здатності обробляти великі набори даних з високою вимірністю, знижувати перенавчання і надавати інтерпретовані метрики важливості ознак. Це метод ансамблевого навчання, який використовується головним чином для класифікаційних і регресійних завдань. Він працює шляхом побудови кількох

дерев рішень під час навчання та видачі класу, який є модальним для класів (класифікація) або середнім прогнозом (регресія) окремих дерев [11]. Ось детальні наукові принципи, що лежать в основі алгоритму Random Forest:

- Дерева рішень як базове навчання: Random Forest побудований на деревах рішень, які є простими нелінійними класифікаторами, що розподіляють дані на підмножини на основі значень ознак. Кожен вузол у дереві представляє правило рішення щодо певної ознаки, а кожна гілка представляє результат цього правила. Листи представляють кінцевий результат класифікації або регресії.
- Беггінг (Bootstrap Aggregating): Random Forest використовує техніку під назвою беггінг, де кілька підмножин даних генеруються шляхом випадкової вибірки з поверненням з початкового набору даних. Кожна підмножина використовується для навчання окремого дерева рішень. Цей процес допомагає знизити дисперсію і запобігає перенавчанню.
    a. Незалежні дерева: кожне дерево в лісі навчається незалежно від інших, що забезпечує різноманітність серед дерев. Дерева можуть бути глибокими і зазвичай вирощуються без обрізання, що дозволяє їм захоплювати більш складні шаблони.
- Випадковий вибір ознак: на кожному розгалуженні дерева вибирається випадкова підмножина ознак, і найкраще розділення вибирається тільки з цієї підмножини. Цей процес відомий як баггінг ознак. Він вводить додаткову випадковість і ще більше декорелює дерева, покращуючи стійкість моделі.
- Агрегація прогнозів: у контексті класифікації кожне дерево в лісі віддає голос за класову мітку. Класова мітка, яка отримує більшість голосів, вибирається як кінцевий прогноз.
    a. Середнє значення (регресія): Для регресійних завдань кінцевий прогноз отримується шляхом усереднення прогнозів усіх окремих дерев.
- Компроміс зміщення-розсіювання: поєднання кількох дерев знижує ризик високого зміщення (недонавчання) порівняно з одним деревом рішень.

a. Зниження розсіювання: Агрегація кількох різноманітних дерев також знижує розсіювання (перенавчання), що призводить до більш стабільних і точних прогнозів.

- Оцінка помилки за позабутстраповими зразками (OOB): під час навчання, оскільки кожне дерево навчається на bootstrap-вибірці, близько однієї третини даних не використовується для навчання кожного дерева (OOB-зразки). Ці OOB-зразки використовуються для оцінки помилки моделі без потреби в окремому валідаційному наборі.
  a. OOB-помилка: Середній рівень помилок, розрахований з використанням OOB-зразків, слугує як необмежена оцінка помилки узагальнення моделі Random Forest.
- Важливість ознак: Random Forest може обчислювати важливість кожної ознаки. Це здійснюється шляхом вимірювання зменшення Gini impurity (або іншого критерію) на кожному розділенні і усередненням його по всіх деревах у лісі. Ознаки, які стабільно більше зменшують impurity, вважаються більш важливими.
  a. Permutation Importance: Інший спосіб оцінити важливість ознак - виміряти збільшення помилки моделі при перемішуванні значень певної ознаки. Більше збільшення помилки вказує на більшу важливість ознаки.
- Обробка відсутніх значень: Random Forest може обробляти відсутні значення, використовуючи замінні розділення. Коли значення відсутнє для ознаки в точці розділення, модель використовує замінну ознаку, яка найкраще наближає початкове розділення, щоб ухвалити рішення.
- Масштабованість і паралелізація:
  a. Паралельна обробка: Незалежна природа навчання дерев у Random Forest дозволяє легко паралелізувати процес. Кожне дерево може бути вирощене на окремому процесорному ядрі, що значно прискорює процес навчання для великих наборів даних.
  b. Масштабованість: Random Forest є масштабованим для великих наборів даних і високовимірних даних. Використання випадкової вибірки і вибору ознак забезпечує ефективність алгоритму з точки зору обчислень.

- Математична формуляція:
  a. Побудова дерева: для кожного дерева T_b, b = 1, 2, … , B (де B – кількість дерев):
     i. Зробити bootstrap-вибірку Z* розміру N з навчальних даних.
     ii. Виростити дерево T_b на bootstrap-даних шляхом рекурсивного повторення наступних кроків для кожного вузла:
        1. Вибрати m змінних випадковим чином з доступних p ознак.
        2. Вибрати найкращу змінну і точку розділення серед m.
        3. Розділити вузол на два дочірніх вузла.
  b. Прогноз:
     i. Для класифікації: $\hat{y} = argmax_c \sum_{b=1}^{B} I(T_b(x) = c)$
     ii. Для регресії: $\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$
  c. OOB-помилка:
     i. Обчислити OOB-помилку шляхом усереднення помилок прогнозу для всіх дерев з використанням OOB-зразків.

2.3.2. Переваги «Випадкового лісу» порівняно з іншими алгоритмами машинного навчання:

- Метод стійкий до перенавчання - як зазначалося, багато дерев і випадковість у виборі ознак роблять «Random Forest» набагато менш схильним до цього типу помилок, ніж одне дерево.
- Висока точність – «Random Forest» зазвичай мають найвищу точність серед класичних алгоритмів класифікації та регресії.
- Може працювати з великими наборами даних з великою кількістю ознак - випадковий ліс може ефективно працювати з великими і складними наборами даних.

Недоліки:

- Складна модель та обчислювальні вимоги. Навчання складного набору дерев може зайняти більше часу і вимагати більше обчислювальних ресурсів. Це може

різко знизити ефективність цього методу. Він не може бути використаний у деяких випадках з дуже великими наборами даних.

- Складність інтерпретації моделі. Оскільки набір дерев менш інтуїтивно зрозумілий, ніж одне дерево рішень, складно точно передбачити, на якому дереві базується результат. Навіть коли людина має доступ до інформації про дерево, на якому ґрунтується результат, знання того, як і чому, не пояснюється і вимагає спеціальних знань.
- Менш ефективний на даних, які мають "шумову" складову. У таких випадках цей метод може бути значно менш ефективним порівняно з іншими, більш спеціалізованими.

Висновки за розділом 2

Основні типи біосигналів, такі як ЕКГ, ЕЕГ, ЕМГ та ЕОГ, мають важливе значення для діагностики та моніторингу фізіологічного стану пацієнтів. Розглядаються різні підходи до їхньої класифікації залежно від джерела сигналу, методу вимірювання та характеру сигналу. Огляд існуючих програмних додатків та інформаційних систем підкреслює важливість інструментів, таких як Bioconductor, Kubios HRV, BioSig, MNE-Python та EEGLAB, для аналізу та обробки біосигналів. Ці інструменти надають можливості для передпроцесингу, аналізу та візуалізації біосигналів, а також інтегруються з RStudio, що робить їх зручними для використання у наукових дослідженнях та медичній практиці. Методи машинного навчання, такі як логістична регресія, метод опорних векторів (SVM), дерева рішень та випадкові ліси, а також глибинне навчання, мають широке застосування в медичних дослідженнях та практиці. Кожен з цих методів має свої переваги та обмеження, які визначають їхню ефективність для різних типів даних та завдань. Зокрема, метод Random Forest виявляється особливо корисним для класифікації біосигналів завдяки своїй здатності обробляти великі набори даних з високою вимірністю, знижувати перенавчання та надавати інтерпретовані метрики важливості ознак. Він забезпечує високу

точність та стійкість до перенавчання, що робить його потужним інструментом для аналізу медичних даних. Таким чином, використання сучасних методів машинного навчання та програмних інструментів для аналізу біосигналів може значно підвищити точність діагностики та ефективність медичних досліджень, сприяючи розвитку персоналізованої медицини та покращенню якості медичного обслуговування.

РОЗДІЛ 3.
ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАДАЧІ КЛАСИФІКАЦІЇ БІОСИГНАЛІВ

3.1. **Аналіз інструментальних засобів для вирішення задач машинного навчання методів для класифікації біосигналів**

MATLAB – відомий між інженерами й дослідниками, найбільш в галузях в яких з'являється потрібність до підвищеної надійності числових розрахунків

Інструменти: вбудовані функції з метою простежування сигналів та відтворень. Систематично є гарним для опрацювання біосигналу.

Середовище: підсвідомо доступне середовище та графічний інтерфейс.

Ліцензія: нерідко студенти мають доступ до «MATLAB» у академічних установах, але ціна іноді може виглядати доволі зависокою для індивідуальних користувачів.

2. Python. Представляє собою майже найпопулярнішу мову програмування в світі науки про дані та машинного навчання, завдяки своїй адаптивності й неосяжному набору бібліотек.

Бібліотеки: Scikit-learn - традиційне в машинному навчанні. Надає просторий спектр алгоритмів машинного навчання, в тому числі класифікацію, регресію та кластеризацію, немало підходячи для нульового (елементарного) аналізу біосигналів. TensorFlow та PyTorch для глибокого навчання. Дозволяють створювати складні моделі глибокого навчання, які можуть виявляти складні закономірності в біосигналах.

Інтеграція: здатний вільно інтегруватися з рештою технологій та інструментами.

Спільнота: обширна спільнота розробників та обширна документація.

Отже «Python» дійсно є непоганою середою для аналізу біосигналів, але вимагає широке усвідомлення в програмуванні та алгоритмів машинного навчання для продуктивного застосування.

3. WEKA

Застосування: WEKA є новітнім набором ПЗ для аналізу даних і прикладного машинного навчання.

Переваги: інтуїтивно зрозумілий графічний інтерфейс, широкий вибір готових до використання алгоритмів машинного навчання.

Недоліки: Обмежена масштабованість і гнучкість порівняно з програмуванням на Python або MATLAB.

