ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методики оцінювання стану пацієнтів відділення невідкладної допомоги на основі алгоритмів машинного навчання»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Андрій СПІЦИН
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-61
Андрій СПІЦИН

Керівник: Володимир САДОВЕНКО
к.ф.-м.н., доц.

Рецензент:
науковий ступінь, вчене звання
Ім’я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально–науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти – Магістр

Спеціальність - 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____ Ірина ЗАМРІЙ
«____»_____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Спіцину Андрію Ярославовичу

1. Тема кваліфікаційної роботи: «Розробка методики оцінювання стану пацієнтів відділення невідкладної допомоги на основі алгоритмів машинного навчання» керівник кваліфікаційної роботи Володимир САДОВЕНКО, к.ф.-м.н., доцент, затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 року № 320
2. Строк подання студентом роботи «17» грудня 2024 року
3. Вхідні дані до роботи:
   3.1 Інструменти розробки програмного забезпечення
   3.2 Науково-технічна література, пов’язана з системами оцінки пріоритетності пацієнтів

3.3 Науково-технічна література, пов’язана з розробкою моделей машинного навчання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1 Дослідження процесу оцінки пріоритетності пацієнтів

4.2 Аналіз систем оцінки пріоритетності пацієнтів

4.3 Аналіз методів машинного навчання

4.4 Розробка системи оцінки пріоритетності пацієнтів на основі моделі машинного навчання

4.5 Тестування розробленої системи

4.6 Висновки

5. Перелік графічного матеріалу

5.1 Мета, об’єкт, предмет дослідження

5.2 Порівняльна таблиця систем сортування

5.3 Засоби розробки

5.4 Алгоритм оцінки пріоритетності

5.5 Діаграма потоку даних в методі напівконтрольованого навчання

5.6 Діаграма процесу обробки даних

5.7 Результат обробки вхідних даних пацієнтів

5.8 Результат роботи системи

5.10 Порівняльний аналіз результатів

5.11 Висновки

5.12 Публікації та апробація роботи

6. Дата видачі завдання «16» жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз науково-технічної літератури	16.10.24 – 19.10.24	Виконано
2	Аналіз систем оцінки пріоритетності пацієнтів	21.10.24 – 23.10.24	Виконано
3	Аналіз методів машинного навчання	23.10.24 – 26.10.24	Виконано
4	Розробка моделі	28.10.24 – 9.11.24	Виконано
5	Аналіз отриманих результатів	11.11.24 – 16.11.24	Виконано
6	Оформлення роботи	18.11.24 – 23.11.24	Виконано
7	Попередній захист роботи	25.11.24	Виконано

Здобувач вищої освіти _____ Андрій СПІЦИН

(підпис)

Керівник кваліфікаційної роботи _____ Володимир САДОВЕНКО

(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 80 стор., 3 табл., 12 рис., 30 джерел.

Мета дослідження – підвищити точність та зменшити час оцінки пріоритетності пацієнтів за рахунок застосування моделі машинного навчання.

Об'єкт дослідження – процес оцінювання пріоритетності пацієнтів.

Предмет дослідження – модель машинного навчання для визначення пріоритетності пацієнтів.

Короткий зміст роботи: У роботі проаналізовано сучасні методи оцінки пріоритетності пацієнтів, їх переваги та недоліки. Проаналізовано існуючі методи машинного навчання для оцінки пріоритетності пацієнтів. Підібрано статистичні дані для навчання та тестування моделі машинного навчання. Побудовано модель оцінювання пріоритетності пацієнтів. Натреновано побудовану модель машинного навчання. Проведено тестування розробленої системи оцінки пріоритетності пацієнтів.

Ключові слова: машинне навчання, пріоритетність пацієнтів, оптимізація потоку пацієнтів, медичні технології, напівконтрольоване навчання

ABSTRACT

The text of the qualification thesis for obtaining a master's degree: 80 pages, 3 tables, 12 figures, 30 references.

Objective of the research – to improve the accuracy and reduce the time required for prioritizing patients through the application of a machine learning model.

Object of the research – the process of prioritizing patients.

Subject of the research – a machine learning model for determining patient priority.

Summary of the work: The research analyzes modern methods for prioritizing patients, their advantages, and disadvantages. Existing machine learning methods for prioritizing patients were reviewed. Statistical data were selected for training and testing the machine learning model. A model for evaluating patient priority was developed and trained. Testing of the developed patient prioritization system was conducted.

Keywords: machine learning, patient prioritization, patient flow optimization, medical technologies, semi-supervised learning.

ЗМІСТ

ВСТУП

Сектор охорони здоров'я стикається зі зростаючим тиском через збільшення кількості пацієнтів, обмеженість ресурсів та необхідність надання високоякісної медичної допомоги. Ефективне визначення пріоритетності пацієнтів має вирішальне значення для забезпечення своєчасного та належного надання медичної допомоги. Традиційні методи визначення пріоритетності пацієнтів, які часто покладаються на людське судження і статичні критерії, можуть бути суб'єктивними і неефективними.

Приймаючи рішення про пріоритетність пацієнтів, людина схильна до упередженості та помилок, що потенційно може призвести до неправильної оцінки стану пацієнта та як наслідок затримок у лікуванні термінових випадків.

Крім того лікарні, балансуючи між терміновими випадками та рутинним лікуванням, часто стикаються з проблемою розподілу ресурсів, таких як вільні лікарі, ліжка а також медикаменти. При некоректній оцінці пріоритетності пацієнтів лікарня ризикує неоптимально використовувати наявні ресурси, що знизить кількість пацієнтів, яких може обробити лікарня.

Особливо гостро такі проблеми стоять у відділенні невідкладної допомоги. Через терміновий характер відділення, некоректна оцінка пріоритетності пацієнтів може призвести до стрімкого погіршення стану пацієнтів.

Поява машинного навчання пропонує трансформаційний підхід до підвищення точності, ефективності та справедливості систем визначення пріоритетності пацієнтів. Інтеграція великих даних в охорону здоров'я дозволяє збирати та аналізувати інформацію про пацієнтів у безпрецедентних масштабах. Алгоритми машинного навчання можуть обробляти і навчатися на цих даних, виявляючи закономірності і роблячи прогнози, які виходять за межі людських можливостей.

Моделі машинного навчання можуть мінімізувати людський фактор, надаючи засновані на даних об'єктивні критерії пріоритетності, забезпечуючи більш справедливий розподіл медичної допомоги. Також вони можуть аналізувати величезні обсяги даних для більш точного прогнозування, що дозволяє краще розподіляти ресурси. Це може призвести до скорочення часу очікування, оптимального використання медичного персоналу та покращення результатів лікування пацієнтів.

Саме через ці причини дослідження з розробки моделі машинного навчання для визначення пріоритетності пацієнтів є дуже актуальним у сучасному медичному ландшафті. Це дослідження має на меті створити більш ефективну, справедливу та дієву систему визначення пріоритетності пацієнтів, що має важливе значення для вирішення сучасних проблем в охороні здоров'я.

Науковим завданням дослідження є розробка методики оцінювання терміновості та складності пацієнтів відділу невідкладної допомоги за допомогою методів машинного навчання.

Об’єктом дослідження є процес оцінювання пріоритетності пацієнтів

Предметом дослідження є методи оцінювання пріоритетності пацієнтів

Метою роботи є підвищити точність та зменшити час оцінки пріоритетності пацієнтів за рахунок застосування моделі машинного навчання.

Основним завданням роботи є:

- проаналізувати сучасні методи оцінки пріоритетності пацієнтів, їх переваги та недоліки;
- проаналізувати існуючі методи машинного навчання для оцінки пріоритетності пацієнтів;
- підібрати статистичні дані для навчання та тестування моделі машинного навчання;
- побудувати модель оцінювання пріоритетності пацієнтів;
- натренувати побудовану модель машинного навчання;
- провести тестування розробленої моделі.

Методами дослідження є методи машинного навчання.

1 АНАЛІЗ ОЦІНКИ ПРІОРИТЕТНОСТІ ПАЦІЄНТІВ У ВІДДІЛІ НЕВІДКЛАДНОЇ ДОПОМОГИ

1.1 Особливості визначення пріоритетності пацієнтів у відділенні невідкладної допомоги

Відділення невідкладної медичної допомоги є критично важливими пунктами надання допомоги пацієнтам з терміновими та загрозливими для життя станами. Ефективне визначення пріоритетності пацієнтів у відділенні невідкладної допомоги має вирішальне значення для того, щоб ті, хто потребує негайної медичної допомоги, отримали її якнайшвидше. Процес визначення пріоритетності пацієнтів включає кілька систематичних кроків і спирається на різні встановлені методи та критерії.

Ключовим моментом у визначенні пріоритетності пацієнтів у відділеннях невідкладної допомоги є процес медичного сортування. Медичне сортування - це система клінічної оцінки, яка використовується для визначення пріоритетності лікування пацієнтів на основі тяжкості їхнього стану. Цей процес, як правило, проводиться підготовленою медсестрою або іншими медичними працівниками на етапі первинного контакту з пацієнтом.

Після прибуття кожен пацієнт проходить попереднє обстеження, під час якого вимірюються основні життєво важливі показники (такі як частота серцевих скорочень, артеріальний тиск, частота дихання і температура). Також записується основна скарга та історія хвороби пацієнта.

На основі первинної оцінки пацієнтів розподіляють за різними рівнями терміновості. Найпоширеніші системи сортування включають Манчестерську систему сортування (MTS), Індекс тяжкості невідкладних станів (ESI) та Просте сортування та швидке лікування (START). Кожна система має конкретні критерії

для розподілу пацієнтів на рівні від невідкладного (що потребує негайного втручання для порятунку життя) до нетермінового (що потребує лікування, але не негайного втручання).

Після розподілу за категоріями пацієнти розподіляються у відповідні лікувальні відділення в межах відділу невідкладної допомоги. Пацієнти з вищим пріоритетом направляються в зони інтенсивної терапії, в той час як пацієнти з нижчим пріоритетом можуть чекати в зонах менш інтенсивної терапії.

1.2 Роль пріоритизації у оптимізації потоку пацієнтів

Крім особливостей процесу визначення пріоритетності пацієнтів у відділі невідкладної допомоги, важливо також розуміти, який вплив цей процес має на оптимізацію потоку пацієнтів. Ефективна пріоритизація пацієнтів є не тільки критично важливим компонентом для надання своєчасної допомоги, а й життєво важливим фактором підвищення загальної ефективності та функціональності відділень невідкладної допомоги.

Процес первинного медичного сортування сприяє впорядковуванню потоку пацієнтів у відділенні невідкладної медичної допомоги. Систематично оцінюючи та розподіляючи пацієнтів за категоріями на основі невідкладності їхніх станів, можна ефективно скеровувати пацієнтів на відповідні шляхи надання медичної допомоги, підвищуючи операційну ефективність.

Такий організований підхід надає низку переваг для покращення потоку пацієнтів:

- зменшення затримок для критичних пацієнтів. Високопріоритетні пацієнти, такі як пацієнти із загрозливими для життя станами, негайно скеровуються до відділень інтенсивної терапії. Таке швидке реагування має вирішальне значення для запобігання затримкам у

наданні життєво необхідного лікування та швидкої стабілізації стану пацієнтів;