4. Java. Використовуватися для машинного навчання, хоча вона не так популярна у цій області, як Python.

Виконання: хороша продуктивність для великих систем і можливість розгортання великих додатків.

Бібліотеки: Weka, Deeplearning4j для машинного навчання.

5. Julia. Відносно нова мова, яка поєднує швидкість C зі зручністю Python.

Швидкість: висока продуктивність, особливо при обробці великих масивів даних.

Наукові обчислення: велика кількість пакетів для статистики та машинного навчання.

6. R. Мова програмування та середовище, які популярні в наукових дослідженнях, особливо в статистиці, біоінформатиці та епідеміології. R традиційно використовується в статистиці та біоінформатиці.

Переваги:

1. Велика кількість пакетів для статистичного аналізу та обробки даних, зокрема для біомедичних досліджень.

2. Статистичні можливості: незмірний набір пакетів для статистичного аналізу, що робить його найліпшим варіантом для важких статистичних розрахунків. Пакети як randomForest, caret, та ranger приділяє величезні можливості для обробки аналізу Random Forest.

3. Як й «Python», у «R» одна з найактивніших та великих спільнот у наукових дослідженнях, це означає багатий вибір навчальних ресурсів, форумів підтримки, та вільно доступних пакетів для розширення функціональності.

4. Інтеграція з RStudio: RStudio - це широко використовуване середовище розробки для мови R, яке спрощує операції кодування, тестування та візуалізації. Вона має вбудовані інструменти для робочого простору, візуалізації та щоденника коду під назвою R Markdown. Завдяки цьому процес інтеграції досліджень стає більш структурованим.

5. Гнучкість у візуалізації даних: Мова R, особливо з пакетом ggplot2, надає користувачеві універсальний спосіб створення високоякісної графіки. Цей момент має вирішальне значення для проведення необхідного аналізу та інтерпретації біосигналів.

6. Відкритий вихідний код: R є безкоштовною мовою з відкритим вихідним кодом, що дозволяє необмежено адаптувати існуючі фреймворки до індивідуальних потреб.

Недоліки:

1. Швидкість виконання: R може бути досить повільною при роботі з великими наборами даних, особливо у порівнянні з мовами програмування, призначеними для роботи на високопродуктивних обчислювальних системах, такими як Python з вбудованими бібліотеками C++.

2. Керування пам'яттю: R завантажує кожен фрагмент даних в оперативну пам'ять, що може призвести до проблем з продуктивністю при роботі з великими обсягами даних.

3. Взаємодія з іншими мовами: Хоча існують пакунки, які дозволяють інтерфейсувати R з іншими мовами програмування (наприклад, Rcpp для C++), це потребує додаткового налаштування та оптимізації.

Підводячи підсумок, вибір для Random Forest - це R. Причиною такого вибору є численні спеціалізовані пакети, створені для цієї мови програмування, такі як randomForest, які вже були оптимізовані для алгоритму, що розглядається. Крім того, ці пакети дозволяють широко модифікувати параметри, перевіряти моделі та аналізувати важливість змінних, що може мати вирішальне значення для правильної інтерпретації результатів у галузі класифікації біосигналів. Таким чином, використання R є особливо корисним для потреб, які потребують розширеного і глибокого статистичного аналізу та інтерпретації даних.

3.2. **Розвідувальний аналіз даних**

Кожен IMU має значення x, y і z + кути Ейлера (крени, тангаж і поворот). Для кожного часового вікна (1 с даних) існує кілька статистичних обчислень, як-от ексцес, дисперсія тощо. [12]

Є два датасети з назвами:

1. «pml-training.csv»: на якому буде проводитися навчання. Кількість записів: 19622. Кількість змінних: 160
2. «pml-testing.csv»: тестувальний датасет. Кількість записів: 20. Кількість змінних: 160

Майже усі змінні між двома датасетами є однаковими, але у «pml-training.csv» присутня результативна змінна «classe», яка оцінює результат виконаної вправи. А у «pml-testing.csv» замість неї присутня інша змінна – «problem_id», яка ідентифікує кожен запис від 1 до 20.

3.2.1. Короткий опис кожної змінної

Усі частині тіла, на яких були датчики та гантель (А саме: пояс (belt), рука (arm), гантель (dumbbell), предпліччя (forearm)), мають стандартизовані назви та типи змінних, тому описувати усі не має сенсу, через це задля розуміння спростимо до прикладу з рукою (arm):

- roll_arm: кут повороту руки навколо осі, яка йде вздовж тіла;
- pitch_arm: кут нахилу рук вперед-назад;
- yaw_arm: кут обертання рук навколо вертикальної осі;
- total_accel_arm: загальне прискорення, виміряне на руках;
- gyros_arm_x: кутова швидкість обертання рук навколо осі X;
- gyros_arm_y: кутова швидкість обертання рук навколо осі Y;
- gyros_arm_z: кутова швидкість обертання рук навколо осі Z;
- accel_arm_x: прискорення вздовж осі X на руках;
- accel_arm_y: прискорення вздовж осі Y на руках;
- accel_arm_z: прискорення вздовж осі Z на руках;
- magnet_arm_x: магнітне поле вздовж осі X на руках;
- magnet_arm_y: магнітне поле вздовж осі Y на руках;
- magnet_arm_z: магнітне поле вздовж осі Z на руках.

Отже зрозуміло, що датасет повний інформації стосовно фізичних активностей, записаних в ньому.

Відомо, що дані були зібрані для шести користувачів (user_name), які виконували ці вправи.

Також присутні змінні з іменами користувачів (user_name), які виконували ці вправи, вони будуть використані для побудови трьохвимірних графіків та вирішення невизначених записів (Які знаходяться в «pml-testing.csv»), щоб можна було надати якісну оцінку.

3.3. **Обробка даних**

Для подальшої роботи обираються змінні, зв’язані з датчиками на поясі, на це впливає декілька причин:

1. Зменшення обсягу даних: використання лише змінних, пов'язаних з руками, може значно зменшити кількість даних, що аналізуються. Це може прискорити обчислення та зменшити вимоги до пам'яті.

2. Фокусування на важливих змінних:

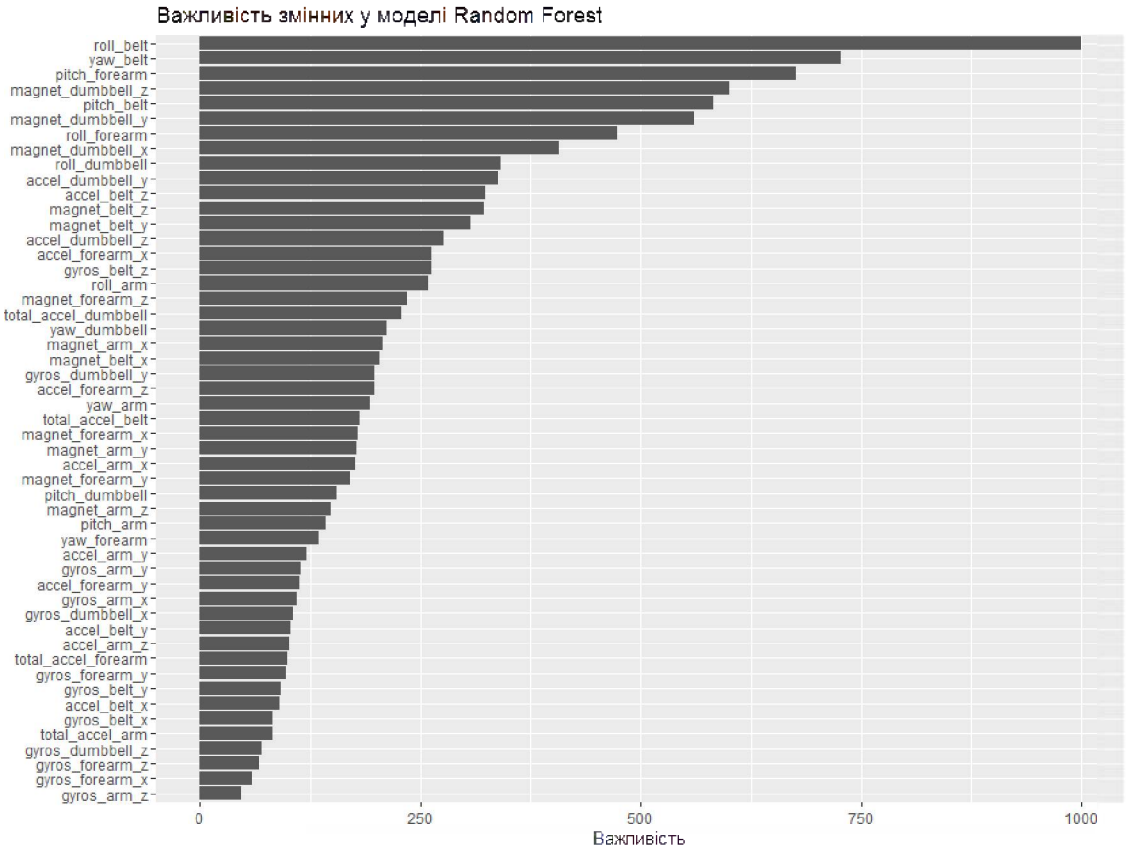


Рисунок 3.1 – Важливість змінних

Як можна побачити, найбільш важливі змінні для моделі випадкового лісу є змінні пов’язані з датчиками на поясі, тому ліпше всього буде використовувати змінні пов’язані саме з цим датчиком. Ця таблиця вказує на те, що змінні,

пов'язані з датчиками на поясі, є найбільш інформативними для вирішення завдання, фокусування на цих змінних може покращити точність моделі.

3.4. **Робота з пропущеними значеннями**

В першу чергу треба обробити дані, зрозуміло, що в датасеті є дані/записи, що були пропущені, невизначені або непотрібні для аналізу. Ці дані створюють шум. Необхідно їх прибрати для ліпшої точності моделі.

main.R × | train_data × | test_data ×

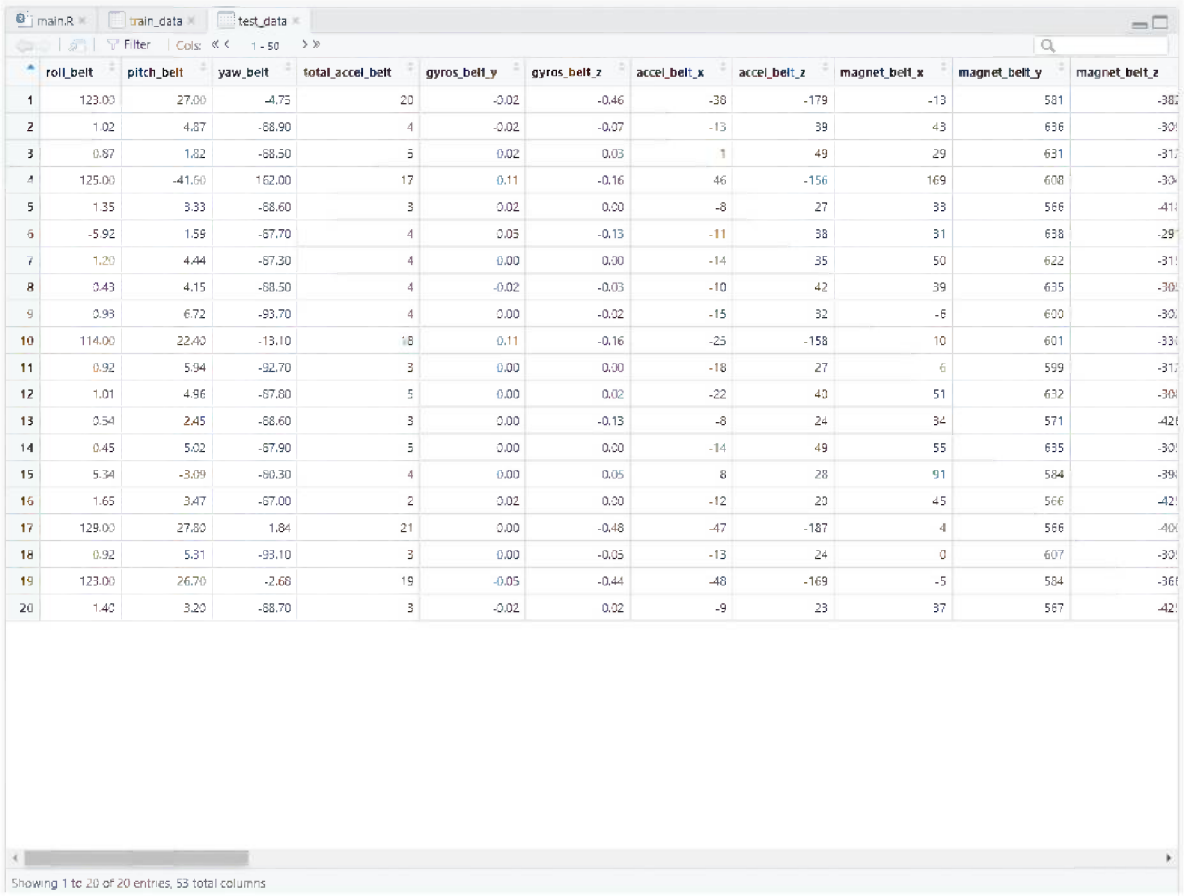
Filter | Cols: « < 4 - 53 > »

	total_accel_belt	gyros_belt_y	gyros_belt_z	accel_belt_x	accel_belt_z	magnet_belt_x	magnet_belt_y	magnet_belt_z
1	3	0.00	-0.02	-21	22	-3	599	-313
2	3	0.00	-0.02	-22	22	-7	608	-311
3	3	0.00	-0.02	-20	23	-2	600	-305
4	3	0.00	-0.03	-22	21	-6	604	-310
5	3	0.02	-0.02	-21	24	-6	600	-302
6	3	0.00	-0.02	-21	21	0	603	-312
7	3	0.00	-0.02	-22	21	-4	599	-311
8	3	0.00	-0.02	-22	21	-2	603	-313
9	3	0.00	-0.02	-20	24	1	602	-312
10	3	0.00	0.00	-21	22	-3	609	-308
11	3	0.00	-0.02	-21	23	-5	596	-317
12	3	0.00	-0.02	-22	23	-2	602	-319
13	3	0.00	0.00	-22	21	-3	606	-309
14	3	0.00	-0.02	-22	21	-8	598	-310
15	3	0.00	0.00	-21	22	-1	597	-310
16	3	0.00	0.00	-21	23	0	592	-305
17	3	0.00	-0.02	-21	22	-6	598	-317
18	3	0.02	0.00	-21	21	1	600	-316
19	3	0.00	-0.02	-20	21	-3	603	-313
20	3	0.00	-0.02	-22	22	-1	604	-314
21	3	0.00	-0.02	-20	20	-10	607	-304
22	3	0.02	-0.02	-21	21	-2	604	-313
23	3	0.00	-0.02	-21	21	-4	606	-311
24	3	0.00	-0.02	-20	22	-3	601	-318
25	3	0.00	0.00	-19	21	-8	605	-319
26	3	0.00	0.00	-21	22	-10	601	-312
27	3	0.00	-0.02	-22	22	3	597	-320
28	3	0.00	-0.02	-21	21	-4	599	-317

Showing 1 to 29 of 19,622 entries, 53 total columns

Рисунок 3.2 – Обробка пропущених значень (train_data)

Після фільтрування датасету очевидно, що кількість змінних скоротилася з 160 до 53 стовпців, обробка виконана вірно. По тому ж самому алгоритму оброблено й файл «test_data» - так само 53 стовпця:



main.R | train_data | test_data

Filter | Cols: « < 1 - 50 > »

	roll_belt	pitch_belt	yaw_belt	total_accel_belt	gyros_belt_y	gyros_belt_z	accel_belt_x	accel_belt_z	magnet_belt_x	magnet_belt_y	magnet_belt_z
1	123.00	27.00	-4.75	20	-0.02	-0.46	-38	-179	-13	581	-382
2	1.02	4.87	-88.90	4	-0.02	-0.07	-13	39	43	636	-303
3	0.87	1.82	-88.50	5	0.02	0.03	1	49	29	631	-312
4	125.00	-41.60	162.00	17	0.11	-0.16	46	-156	169	608	-304
5	1.35	3.33	-88.60	3	0.02	0.00	-8	27	33	566	-411
6	-5.92	1.59	-87.70	4	0.05	-0.13	-11	38	31	638	-291
7	1.20	4.44	-87.30	4	0.00	0.00	-14	35	50	622	-311
8	0.43	4.15	-88.50	4	-0.02	-0.03	-10	42	39	635	-305
9	0.98	6.72	-93.70	4	0.00	-0.02	-15	32	-6	600	-302
10	114.00	22.40	-13.10	18	0.11	-0.16	-25	-158	10	601	-331
11	0.92	5.94	-92.70	3	0.00	0.00	-18	27	6	599	-317
12	1.01	4.96	-87.60	5	0.00	0.02	-22	40	51	632	-308
13	0.54	2.45	-88.60	3	0.00	-0.13	-8	24	34	571	-428
14	0.45	5.02	-87.90	5	0.00	0.00	-14	49	55	635	-303
15	5.34	-3.09	-80.30	4	0.00	0.05	8	28	91	584	-398
16	1.65	3.47	-87.00	2	0.02	0.00	-12	20	45	566	-423
17	129.00	27.80	1.84	21	0.00	-0.48	-47	-187	4	586	-406
18	0.92	5.31	-93.10	3	0.00	-0.05	-13	24	0	607	-301
19	123.00	26.70	-2.68	19	-0.05	-0.44	-48	-169	-5	584	-368
20	1.40	3.20	-88.70	3	-0.02	0.02	-9	23	37	567	-423

Showing 1 to 20 of 20 entries, 53 total columns

Рисунок 3.3 – Обробка пропущених значень (test_data)

3.5. **Аналіз статистичних зв’язків**

Створюється таблиця невідповідності, яка покаже результати порівняння реальних та прогнозованих значень для п’яти класів («A», «B», «C», «D», «E»):

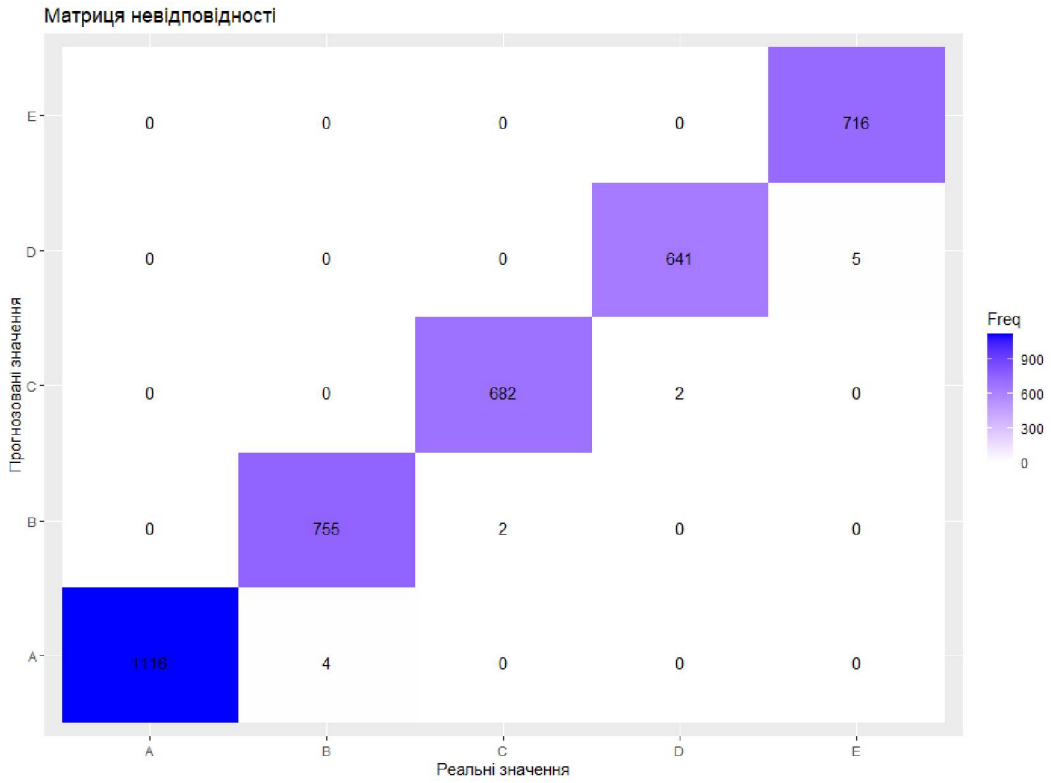


Рисунок 3.4 – Матриця невідповідності

Висновки стосовно матриці невідповідності:

1. Висока точність для класів: прогнозування для класів A, B, C, D та E є дуже точним, оскільки більшість елементів знаходяться на діагоналі матриці, що означає правильне передбачення класів. Це вказує на те, що модель добре навчається і має високу точність у класифікації цих класів.