- оптимізація використаних ресурсів. Пацієнти з різним ступенем терміновості отримують відповідні ресурси. Ресурси інтенсивної терапії зарезервовані для тих, хто їх гостро потребує, тоді як менш термінові випадки лікуються у відділеннях, які не потребують інтенсивного використання ресурсів. Такий баланс запобігає надмірному використанню закладів інтенсивної терапії та сприяє ефективному використанню медичного персоналу та обладнання;
- зменшення заторів. Спрямовуючи пацієнтів до відповідних відділень відповідно до їхньої терміновості, процес сортування допомагає запобігти утворенню заторів у певних зонах відділення. Такий ефективний розподіл пацієнтів по відділенню покращує загальний потік пацієнтів, забезпечуючи безперебійну роботу відділення;
- динамічне сортування. Ефективне сортування включає регулярну переоцінку пацієнтів у зоні очікування. Якщо стан пацієнта погіршується, його пріоритетність може бути змінена і він може бути переведений на вищий рівень гостроти, що гарантує швидке задоволення його потреб, які змінюються. Таке динамічне регулювання допомагає звести до мінімуму час очікування для тих, хто потребує невідкладної допомоги;
- оптимізація робочих процесів. Розподіл пацієнтів за категоріями на основі терміновості сприяє безперебійним робочим процесам у відділенні невідкладної медичної допомоги. Медичні працівники можуть зосередитися на наданні допомоги, а не на вирішенні логістичних питань, оскільки пацієнти направляються у відповідні відділення, а ресурси використовуються ефективно. Такий оптимізований підхід зменшує затримки і покращує загальний темп роботи.

Крім того такий підхід, постійно покращується. Системи сортування генерують цінні дані про потік пацієнтів, використання ресурсів та результати

лікування. Аналіз цих даних дозволяє лікарням виявляти недоліки, відстежувати ефективність і впроваджувати науково обґрунтовані вдосконалення. Постійний моніторинг і коригування на основі аналізу даних сприяють стабільній операційній ефективності.

Загалом процес визначення пріоритетності пацієнтів у відділеннях невідкладної допомоги відіграє вирішальну роль в оптимізації потоку пацієнтів, скороченні часу очікування та підвищенні операційної ефективності. Ефективні системи сортування гарантують, що пацієнти отримують потрібну допомогу в потрібний час, що має важливе значення для покращення результатів лікування та підтримки безперебійної роботи відділень невідкладної допомоги. Оскільки системи охорони здоров'я продовжують розвиватися, використання передових технологій і підходів на основі даних у сортуванні залишатиметься життєво важливим для вирішення складних завдань невідкладної медичної допомоги та забезпечення найвищих стандартів обслуговування пацієнтів.

1.3 Аналіз сучасних методів визначення пріоритетності пацієнтів

1.3.1 The Manchester Triage System

The Manchester Triage System (MTS) або Манчестерська система сортування є однією з найпоширеніших систем сортування у відділеннях невідкладної медичної допомоги в Європі та за її межами. Створена в 1990-х роках, MTS була розроблена для стандартизації процесу сортування, гарантуючи, що пацієнти оцінюються і пріоритезуються на основі невідкладності їх клінічного стану.

MTS розподіляє пацієнтів на п'ять пріоритетних рівнів, кожен з яких позначений певним кольоровим кодом. Ці категорії ґрунтуються на представленні симптомів і клінічних даних з метою полегшення швидкої і точної оцінки стану пацієнтів:

1. Червоний (невідкладний): Пацієнти потребують негайної уваги та втручання, щоб запобігти втраті життя, кінцівки або функції основних органів.
2. Помаранчевий (дуже терміновий): Стан пацієнта є потенційно небезпечним для життя або може швидко погіршитися, що вимагає втручання протягом 10 хвилин.
3. Жовтий (терміновий): Пацієнта необхідно оглянути протягом 60 хвилин, щоб запобігти подальшому погіршенню стану або значному дискомфорту.
4. Зелений (стандартний): Пацієнти повинні бути оглянуті протягом 120 хвилин; їхні стани не загрожують життю, але потребують своєчасної медичної допомоги.
5. Синій (нетерміновий): Пацієнти можуть безпечно чекати довше, ніж 120 хвилин; їхні стани незначні і навряд чи погіршаться.

Кожна категорія в MTS визначається набором блок-схем і алгоритмів, якими керуються медичні працівники при оцінці терміновості стану пацієнта. Ці блок-схеми враховують основну скаргу пацієнта, життєво важливі показники та специфічні клінічні симптоми.

Для впровадження MTS необхідно виконати кілька важливих кроків для забезпечення її ефективності та узгодженості в різних закладах охорони здоров'я. Одним із таких кроків є навчання. Широкі навчальні програми мають важливе значення для медичних сестер та інших медичних працівників, які займаються сортуванням. Ці програми охоплюють принципи MTS, використання блок-схем та навчання на основі сценаріїв для покращення навичок прийняття рішень.

Наступним кроком є стандартизація, оскільки MTS забезпечує стандартизований підхід до сортування, зменшуючи варіативність в оцінці пацієнтів і визначенні пріоритетів. Для підтримання послідовності використовуються стандартизована документація та інструменти оцінки.

Переваги MTS:

1. Стандартизація та послідовність: Забезпечуючи структурований підхід до сортування, MTS мінімізує суб'єктивну варіативність і забезпечує послідовну оцінку стану пацієнта в різних медичних закладах та умовах;
2. Всебічне охоплення: Блок-схеми та алгоритми MTS охоплюють широкий спектр клінічних сценаріїв і скарг, що робить її придатною для різних пацієнтів;
3. Доказовість: Розробка та постійне вдосконалення MTS ґрунтується на клінічних дослідженнях та доказах, що підвищує її надійність та обґрунтованість;
4. Покращення результатів лікування: Визначаючи пріоритетність пацієнтів на основі клінічної нагальності, MTS допомагає забезпечити своєчасне втручання для тих, хто його найбільше потребує, що потенційно покращує клінічні результати та задоволеність пацієнтів.

Недоліки MTS:

1. Складність: Детальні алгоритми та блок-схеми MTS можуть бути складними і важкими для навігації, особливо для менш досвідчених медичних працівників;
2. Вимоги до навчання: Впровадження MTS вимагає значної підготовки та постійного навчання персоналу, який займається сортуванням, що може бути ресурсномістким;
3. Суб'єктивність: Хоча MTS спрямована на стандартизацію сортування, деякі елементи все ще покладаються на клінічне судження, яке може відрізнятися у різних лікарів і призводити до непослідовних рішень щодо сортування.