2. Низький рівень помилок: майже всі значення, які не знаходяться на діагоналі, мають значення 0, що свідчить про відсутність помилкових передбачень або дуже малу їх кількість. Це означає, що модель робить дуже мало помилок при прогнозуванні класів.

3. Особливо висока точність для класу «A»: клас A має 1116 правильних прогнозів і лише 4 неправильні (помилково класифіковані як B). Це свідчить про те, що модель надзвичайно точна в класифікації класу A.

4. Висока точність для інших класів:

- Клас B має 755 правильних прогнозів і 4 неправильні.
- Клас C має 682 правильних прогнозів і лише 2 неправильні.
- Клас D має 641 правильних прогнозів і 5 неправильних.
- Клас E має 716 правильних прогнозів і 0 неправильних.

5. Відсутність переплутаних класів: відсутність значень поза діагоналлю матриці свідчить про те, що модель практично не переплутує класи між собою. Це свідчить про хорошу здатність моделі розрізняти класи.

Тепер, коли всі підготовчі процеси закінчено, створюється кореляційна матриця для всіх змінних belt:

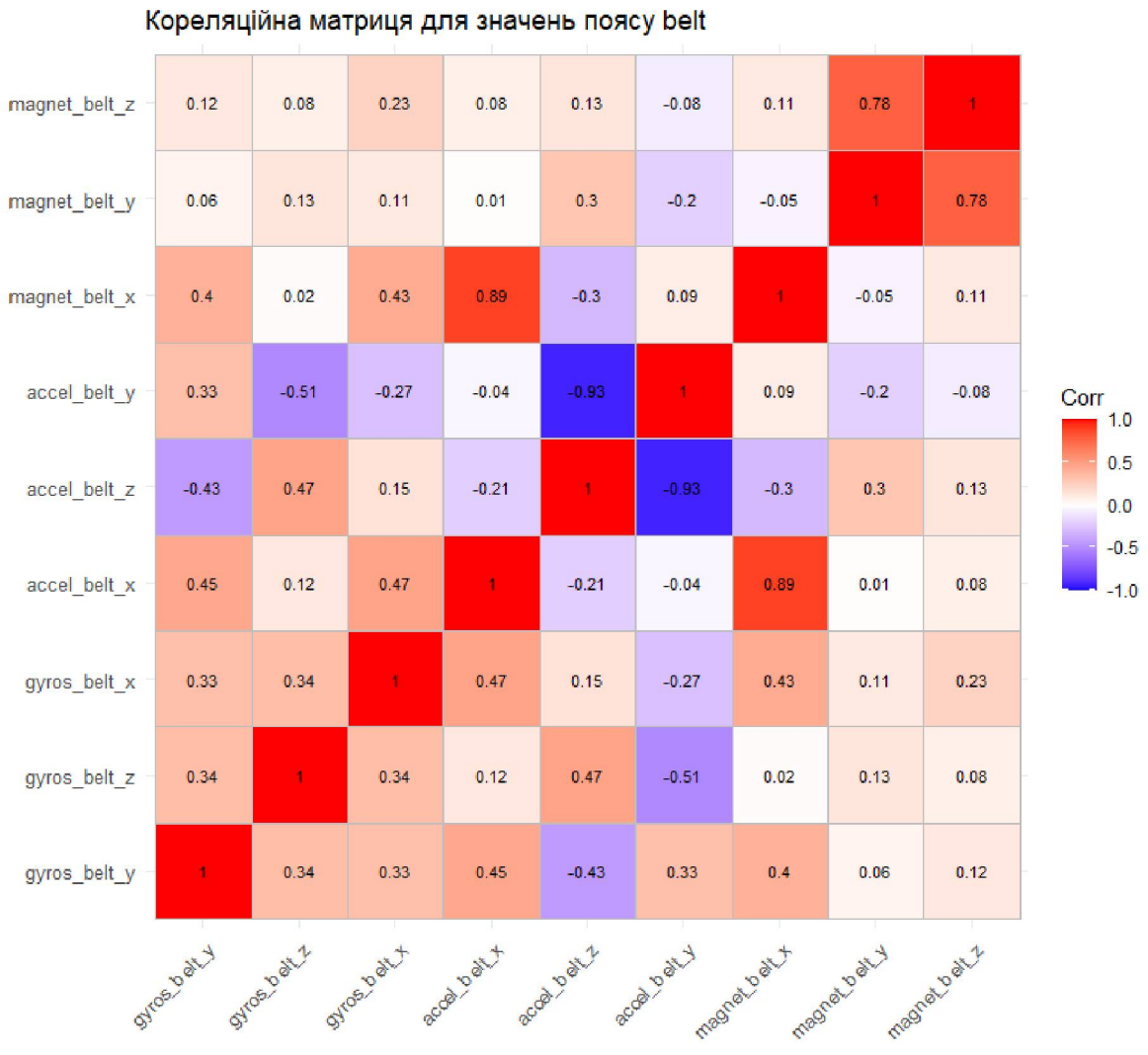


Рисунок 3.5 – Кореляційна матриця

Відомо наступне:

1. Висока взаємозалежність між деякими змінними:

- **accel_belt_y** та **accel_belt_z** (-0.93): Сильна негативна кореляція свідчить про те, що ці змінні мають протилежні напрямки зміни. Це може вказувати на специфічні фізичні взаємодії або положення поясу, коли одна змінна збільшується, а інша зменшується.
- **magnet_belt_x** та **accel_belt_x** (0.89): Сильна позитивна кореляція свідчить про те, що ці змінні змінюються в одному напрямку. Це може бути ознакою того, що магнітні і акселерометричні вимірювання на осі X тісно пов'язані.

2. Помірна взаємозалежність між іншими змінними:

- **magnet_belt_y** та **magnet_belt_z** (0.78): Позитивна кореляція вказує на те, що магнітні вимірювання на осях Y та Z змінюються разом. Це може бути корисно для розуміння орієнтації поясу.
- **accel_belt_z** та **gyros_belt_z** (0.47): Помірна позитивна кореляція вказує на те, що акселерометричні та гіроскопічні вимірювання на осі Z можуть бути взаємопов'язані, що може вказувати на специфічні рухи поясу.
- **accel_belt_x** та **gyros_belt_y** (0.45): Помірна позитивна кореляція вказує на те, що акселерометричні вимірювання на осі X та гіроскопічні на осі Y змінюються разом.

3. Можливі конфлікти між змінними:

- **accel_belt_y** та **gyros_belt_z** (-0.51): Негативна кореляція вказує на те, що коли значення однієї змінної зростає, інша зменшується. Це може вказувати на те, що ці вимірювання можуть відображати різні аспекти руху або положення поясу.

4. Слабка або відсутня кореляція між багатьма змінними:

- Багато змінних мають кореляцію близьку до нуля, що вказує на відсутність лінійної залежності між ними. Це означає, що ці змінні можуть не мати

прямого впливу одна на одну і можуть бути використані незалежно у моделюванні.

5. Важливість у моделюванні:

- Високі значення кореляції між змінними можуть свідчити про мультиколінеарність, що може вплинути на моделі машинного навчання. Необхідно враховувати це під час вибору змінних для моделювання.

3.6. **Моделі «Random Forest»**

Отже, враховуючи вищесказане, були створені три моделі: перша модель загальна, для всіх змінних усього датасету «pml-training.csv», друга ж модель використовує, для навчання, змінні, пов’язані з датчиком поясу. Також присутня третя модель, яка відокремлена змінними лише про руки. Оцінка їхньої точності:

3.6.1. Модель №1. «Загальна»

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1116    4    0    0    0
         B    0  755    2    0    0
         C    0    0  682    2    0
         D    0    0    0  641    5
         E    0    0    0    0  716

Overall Statistics

               Accuracy : 0.9967
                 95% CI : (0.9943, 0.9982)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9958