Загалом Манчестерська система сортування – це надійна і широко прийнята система сортування, яка пропонує численні переваги для визначення пріоритетності пацієнтів у відділеннях невідкладної медичної допомоги. Її

структурований підхід, заснований на клінічних доказах, підвищує послідовність і надійність оцінки стану пацієнта, сприяючи поліпшенню клінічних результатів. Однак успішне впровадження MTS вимагає ретельного врахування її складності та як наслідок потреб у навчанні.

1.3.2 The Emergency Severity Index

The Emergency Severity Index (ESI) або Індекс тяжкості невідкладних станів – це п'ятирівнева система сортування, яка широко використовується у відділеннях невідкладної медичної допомоги у США та багатьох інших країнах. Розроблена наприкінці 1990-х років, ESI має на меті визначити пріоритетність пацієнтів на основі тяжкості їхніх станів та очікуваних потреб у ресурсах. У цьому розділі надано всебічний огляд ESI, включаючи його структуру, впровадження, переваги, недоліки та міркування щодо використання в сучасних відділах невідкладної допомоги.

Так само як і MTS, ESI класифікує пацієнтів за п'ятьма рівнями, де рівень 1 є найбільш терміновим, а рівень 5 - найменш терміновим. Однак на відміну віл MTS, у ESI при визначенні пріоритетності пацієнтів враховується ресурсний фактор.

Система розроблена таким чином, щоб бути швидкою та ефективною, дозволяючи медсестрам сортування приймати швидкі рішення щодо пріоритетності пацієнта на основі клінічних критеріїв та потреб у ресурсах:

Рівень 1: Пацієнти потребують негайного втручання для порятунку життя.

Рівень 2: Пацієнти перебувають у групі високого ризику, розгублені, мляві, дезорієнтовані або відчувають сильний біль чи дистрес.

Рівень 3: Пацієнти потребують багатьох ресурсів, але їхньому життю та здоров'ю не загрожує безпосередня небезпека.

Рівень 4: Пацієнти потребують лише одного ресурсу.

Рівень 5: Пацієнти не потребують жодних ресурсів, окрім огляду лікаря.

Процес сортування ESI включає первинну оцінку, щоб визначити, чи потребує пацієнт негайного втручання для порятунку життя (рівень 1). Якщо ні, то медсестра оцінює стан пацієнта, щоб визначити, чи є він у групі високого ризику або у важкому дистресовому стані (рівень 2). Для рівнів 3, 4 і 5 медсестра оцінює кількість необхідних ресурсів, таких як лабораторні аналізи, візуалізація або процедури.

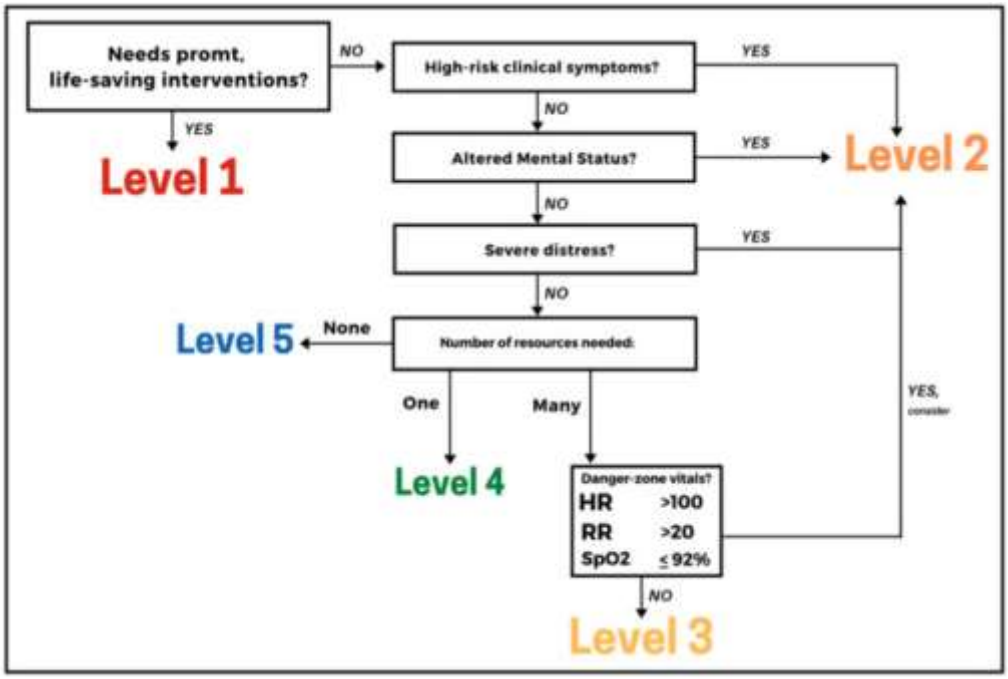


Рис 1.1 Алгоритм ESI

Впровадження ESI потребує подібних заходів до MTS, для забезпечення його ефективного використання в умовах відділення невідкладної допомоги, включаючи тренування медичного персоналу специфіці цієї сортувальної системи та стандартизації документації.

Переваги ESI:

1. Ефективність: ESI розроблений як швидкий і простий інструмент сортування, що дозволяє швидко оцінити пацієнта і визначити його пріоритетність.

2. Управління ресурсами: Прогнозуючи кількість ресурсів, необхідних для кожного пацієнта, ESI допомагає оптимізувати розподіл ресурсів і покращує потік пацієнтів у відділенні невідкладної допомоги.

3. Масштабованість: ESI може бути легко впроваджений в різних умовах відділення невідкладної допомоги, від великих міських лікарень до невеликих сільських закладів.

4. Науково обґрунтований: ESI постійно оновлюється на основі клінічних досліджень та зворотного зв'язку, що забезпечує його актуальність і точність у сучасній практиці.