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            1.0000   0.9947   0.9971   0.9969   0.9931
Specificity            0.9986   0.9994   0.9994   0.9985   1.0000
Pos Pred Value         0.9964   0.9974   0.9971   0.9923   1.0000
Neg Pred Value         1.0000   0.9987   0.9994   0.9994   0.9984
Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
Detection Rate         0.2845   0.1925   0.1738   0.1634   0.1825
Detection Prevalence   0.2855   0.1930   0.1744   0.1647   0.1825
Balanced Accuracy      0.9993   0.9970   0.9982   0.9977   0.9965
```

Рисунок 3.6 – Загальна точність та інші показники для загальної моделі

Загальні висновки:

1. Висока точність моделі:

- Загальна точність моделі складає 99.67%, що є дуже високим показником. Це означає, що модель правильно класифікує більшість випадків.
- 95% довірчий інтервал для точності становить від 99.43% до 99.82%, що підтверджує стабільну високу точність моделі.

2. Kappa коефіцієнт: значення становить 0.9958, що вказує на майже ідеальну угоду між передбаченими та фактичними класами після усунення випадкових угод.

3. Висновки за класами: всі класи мають дуже високі показники за чутливістю, специфічністю та позитивною прогностичною значущістю (від 99%), це означає що модель здатна до правильної ідентифікації майже всіх випадків класу, отже модель випадкового лісу показує відмінні результати класифікації для всіх класів із дуже високою точністю та мінімальною кількістю помилок. Це свідчить про те, що модель добре навчена і може бути ефективно використана для подальшого аналізу і прогнозування на схожих наборах даних.

3.6.2. Модель №2. «belt»

Оскільки вже відомо, що всі змінні за поясом мають дуже велику вагу в побудуванні моделі, можна очікувати досить великі показники точності.

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1042   27   37   30   14
         B   26  695   26    4    1
         C   14   30  589   10    6
         D   27    6   32  594    4
         E    7    1    0    5  696

Overall Statistics

               Accuracy : 0.9217
                 95% CI : (0.9129, 0.93)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9009

 Mcnemar's Test P-Value : 0.000543

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9337   0.9157   0.8611   0.9238   0.9653
Specificity            0.9615   0.9820   0.9815   0.9790   0.9959
Pos Pred Value         0.9061   0.9242   0.9076   0.8959   0.9817
Neg Pred Value         0.9733   0.9798   0.9710   0.9850   0.9922
Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
Detection Rate         0.2656   0.1772   0.1501   0.1514   0.1774
Detection Prevalence   0.2931   0.1917   0.1654   0.1690   0.1807
Balanced Accuracy      0.9476   0.9488   0.9213   0.9514   0.9806
```

Рисунок 3.7 – Загальна точність та інші показники для моделі «belt»

Як видно з рисунку – результат є досить очікуваним, оскільки, дійсно, змінні по датчикам поясу є доволі значимі для побудови моделі. Окрім цього також слід зазначити, що результат трішки гірший ніж у загальному випадку, відбувається це через те що вибірка йдеться саме і тільки по даним поясу – загальна оцінка точності занижується, це пов'язано з тим що «важливість» інших змінних також впливає на результат та фінальну оцінку всієї вправи, але результат точності для вибірки інших окремих змінних був би ще гіршим саме через цю проблематику.

3.6.3. Модель №3. «arm»

Відомо, що показник точності змінних «belt» є найважливішими у побудові моделі, доказ цього буде представлений побудовою моделі для поясу, щоб дізнатись їх точність:

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1052   46    4    2    2
         B   30  656   61   15   14
         C   20   26  581   36   12
         D    9   18   25  563   43
         E    5   13   13   27  650

Overall Statistics

               Accuracy : 0.8927
                 95% CI : (0.8826, 0.9022)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8643