Недоліки ESI:

1. Зосередженість на ресурсах: Акцент ESI на потребах у ресурсах може не враховувати тяжкість деяких станів, що може призвести до затримки в наданні допомоги певним пацієнтам.

2. Вимоги до навчання: І хоч ESI є більш простою системою сортування, вона все одно вимагає підготовки та постійного навчання персоналу сортування.

3. Варіативність: Відмінності в клінічних судженнях можуть призвести до непослідовних рішень щодо сортування, особливо для рівнів 3, 4 і 5, де необхідна оцінка ресурсів.

Підсумовуючи, Індекс тяжкості невідкладного стану (ESI) – це надійна та ефективна система сортування, яка пропонує численні переваги для визначення пріоритетності пацієнтів у відділеннях невідкладної медичної допомоги. Його структурований підхід, заснований на клінічних доказах, підвищує ефективність і точність оцінки пацієнтів, сприяючи поліпшенню клінічних результатів і оптимізації використання ресурсів.

1.3.3 Simple triage and rapid treatment

Simple triage and rapid treatment (START) або Просте сортування та швидке лікування – більш проста система медичного сортування, що широко застосовується у США. Через свою простоту частіше використовується для сортування пацієнтів під час інцидентів з масовими жертвами. Система START спрямована на швидке розподілення постраждалих за категоріями на основі тяжкості їхніх травм і терміновості надання їм медичної допомоги.

Система сортування START використовує кольорову класифікацію для визначення пріоритетності пацієнтів за чотирма категоріями:

1. Чорний (померлий/померла): Постраждалі, які померли або отримали настільки серйозні травми, що їхнє виживання малоймовірне.
2. Червоний (невідкладна допомога): Загрозливі для життя травми, що потребують негайної медичної допомоги.
3. Жовтий (відкладний): Серйозні, але не небезпечні для життя травми, які можуть зачекати на медичну допомогу.
4. Зелений (незначні): Незначні травми, які не потребують негайної медичної допомоги.

Процес сортування передбачає оцінку пацієнтів за трьома основними критеріями: дихання, кровообіг та психічний стан.

Алгоритм зображено на рисунку 1.2.

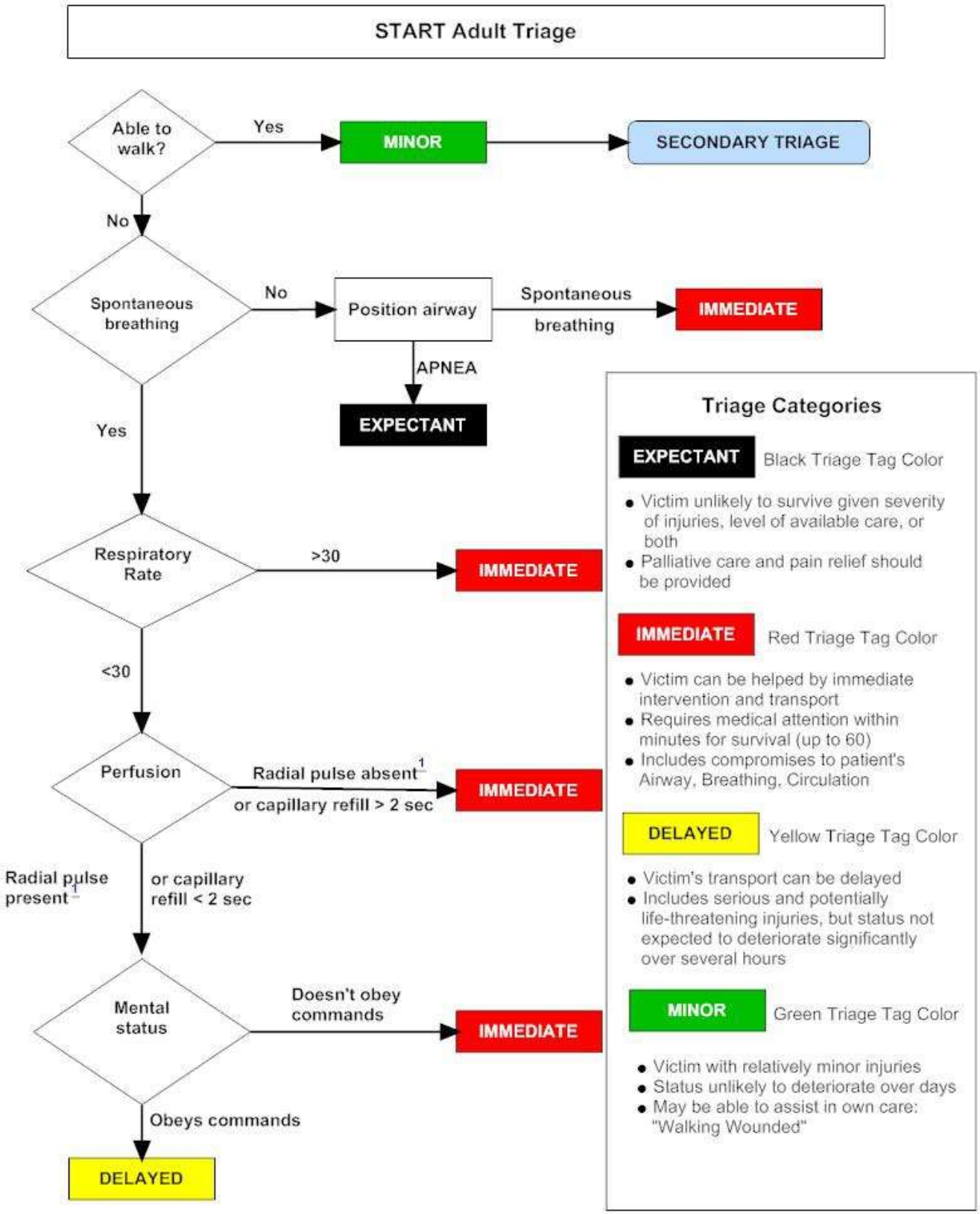


Рис 1.2 Алгоритм START

Переваги START:

1. Швидкість та ефективність: Головною перевагою системи START є її призначення для швидкого сортування великої кількості пацієнтів за короткий проміжок часу. Ця ефективність має вирішальне значення під час інцидентів з масовими жертвами, коли медичні ресурси обмежені.