 Mcnemar's Test P-Value : 1.803e-05

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9427   0.8643   0.8494   0.8756   0.9015
Specificity            0.9808   0.9621   0.9710   0.9710   0.9819
Pos Pred Value         0.9512   0.8454   0.8607   0.8556   0.9181
Neg Pred Value         0.9773   0.9673   0.9683   0.9755   0.9779
Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
Detection Rate         0.2682   0.1672   0.1481   0.1435   0.1657
Detection Prevalence   0.2819   0.1978   0.1721   0.1677   0.1805
Balanced Accuracy      0.9617   0.9132   0.9102   0.9233   0.9417
```

Рисунок 3.8 – Загальна точність та інші показники для моделі «arm»

Дійсно, результат «accuracy» (точність) є 89%, отже становиться зрозумілим те, що найліпшим вибором між моделями, за частинами тіла, буде модель №2, але загальна модель має найліпшу точність.

3.7. **Результати**

Надалі треба дійти до присвоєння змінним з тестового датасету певний відповідний ранг (Від A до E). Програмна реалізація виконає «очікувану» оцінку для кожного з 20 зразків (problem_id: 1-20):

	problem_id	predicted_classe
1	1	B
2	2	A
3	3	B
4	4	A
5	5	A
6	6	E
7	7	D
8	8	B
9	9	A
10	10	A
11	11	B
12	12	C
13	13	B
14	14	A
15	15	E
16	16	E
17	17	A
18	18	B
19	19	B
20	20	B

Рисунок 3.9 – Очікувана оцінка для моделі «Загальна»

	problem_id	predicted_classe
1	1	B
2	2	A
3	3	B
4	4	A
5	5	A
6	6	E
7	7	D
8	8	B
9	9	A
10	10	A
11	11	C
12	12	C
13	13	B
14	14	A
15	15	E
16	16	E
17	17	A
18	18	B
19	19	B
20	20	B

Рисунок 3.10 – Очікувана оцінка для моделі «belt»

	problem_id	predicted_classe
1	1	B
2	2	B
3	3	A
4	4	A
5	5	A
6	6	E
7	7	D
8	8	B
9	9	A
10	10	A
11	11	B
12	12	C
13	13	B
14	14	A
15	15	E
16	16	E
17	17	A
18	18	B
19	19	B
20	20	B

Рисунок 3.11 – Очікувана оцінка для моделі «arm»

У тестовому датасеті є 20 проблемних зразків, які не мають рангів, отже на основі навчених моделях можна зробити очікувану оцінку за змінними, що були записані в ці моделі. Як вже зазначалось раніше, чим точніше модель – тим ліпше вона може робити оцінку. Через це спостерігається різниця в оцінках.

Датасет «pml-testing.csv» не має результуючої змінної – Реальні значення класів (або істинні значення класів): це фактичні мітки або категорії, до яких належать дані у вашому тестовому наборі. Вони використовуються для порівняння з прогнозованими значеннями, щоб оцінити точність моделі. Наприклад, коли йде спроба класифікувати типи фізичних вправ (A, B, C, D, E), реальні значення класів – це правильні типи вправ для кожного зразка в наборі даних, які були зібрані та перевірені раніше. Однак все ще можливо побудувати прогнозовані значення для тестового набору, щоб дізнатися, які класи модель прогнозує для кожного problem_id. Результат наданий в таблиці 3.2.

Важливо відзначити, зміни в очікуваній оцінці ранга між моделями з різними вибірками змінних обумовлені відмінностями в інформативності, варіабельності, шумі, мультиколінеарності та адаптації моделі до певних змінних. Але чітко зрозуміло що найближча до «загальної» моделі є модель «belt» з мінімальними змінами в оцінці.

Підводячи підсумок: прогнозовані значення для тестового набору зроблені, відповідають точності. Моделі «Загальна» та «belt» є добре навченими та мають гарну стабільність з надійністю у класифікації рангу.

Таблиця 3.1

Вихідні дані загальної точності для кожної моделі

Назва моделі	Точність
Загальна	0.9967
belt	0.9217
arm	0.8927

Таблиця 3.2

Вихідні дані очікуваної оцінки для кожної моделі

ID	Оцінка (Загальна)	Оцінка (belt)	Оцінка (arm)
1	B	B	B
2	A	A	B
3	B	B	A
4	A	A	A
5	A	A	A
6	E	E	E
7	D	D	D
8	B	B	B
9	A	A	A
10	A	A	A
11	B	C	B
12	C	C	C
13	B	B	B
14	A	A	A
15	E	E	E
16	E	E	E
17	A	A	A
18	B	B	B
19	B	B	B
20	B	B	B

Висновки за розділом 3

У Розділі 3 виконано аналіз існуючого програмного забезпечення для вирішення задачі класифікації біосигналів, визначено межі придатності окремих

програмних комплексів. Зокрема, розглянуто інструментальні засоби, такі як MATLAB, Python, WEKA, Java, Julia, R. Проведено розвідувальний аналіз даних, включаючи опис змінних та статистичний аналіз. Здійснено обробку даних, яка включала обробку пропущених значень та скорочення кількості змінних з 160 до 53. Виконано аналіз статистичних зв'язків між змінними, зокрема виявлено високу взаємозалежність між деякими змінними поясу (belt). Побудовано три моделі класифікації на основі Random Forest: загальна модель, модель із змінними поясу (belt), і модель із змінними руки (arm). Найвищу точність продемонструвала загальна модель (99.67%), проте модель із змінними поясу також показала високу точність та стабільність у класифікації.

ВИСНОВКИ

У даній роботі виконано аналіз актуальності та перспективності використання методів машинного навчання для аналізу біосигналів у медичних дослідженнях та практиці. Результати досліджень підтверджують ефективність сучасних алгоритмів у забезпеченні високої точності та швидкості обробки медичних даних, біосигналів та датчиків, що сприяє покращенню якості діагностики. В роботі продемонстровано значний потенціал використання методів машинного навчання для аналізу біосигналів, що сприяє розвитку персоналізованої медицини та покращенню якості медичних послуг. Отримані результати підтверджують ефективність запропонованих підходів та моделей. Розглянуто існуючи алгоритми та програмні комплекси. Оглянуто які є найважливіші дані в датасеті. Виконано програмну реалізацію, навчальний набір даних було поділено на навчальну (80%) та тестову (20%) вибірки, попередньо оброблені за пропущеними значеннями та відфільтровані для кращої точності. Розроблено три моделі машинного навчання методом випадкового лісу за допомогою матриці невідповідності. Виконане прогнозування на тестовій вибірці показало високу точність моделі. Також виконане прогнозування для повного тестового набору даних з метою оцінки моделі на нових даних – результати прогнозування були збережені у вигляді таблиці з ідентифікаторами проблем та прогнозованими класами. Завдяки проведеним діям та аналізу було отримано високоточну модель для класифікації фізичних вправ на основі даних із датчиків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Review on Wearable Technology in Sports: Concepts, Challenges and Opportunities. [Електронний ресурс]. – режим доступу: URL: https://www.mdpi.com/2076-3417/13/18/10399 (дата звернення - 04.05.2024)

2. Filter method for feature selection. [Електронний ресурс]. – режим доступу: URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0291070 (дата звернення - 04.05.2024)

3. Реабілітація. [Електронний ресурс]. – режим доступу: URL: https://shorturl.at/44OiR

4. A Novel Machine Learning-Based Prediction Method for Early Detection and Diagnosis of Congenital Heart Disease Using ECG Signal Processing. [Електронний ресурс]. – режим доступу: URL: https://www.mdpi.com/2227-7080/12/1/4 (дата звернення - 06.05.2024)

5. Machine learning in biosignals processing for mental health: A narrative review. [Електронний ресурс]. – режим доступу: URL: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2022.1066317/full (дата звернення - 10.05.2024)

6. Bio-Signals in Medical Applications and Challenges Using Artificial Intelligence. [Електронний ресурс]. – режим доступу: URL: https://www.mdpi.com/2224-2708/11/1/17 (дата звернення - 13.05.2024)

7. Biological Signal. [Електронний ресурс]. – режим доступу: URL: https://www.sciencedirect.com/topics/engineering/biological-signal (дата звернення - 19.05.2024)

8. Logistic Regression for Machine Learning in Process Tomography. [Електронний ресурс]. – режим доступу: URL: https://www.mdpi.com/1424-8220/19/15/3400 (дата звернення - 23.05.2024)

9. Support Vector Machine. [Електронний ресурс]. – режим доступу: URL: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-10-16 (дата звернення - 21.05.2024)

10. Recent advances in decision trees: an updated survey. [Електронний ресурс]. – режим доступу: URL: https://link.springer.com/article/10.1007/s10462-022-10275-5 (дата звернення - 26.05.2024)

11. Aurélien Géron. Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. 2019p. 197c.

12. Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13). Stuttgart, Germany: ACM SIGCHI, 2013. [Електронний ресурс]. – режим доступу: URL: https://www.kaggle.com/datasets/prashant111/weight-lifting-exercises (дата звернення - 26.01.2024)

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **бакалавр**
Галузь знань: **12 – Інформаційні технології**
Спеціальність: **123 – Комп'ютерна інженерія.**

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної
та прикладної системотехніки
_____д.т.н., проф. Шматков С. І.
«21» грудня 2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Варосяна Арсена Левоновича

(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Комп'ютерне опрацювання біосигналів на основі методів машинного навчання»**

керівник роботи Бакуменко Ніна Станіславівна, к.т.н, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» травня 2024року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити
   1) Постановка задачі класифікації станів об'єктів за допомогою аналізу біосигналів.
   2) Аналіз існуючих методів класифікації об'єктів.
   3) Вибір та обґрунтування методу класифікації об'єктів.
   4) Розробка математичної моделі вирішення задачі класифікації.
   5) Розробка програмно-алгоритмічної моделі системи класифікації станів об'єктів за допомогою аналізу біосигналів.
   6) Тестування моделі та аналіз отриманих результатів.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Підбір наукової літератури	21.12.2023 - 25.01.2024
2	Огляд сучасних методів класифікації об'єктів	19.12.2023 - 2.01.2024
3	Аналіз інструментальних засобів для вирішення задач машинного навчання методів для класифікації біосигналів	2.01.2024 - 2.02.2024
4	Розробка комп'ютерної моделі аналізу біосигналів	2.01.2024 - 2.02.2024
5	Тестування та апробація розробленої системи	3.02.2024 - 30.03.2024
6	Корегування моделі після тестування	31.03.2024 - 29.04.2024
7	Оформлення пояснювальної записки	3.03.2024 - 30.04.2024
8	Представлення дипломного проекту керівнику дипломної роботи та рецензенту.	31.03.2024 - 27.05.2024

5. Дата видачі завдання 21.12.2023

Студент А. Л. Варосян
ініціали, прізвище


підпис

Керівник роботи Н. С. Бакуменко
ініціали, прізвище


підпис

Додаток Б

Затверджую

_____

«_____»_____2023 р.

Технічне завдання

на розробку прототипу «Комп'ютерне опрацювання біосигналів на основі методів машинного навчання».

1.	Введення	1.1 Комп'ютерне опрацювання біосигналів на основі методів машинного навчання. 1.2. Галузь застосування: Інформаційні технології
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп’ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/909 від «03» травня 2024 року (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).

3.	Призначення розробки	3.1. Мета розробки: Створення системи обробки біосигналів на основі методів машинного навчання для підвищення точності та ефективності медичних діагностичних систем. 3.2. Призначення розробки: Забезпечення можливості ефективної обробки біосигналів за допомогою машинного навчання для підтримки клінічних рішень, покращення діагностичних висновків та оптимізації лікувальних процедур. 3.3. Вхідні дані: Вхідними даними є біосигнали (наприклад, ЕКГ, ЕЕГ), які отримуються від пацієнтів у реальному часі або з баз даних медичних записів. 3.4. Вихідні дані розробки: Результатом розробки є моделі машинного навчання, які можуть ефективно обробляти та аналізувати біосигнали для надання точних медичних діагнозів та рекомендацій.

4.	Технічні вимоги до програмного виробу	4.1. Функціональні вимоги: - Здатність зчитування та аналізу різних типів біосигналів за допомогою інтеграції з медичними пристроями та сенсорами. - Забезпечення точності та надійності обробки даних біосигналів через використання передових алгоритмів машинного навчання. 4.2. Нефункціональні вимоги: - Висока швидкість обробки біосигналів для забезпечення реального часу відгуків та аналізу. - Масштабованість системи для ефективної роботи з великою кількістю одночасних користувачів та потоків даних. - Інтуїтивно зрозумілий інтерфейс для користувачів різного рівня кваліфікації. 4.3. Вимоги до інтеграції: - Сумісність з різними медичними датчиками та системами для забезпечення універсальності використання. - Сумісність з різними типами пристроїв та операційних систем. - Гнучкість в інтеграції з іншими аналітичними платформами та системами зберігання даних.

		- Підтримка різноманітних форматів даних, що отримуються з медичних датчиків. - 4.4. Вимоги до безпеки: - Захист даних біосигналів від несанкціонованого доступу і зловмисних атак з використанням сучасних методів шифрування та аутентифікації. - Забезпечення конфіденційності користувацької інформації згідно з міжнародними стандартами безпеки даних. - Система виявлення та реагування на аномалії в процесі обробки даних для підвищення надійності роботи системи.

5.	Вимоги до програмної документації	Документацією до роботи з темою «Комп'ютерне опрацювання біосигналів на основі методів машинного навчання» вважати: 1) Опис основних вимог та функціональності системи опрацювання біосигналів (представити в розділі 4 пояснювальної записки до кваліфікаційної роботи). 2) Програму і методику випробувань розробленого алгоритму опрацювання біосигналів (представити як додаток В до пояснювальної записки до кваліфікаційної роботи). 3) Опис розробленого прототипу системи для обробки біосигналів (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).
6.	Вимоги до технічно-аналітичних показників	Документацією до проєкту "Комп'ютерне опрацювання біосигналів на основі методів машинного навчання" вважати: 1) Справжнє технічне завдання на розробку системи аналізу біосигналів (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи). 2) Опис розробленої системи аналізу біосигналів (представити в розділі 3

		пояснювальної записки до кваліфікаційної роботи). 3) Джерела базової інформації.	
7.		Дата	Назва етапу

	Стадії і етапи розробки	21.12.2023 - 25.01.2024	Підбір наукової літератури.
		19.12.2023 - 2.01.2024	Огляд сучасних методів класифікації об’єктів.
		2.01.2024 - 2.02.2024	Аналіз інструментальних засобів для вирішення задач машинного навчання методів для класифікації біосигналів.
		2.01.2024 - 2.02.2024	Розробка комп’ютерної моделі аналізу біосигналів.
		3.02.2024 - 30.03.2024	Тестування та апробація розробленої системи.
		31.03.2024 - 29.04.2024	Корегування моделі після тестування.
		3.03.2024 - 30.04.2024	Оформлення пояснювальної записки.
		31.03.2024 - 27.05.2024	Представлення дипломного проекту керівнику дипломної роботи та рецензенту.

8.	Порядок контролю і приймання програмного продукту (моделі)	1. Перевірку ходу розробки виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в примірнику на CD-R компакт-диску.

Виконавець

Студент групи КІ-41

Варосян А. Л.

підпис

Керівник

д. техн. наук

Бакуменко Н. С.

підпис

Додаток В

Програма і методика випробувань програмного виробу

«Комп'ютерне опрацювання біосигналів на основі методів машинного навчання»

1. Об’єкт випробувань

1. Назва розробленого прототипу: «Комп'ютерне опрацювання біосигналів на основі методів машинного навчання».

2. Галузь застосування: 12 – Інформаційні технології

3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональні можливості системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

1. Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп’ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавці та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

4.1. Функціональні вимоги:

- Здатність завантажувати та обробляти набір даних, що містять інформацію про фізичні вправи, виконувані людьми.
- Підтримка різноманітних алгоритмів машинного навчання для аналізу біосигналів.

4.2. Нефункціональні вимоги:

- Висока продуктивність системи з можливістю ефективної обробки великих обсягів даних.
- Простота в управлінні та налаштуванні для забезпечення легкості використання та підтримки.

4.3. Вимоги до інтеграції:

- Можливість інтеграції з додатковими пакетами та бібліотеками в R для розширення функціональності.
- Сумісність з різними версіями RStudio та підтримуваними операційними системами.

4.4. Вимоги до безпеки

- Захист даних при зберіганні та обробці, включно з механізмами шифрування, якщо потрібно.
- Забезпечення конфіденційності, цілісності та доступності інформації.

4.5. Вимоги до моніторингу та звітності:

- Можливість моніторингу процесу обробки даних та оцінки ефективності використаних моделей машинного навчання.
- Наявність звітів про результати обробки та аналізу даних.

5. Вимоги до програмної документації

Програмною документацією до прототипу моделі «Комп'ютерне опрацювання біосигналів на основі методів машинного навчання» вважати:

1) Справжнє Технічне завдання на розробку прототипу моделі (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

2) Опис реалізованого прототипу моделі (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).

3) Джерела базової інформації.

6. Засоби і порядок випробувань

6.1. Засоби випробувань

Засоби випробувань представлено на ПК на яких встановлено наступні програмні засоби: RStudio та необхідні пакети для обробки біосигналів.

6.2. Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

– Ознайомчий (1-й етап);

– Власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1) Перевірка наявності та повноти даних у .csv файлі, що відповідають вимогам до даних, вказаним у технічному завданні.

2) Перевірка якості документації. Перевірку здійснювати за критерієм відповідності вимогам ГОСТ 34.602-89 "Автоматизовані системи. Система стандартів на автоматизовані системи. Зміст і структура технічного завдання"

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

Методика проведення перевірок:

a) Запустити RStudio та завантажити набір даних.

b) Виконати предобробку даних:

- Переконатися, що дані очищені від шумів та аномалій.
- Перевірити відповідність даних умовам задачі.

c) Реалізувати алгоритми машинного навчання:

- Перевірити точність моделі на відомих даних.
- Виконати валідацію моделі за допомогою крос-валідації або іншої техніки.

d) Провести тестування продуктивності:

- Запустити алгоритми на великому наборі даних
- Виміряти час обробки даних та швидкість відгуку моделі

e) Перевірити стабільність та надійність моделі:

- Симулювати різні умови роботи (наприклад, зміни в біосигналах)
- Переконатися, що модель стійка до невеликих змін у вхідних даних.

f) Провести фінальне тестування:

- Забезпечити, що всі функціональні можливості системи працюють коректно.
- Перевірити сумісність та інтеграцію моделі з іншими компонентами системи

g) Якщо всі перевірки пройшли успішно, модель вважається такою, що пройшла випробування.

Тест 1.

1. Перевірка .csv файлів

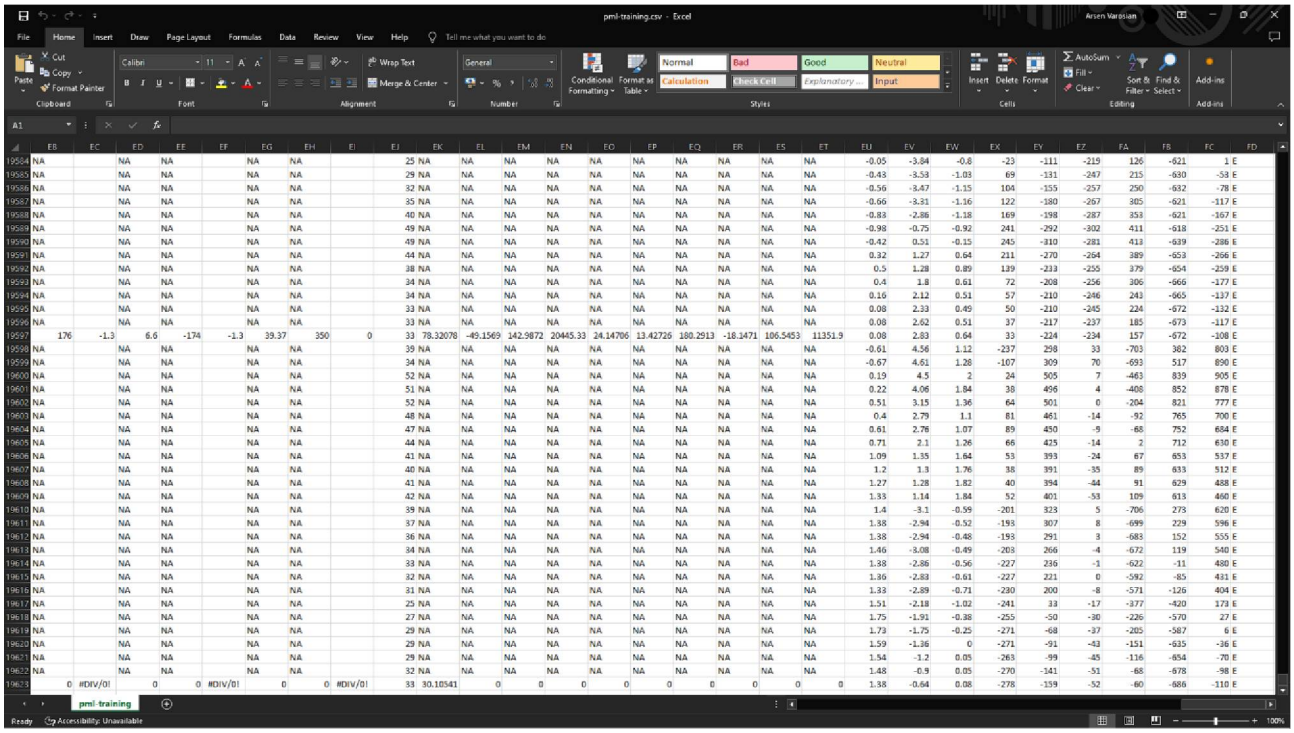


Рис. В.1 pml-training.csv

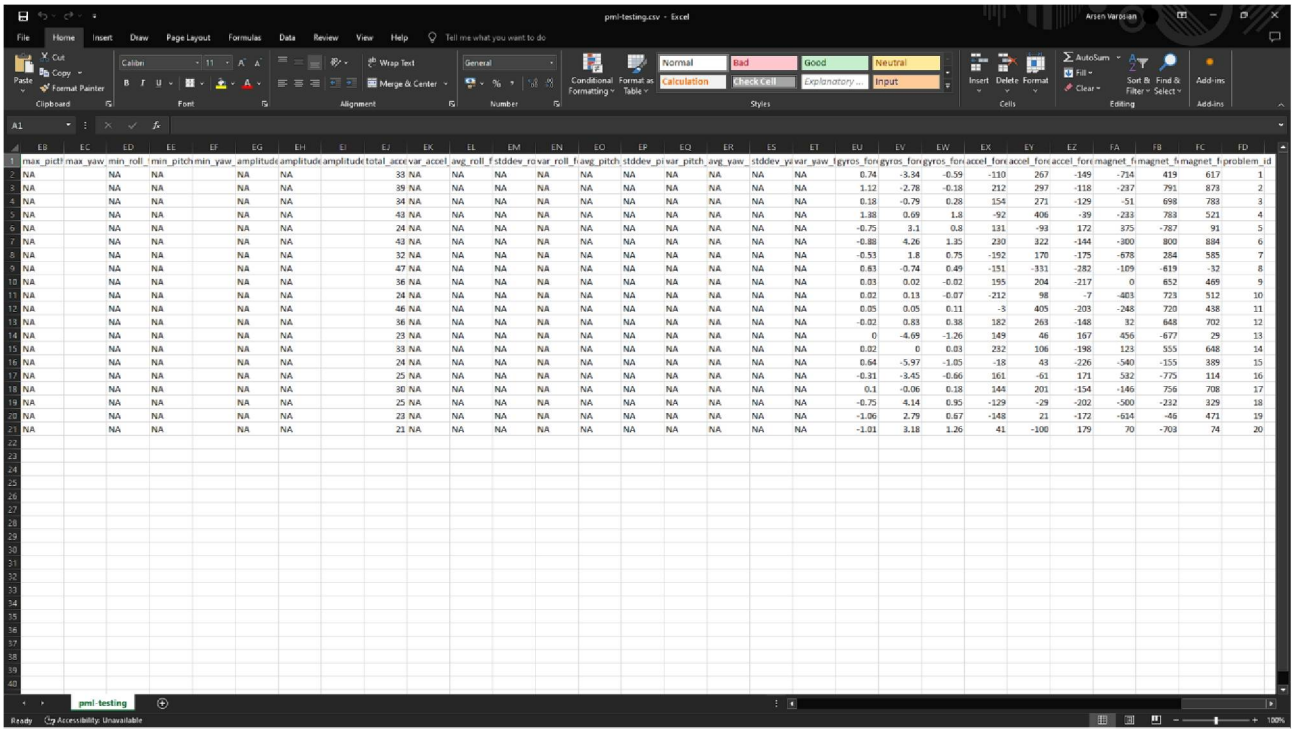


Рис. В.1 pml-testing.csv

Тест 2.

c) Реалізувати алгоритми машинного навчання:

d) Провести тестування продуктивності:

e) Перевірити стабільність та надійність моделі:

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1052   46    4    2    2
         B   30  656   61   15   14
         C   20   26  581   36   12
         D    9   18   25  563   43
         E    5   13   13   27  650

Overall Statistics

               Accuracy : 0.8927
                 95% CI : (0.8826, 0.9022)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8643

 Mcnemar's Test P-Value : 1.803e-05

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9427   0.8643   0.8494   0.8756   0.9015
Specificity            0.9808   0.9621   0.9710   0.9710   0.9819
Pos Pred Value         0.9512   0.8454   0.8607   0.8556   0.9181
Neg Pred Value         0.9773   0.9673   0.9683   0.9755   0.9779
Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
Detection Rate         0.2682   0.1672   0.1481   0.1435   0.1657
Detection Prevalence   0.2819   0.1978   0.1721   0.1677   0.1805
Balanced Accuracy      0.9617   0.9132   0.9102   0.9233   0.9417
```