2. Простота: Система використовує чіткі та зрозумілі критерії, які можуть бути легко засвоєні та застосовані особами, що надають першу допомогу, з мінімальною підготовкою.

3. Уніфікованість: Використання стандартизованої системи кольорового кодування допомагає забезпечити однакове розуміння всіма учасниками реагування тяжкості стану кожного пацієнта.

4. Швидка ідентифікація критичних пацієнтів: Зосереджуючись на критичних життєвих функціях, система START допомагає швидко ідентифікувати пацієнтів, які потребують негайної допомоги.

Недоліки START:

1. Обмежена деталізація: Система START забезпечує швидку оцінку, але ця перевага коштує системі детальної оцінки, необхідної для прийняття більш тонких медичних рішень. Іноді це може призвести до надмірного або недостатнього транспортування.

2. Не враховує всі фактори: Система фокусується насамперед на безпосередніх фізіологічних показниках і не враховує інші важливі фактори, такі як основні медичні стани або вік пацієнта.

3. Менш ефективна для нетравматичних інцидентів: Хоча система START є ефективною для травматичних сценаріїв, вона може бути менш придатною для інцидентів, пов'язаних з нетравматичними медичними невідкладними станами, такими як хімічний або біологічний вплив.

Система сортування START є цінним інструментом для швидкої оцінки та категоризації пацієнтів під час інцидентів з масовими жертвами. Її простота і

швидкість роблять її ідеальним вибором для тих, хто надає першу допомогу в хаотичному середовищі. Однак її обмеженість у деталях і потенційна неточність роблять її гіршим варіантом в рутинних ситуаціях.

1.3.4 South African Triage System

South African Triage Scale (SATS) або Південноафриканська шкала сортування – це інструмент сортування, розроблений для оптимізації процесу визначення пріоритетності пацієнтів на основі тяжкості їхнього стану, особливо в умовах обмежених ресурсів. Він був розроблений як простий, швидкий та ефективний інструмент, що дозволяє медичним працівникам швидко визначати терміновість стану пацієнта та відповідно розподіляти ресурси. SATS набула популярності не лише в Південній Африці, але й в інших країнах з низьким і середнім рівнем доходу, які стикаються з подібними проблемами.

Система SATS відносить пацієнтів до одного з чотирьох пріоритетних рівнів:

1. Червоний (невідкладний): стани, що загрожують життю і потребують негайного втручання.
2. Помаранчевий (дуже терміновий): серйозні стани, що потребують негайної медичної допомоги.
3. Жовтий (терміновий): помірні стани, які є стабільними, але потребують уваги протягом короткого періоду часу.
4. Зелений (нетерміновий): незначні стани, які можуть безпечно зачекати на лікування.

Процес SATS передбачає поєднання короткої клінічної історії, первинного обстеження для виявлення станів, що загрожують життю, та використання дискримінативної фізіологічної оцінки. Ключовим компонентом є шкала раннього попередження (Triage Early Warning Score, TEWS), яка оцінює такі життєво важливі показники, як частота дихання, частота серцевих скорочень, систолічний артеріальний тиск, температура та рівень свідомості.

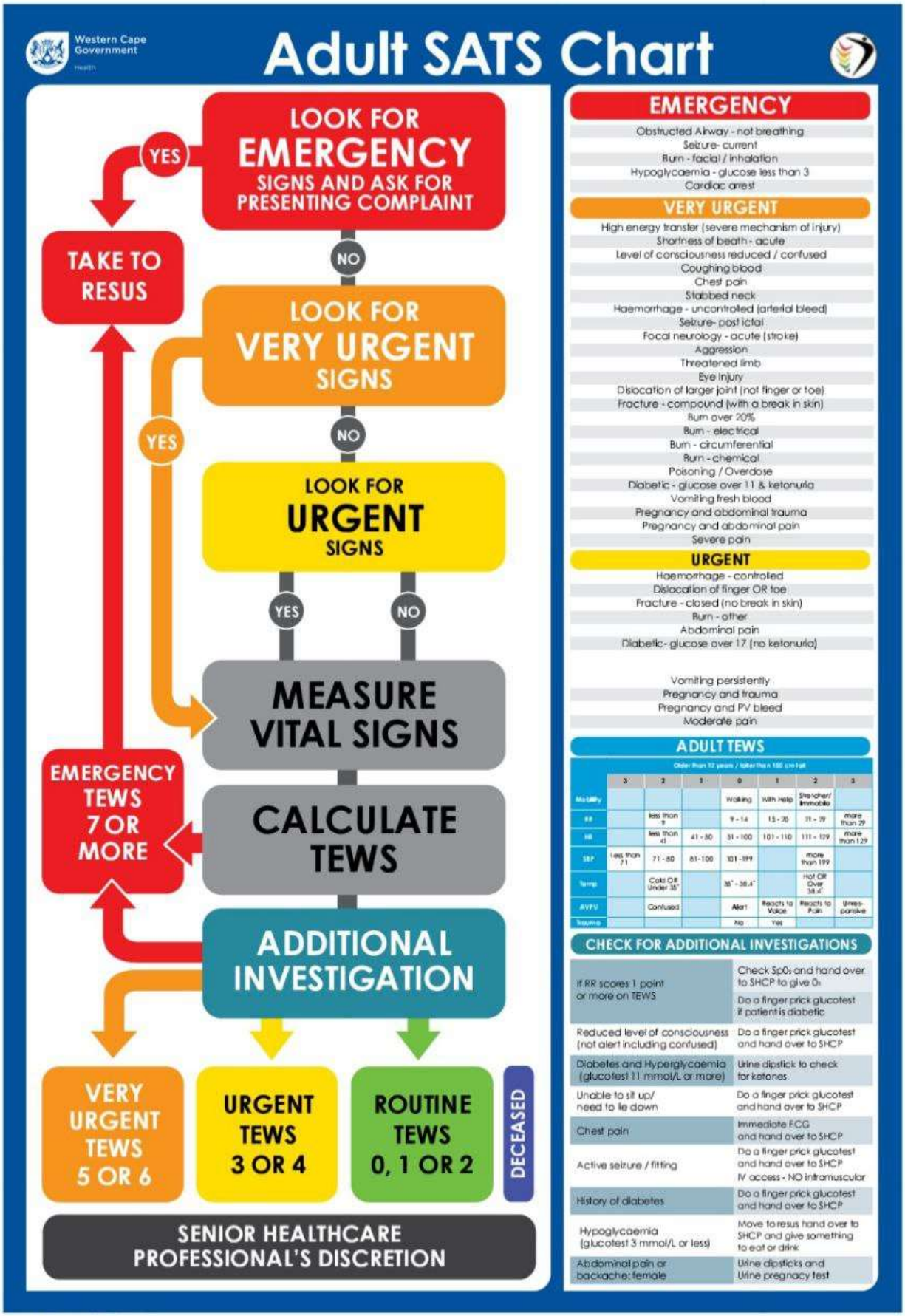


Figure 4: Adult SATS chart

Рис 1.3 Діаграма алгоритму SATS

Переваги SATS:

1. Простота і швидкість: SATS розроблена так, щоб бути простою у використанні і швидкою у застосуванні, що робить її придатною для великих обсягів роботи в умовах обмежених ресурсів. Проста категоризація допомагає медичним працівникам приймати швидкі рішення без тривалого навчання.

2. Адаптивність: Система може бути адаптована до конкретних потреб і обставин різних закладів охорони здоров'я, особливо в країнах з низьким і середнім рівнем доходу. Така гнучкість робить її універсальною для різних медичних закладів.

3. Ефективність використання ресурсів: Визначаючи пріоритетність пацієнтів за ступенем терміновості, SATS допомагає забезпечити ефективне використання обмежених ресурсів охорони здоров'я. Це особливо важливо в умовах дефіциту медичних ресурсів і персоналу.

4. Покращення результатів лікування пацієнтів: Дослідження показали, що впровадження SATS може призвести до покращення результатів лікування пацієнтів за рахунок зниження рівня смертності та покращення своєчасності надання допомоги критично хворим пацієнтам.

5. Навчання та впровадження: SATS можна навчити медичних працівників відносно швидко, а її впровадження не вимагає спеціальних медичних знань або обладнання, що робить її доступною для широкого кола медичних працівників.

Недоліки SATS:

1. Обмежена сфера застосування для складних випадків: SATS може бути не настільки ефективним у роботі з дуже складними медичними випадками, які потребують детального клінічного обстеження. Його простота, хоча і сприяє швидкості, іноді може призвести до надмірного спрощення стану пацієнта.

2. Залежність від точних життєвих показників: Компонент TEWS значною мірою залежить від точного вимірювання життєво важливих показників. Непослідовні або неправильні вимірювання можуть призвести до неправильної класифікації терміновості пацієнта, що потенційно може вплинути на результати лікування.

3. Варіативність навчання: Хоча SATS відносно легко вивчити, якість сортування може суттєво відрізнятися залежно від рівня підготовки та досвіду медичних працівників, які її використовують. Недостатня підготовка може поставити під загрозу ефективність системи.

4. Бракування комплексних оцінок: На відміну від більш досконалих систем сортування, таких як ESI або MTS, SATS може не мати можливості комплексної оцінки та детальних рекомендацій для широкого спектру клінічних сценаріїв.

Система сортування SATS пропонує прагматичний та ефективний підхід до визначення пріоритетності пацієнтів в умовах обмежених ресурсів. Її простота, адаптивність і спрямованість на швидке прийняття рішень роблять її особливо придатною для умов з великою кількістю пацієнтів і обмеженими ресурсами. Однак важливо визнати її обмеження, особливо в порівнянні з більш досконалими системами сортування, такими як ESI і MTS, які пропонують більшу деталізацію і точність, але вимагають більше ресурсів і навчання.

Таблиця 1.1

Порівняння систем сортування

Критерій	MTS (Manchester Triage System)	ESI (Emergency Severity Index)	START (Simple triage and rapid treatment)	SATS (South African Triage Scale)
Точність	Висока	Помірна	Низька	Помірна

Продовження Таблиці 1.1

Порівняння систем сортування

Критерій	MTS (Manchester Triage System)	ESI (Emergency Severity Index)	START (Simple triage and rapid treatment)	SATS (South African Triage Scale)
Швидкість	Помірна	Помірна	Дуже висока	Низька
Простота використання	Складна	Помірна	Дуже проста	Проста
Складність навчання	Висока	Помірна	Низька	Низька

Можемо бачити що різні системи медичного сортування містять різну вхідну інформацію, як наприклад враховування необхідних ресурсів ESI. Через це для більш різносторонньої оцінки пріоритетності пропонується додати оцінку комплексності стану пацієнта до моделі. Таким чином у відділі невідкладної допомоги з’явиться можливість брати це до уваги та направляти менш термінових та комплексних пацієнтів на огляд до більш досвідчених лікарів, за умови, що наявні лікарі з достатнім досвідом для лікування термінових пацієнтів.

1.4 Постановка завдання дисертаційного дослідження

Проаналізувавши процес медичного сортування та його існуючі системи, можна зробити висновок, що цей процес є одним із вирішальних факторів, що впливає на фізичний стан пацієнтів невідкладної допомоги. Через це є сенс розробити модель машинного навчання для оцінки пріоритетності пацієнтів. Подібна технологія має потенціал підвищити точність визначення пріоритетності,

що зробить оптимізацію потоку пацієнтів легше та як результат підвищить кількість успішних результатів лікування.