Рис В.2. arm

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1042   27   37   30   14
         B   26  695   26    4    1
         C   14   30  589   10    6
         D   27    6   32  594    4
         E    7    1    0    5  696

Overall Statistics

               Accuracy : 0.9217
                 95% CI : (0.9129, 0.93)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9009

 Mcnemar's Test P-Value : 0.000543

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9337   0.9157   0.8611   0.9238   0.9653
Specificity            0.9615   0.9820   0.9815   0.9790   0.9959
Pos Pred Value         0.9061   0.9242   0.9076   0.8959   0.9817
Neg Pred Value         0.9733   0.9798   0.9710   0.9850   0.9922
Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
Detection Rate         0.2656   0.1772   0.1501   0.1514   0.1774
Detection Prevalence   0.2931   0.1917   0.1654   0.1690   0.1807
Balanced Accuracy      0.9476   0.9488   0.9213   0.9514   0.9806
```

Рис В.2. belt

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1052   46    4    2    2
         B   30  656   61   15   14
         C   20   26  581   36   12
         D    9   18   25  563   43
         E    5   13   13   27  650

Overall Statistics

               Accuracy : 0.8927
                 95% CI : (0.8826, 0.9022)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8643

 Mcnemar's Test P-Value : 1.803e-05

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9427   0.8643   0.8494   0.8756   0.9015
Specificity            0.9808   0.9621   0.9710   0.9710   0.9819
Pos Pred Value         0.9512   0.8454   0.8607   0.8556   0.9181
Neg Pred Value         0.9773   0.9673   0.9683   0.9755   0.9779
Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
Detection Rate         0.2682   0.1672   0.1481   0.1435   0.1657
Detection Prevalence   0.2819   0.1978   0.1721   0.1677   0.1805
Balanced Accuracy      0.9617   0.9132   0.9102   0.9233   0.9417
```

Рис В.2. Загальна

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КІ-41, Варосян А. Л.

Додаток Г

```
# Завантаження необхідних бібліотек

library(caret)

library(randomForest)

library(ggplot2)

library(plotly)

library(reshape2)

library(dplyr)

library(ggcorrplot)


# Завантаження даних

train_data <- read.csv("pml-training.csv")

test_data <- read.csv("pml-testing.csv")


# Перевірка структури даних

str(train_data)

str(test_data)


# Видалення стовпчиків з пропущеними значеннями (якщо такі є)

train_data <- train_data %>% select_if(~ !any(is.na(.)))

test_data <- test_data %>% select_if(~ !any(is.na(.)))


# Перетворення змінної "classe" на фактор у навчальному наборі

train_data$classe <- as.factor(train_data$classe)
```

```
# Вибір необхідних змінних для моделі, включаючи user_name
selected_vars <- c("roll_belt", "pitch_belt", "yaw_belt", "total_accel_belt",
                   "gyros_belt_y", "gyros_belt_z", "accel_belt_x", "accel_belt_z",
                   "magnet_belt_x", "magnet_belt_y", "magnet_belt_z", "roll_arm",
                   "pitch_arm", "yaw_arm", "total_accel_arm", "gyros_arm_x",
                   "gyros_arm_y", "gyros_arm_z", "accel_arm_x", "accel_arm_y",
                   "accel_arm_z", "magnet_arm_x", "magnet_arm_y", "magnet_arm_z",
                   "roll_dumbbell", "pitch_dumbbell", "yaw_dumbbell",
                   "total_accel_dumbbell", "gyros_dumbbell_x", "gyros_dumbbell_y",
                   "gyros_dumbbell_z", "accel_dumbbell_y", "accel_dumbbell_z",
                   "magnet_dumbbell_x", "magnet_dumbbell_y", "magnet_dumbbell_z",
                   "roll_forearm", "pitch_forearm", "yaw_forearm",
                   "total_accel_forearm", "gyros_forearm_x", "gyros_forearm_y",
                   "gyros_forearm_z", "accel_forearm_x", "accel_forearm_y",
                   "accel_forearm_z", "magnet_forearm_x", "magnet_forearm_y",
                   "magnet_forearm_z", "classe", "user_name", "accel_belt_y",
"gyros_belt_x")

# Фільтрування даних
train_data <- train_data %>% select(all_of(selected_vars))
test_data <- test_data %>% select(all_of(c(setdiff(selected_vars, "classe"),
"problem_id", "user_name")))

# Поділ даних на навчальну (80%) та тестову (20%) вибірки
```

```
set.seed(123)

trainIndex <- createDataPartition(train_data$classe, p = 0.80, list = FALSE)

trainData <- train_data[trainIndex, ]

testData <- train_data[-trainIndex, ]


# Перевірка структури навчальної вибірки

str(trainData)


# Побудова моделі машинного навчання (random forest)

model <- randomForest(classe ~ . - user_name, data = trainData)


# Прогнозування на тестовій вибірці

predictions <- predict(model, testData)


# Оцінка точності моделі

confMatrix <- confusionMatrix(predictions, testData$classe)


# Відображення матриці невідповідності

print(confMatrix)


# Прогнозування для повного тестового набору

test_predictions <- predict(model, test_data)


# Відображення результатів для тестового набору
```

```
test_results <- data.frame(problem_id = test_data$problem_id, predicted_classe =
test_predictions)

print(test_results)


# Побудова графіка важливості змінних

importance <- importance(model)

varImportance <- data.frame(Variable = rownames(importance), Importance = importance[,
1])


ggplot(varImportance, aes(x = reorder(Variable, Importance), y = Importance)) +

  geom_bar(stat = "identity") +

  coord_flip() +

  xlab("Змінні") +

  ylab("Важливість") +

  ggtitle("Важливість змінних у моделі Random Forest")


# Побудова матриці невідповідності у вигляді графіка

confMatrixPlot <- as.data.frame(confMatrix$table)

ggplot(confMatrixPlot, aes(x = Reference, y = Prediction, fill = Freq)) +

  geom_tile() +

  geom_text(aes(label = Freq), vjust = 1) +

  scale_fill_gradient(low = "white", high = "blue") +

  xlab("Реальні значення") +

  ylab("Прогнозовані значення") +

  ggtitle("Матриця невідповідності")
```

```
# Побудова моделі машинного навчання тільки для змінних, пов'язаних з поясом
selected_belt_vars <- c("roll_belt", "pitch_belt", "yaw_belt", "total_accel_belt",
                        "gyros_belt_x", "gyros_belt_y", "gyros_belt_z",
                        "accel_belt_x", "accel_belt_y", "accel_belt_z",
                        "magnet_belt_x", "magnet_belt_y", "magnet_belt_z", "classe",
"user_name")

# Фільтрування даних для нової вибірки (belt)
train_data_belt <- train_data %>% select(all_of(selected_belt_vars))
test_data_belt <- test_data %>% select(all_of(c(setdiff(selected_belt_vars, "classe"),
"problem_id", "user_name")))

# Поділ даних на навчальну (80%) та тестову (20%) вибірки для нової вибірки (belt)
set.seed(123)
trainIndex_belt <- createDataPartition(train_data_belt$classe, p = 0.80, list = FALSE)
trainData_belt <- train_data_belt[trainIndex_belt, ]
testData_belt <- train_data_belt[-trainIndex_belt, ]

# Перевірка структури навчальної вибірки для нової вибірки (belt)
str(trainData_belt)

# Побудова моделі машинного навчання (random forest) для нової вибірки (belt)
model_belt <- randomForest(classe ~ . - user_name, data = trainData_belt)
```

```
# Прогнозування на тестовій вибірці для нової вибірки (belt)

predictions_belt <- predict(model_belt, testData_belt)


# Оцінка точності моделі для нової вибірки

confMatrix_belt <- confusionMatrix(predictions_belt, testData_belt$classe)


# Відображення матриці невідповідності для нової вибірки (belt)

print(confMatrix_belt)


# Прогнозування для повного тестового набору для нової вибірки (belt)

test_predictions_belt <- predict(model_belt, test_data_belt)


# Відображення результатів для тестового набору для нової вибірки (belt)

test_results_belt <- data.frame(problem_id = test_data_belt$problem_id,
predicted_classe = test_predictions_belt)

print(test_results_belt)


# Побудова кореляційної матриці для всіх значень belt

belt_vars <- train_data %>%

  select(starts_with("gyros_belt"), starts_with("accel_belt"),
starts_with("magnet_belt"))


cor_matrix <- cor(belt_vars)


ggcorrplot(cor_matrix, lab = TRUE,
```

```
            title = "Кореляційна матриця для значень поясу belt",
            tl.cex = 10,
            lab_size = 3)

# Вибір змінних, пов'язаних з рукою (arm)
selected_arm_vars <- c("roll_arm", "pitch_arm", "yaw_arm", "total_accel_arm",
                       "gyros_arm_x", "gyros_arm_y", "gyros_arm_z",
                       "accel_arm_x", "accel_arm_y", "accel_arm_z",
                       "magnet_arm_x", "magnet_arm_y", "magnet_arm_z", "classe",
"user_name")

# Фільтрування даних для нової вибірки (arm)
train_data_arm <- train_data %>% select(all_of(selected_arm_vars))
test_data_arm <- test_data %>% select(all_of(c(setdiff(selected_arm_vars, "classe"),
"problem_id", "user_name")))

# Поділ даних на навчальну (80%) та тестову (20%) вибірки для нової вибірки (arm)
set.seed(123)
trainIndex_arm <- createDataPartition(train_data_arm$classe, p = 0.80, list = FALSE)
trainData_arm <- train_data_arm[trainIndex_arm, ]
testData_arm <- train_data_arm[-trainIndex_arm, ]

# Перевірка структури навчальної вибірки для нової вибірки (arm)
str(trainData_arm)
```

```
# Побудова моделі машинного навчання (random forest) для нової вибірки (arm)
model_arm <- randomForest(classe ~ . - user_name, data = trainData_arm)

# Прогнозування на тестовій вибірці для нової вибірки (arm)
predictions_arm <- predict(model_arm, testData_arm)

# Оцінка точності моделі для нової вибірки
confMatrix_arm <- confusionMatrix(predictions_arm, testData_arm$classe)

# Відображення матриці невідповідності для нової вибірки (arm)
print(confMatrix_arm)

# Прогнозування для повного тестового набору для нової вибірки (arm)
test_predictions_arm <- predict(model_arm, test_data_arm)

# Відображення результатів для тестового набору для нової вибірки (arm)
test_results_arm <- data.frame(problem_id = test_data_arm$problem_id, predicted_classe
= test_predictions_arm)
print(test_results_arm)
```