Тому завданням цього дослідження є розробка моделі машинного навчання для оцінки терміновості пацієнтів відділу невідкладної допомоги. Розроблена модель має показувати високу точність оцінки та бути адаптивною до нових і небачених даних.

Основне завдання:

- проаналізувати вимоги до моделі;
- проаналізувати існуючі методи машинного навчання для оцінки пріоритетності пацієнтів;
- підібрати статистичні дані для навчання та тестування моделі машинного навчання;
- побудувати модель оцінювання пріоритетності пацієнтів;
- натренувати побудовану модель машинного навчання;
- провести тестування розробленої моделі.

1.5 Висновки до розділу 1

У першому розділі було оглянуто предметну область медичного сортування у відділі невідкладної допомоги. Виявлено особливості цього процесу та його вплив на оптимізацію потоку пацієнтів. Цей аналіз дає розуміння важливості цього процесу у роботі відділень невідкладної допомоги.

Було проаналізовано сучасні методи оцінки терміновості пацієнтів. Серед них є такі системи як: MTS, ESI, START та SATS, які мають свої особливості, але усі поділяють пацієнтів на 4-5 рівнів терміновості.

MTS є більш детальною системою, яка потребує курсу навчання медичних працівників для її успішного застосування у відділі невідкладної допомоги.

ESI є схожою системою на MTS яка також потребує курсу навчання, однак головною різницею є враховування ресурсів, необхідних для лікування пацієнта, якого оцінюють.

START є простою системою, яка розрахована на інциденти із масовими жертвами, головною перевагою якої є швидкість оцінки пацієнта, однак через це бракує деталізації.

SATS є іншою простою системою оцінки пріоритетності пацієнтів, яка у своєму процесі використовує шкалу раннього попередження (Triage Early Warning Score, TEWS), однак також може бракувати деталізації.

Було поставлено завдання дисертаційного дослідження, яке полягає в розробці моделі машинного навчання для оцінки терміновості пацієнтів відділу невідкладної допомоги.

2 АНАЛІЗ ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ

2.1 Вибір машинного навчання

Для вирішення задачі потрібна система, що здатна приймати параметри стану пацієнта та базуючись на них видавати відповідну оцінку пріоритетності. Однак через те, що точні алгоритми оцінки пріоритетності зазвичай є доволі комплексними, то пропонуються технології машинного навчання. Цей вибір ґрунтується на здатності цих технологій аналізувати великі обсяги медичних даних, виділяти значущі закономірності та робити прогнози на основі численних параметрів стану пацієнта. Проаналізувати подібну кількість інформації людині займе значно більше часу, а також є ризик виникнення помилок через людський фактор.

Крім того, використання машинного навчання дозволяє створити універсальний алгоритм оцінювання, який здатний об'єднати різні системи тріажу та надати більш точні й адаптовані до реальних умов оцінки стану пацієнтів. Такий алгоритм зможе брати до уваги різноманітні фактори, такі як психологічний стан пацієнта, необхідні ресурси та інші, що дозволить поєднувати сильні сторони різних систем тріажу.

Переваги використання машинного навчання для тріажу:

1. Точність і швидкість обробки даних

Алгоритми машинного навчання дозволяють обробляти медичні дані пацієнтів у реальному часі, що особливо важливо в умовах відділення невідкладної допомоги, де своєчасність рішень є критичною. Моделі машинного навчання можуть швидко аналізувати великі набори даних, враховуючи такі показники, як частота серцевих скорочень, артеріальний тиск, насичення крові киснем, беручи їх прямо з медичної апаратури. Така

комплексна оцінка допомагає точніше визначати пріоритетність пацієнтів, забезпечуючи їм своєчасну допомогу відповідно до їх стану.

2. Адаптивність і універсальність алгоритмів

Машинне навчання має здатність вчитися на нових даних, що робить алгоритми гнучкими та здатними адаптуватися до зміни медичних практик і вимог. Залежно від типу відділення та поточних умов, модель може постійно оновлюватися, на основі нових даних уточнюючи правила для більш точного сортування. Така адаптивність дозволяє забезпечити високу надійність при використанні в різних медичних закладах, незалежно від їхнього розташування чи рівня обладнання.

3. Можливість створення універсального алгоритму для інтеграції різних систем тріажу

Як було вже зазначено раніше, однією з основних переваг машинного навчання є можливість створення універсальної моделі, що може інтегрувати елементи різних міжнародних систем тріажу, таких як наприклад Manchester Triage System, Emergency Severity Index, South African Triage Scale, що були описані в першому розділі. Алгоритм, заснований на машинному навчанні, може навчатися правилам різних систем, адаптуючи їх під єдиний стандарт. Це дозволяє медичним закладам використовувати один узагальнений підхід, що відповідає вимогам та особливостям їхнього середовища, незалежно від місцевих відмінностей у підходах до сортування пацієнтів. Універсальний підхід полегшує впровадження та стандартизує процеси тріажу у різних країнах.

4. Автоматизація та ефективність використання ресурсів

За допомогою алгоритмів машинного навчання процес сортування може бути автоматизований, що знижує навантаження на медичний персонал, особливо в умовах пікових навантажень. Автоматизація дозволяє зменшити кількість помилок, пов'язаних з людським фактором, і оптимізує

використання наявних ресурсів, спрямовуючи їх на найбільш термінові випадки. Такий підхід забезпечує ефективний розподіл ресурсів та підвищує якість обслуговування пацієнтів у відділенні невідкладної допомоги.

5. Інтеграція з електронними медичними записами

Сучасні алгоритми машинного навчання можуть бути інтегровані з електронними системами управління медичними записами, що забезпечує доступ до повної історії пацієнтів та їхніх медичних даних у реальному часі. Це дозволяє автоматично враховувати попередні захворювання, хронічні стани та інші фактори, які можуть впливати на оцінку терміновості. Крім того, така інтеграція сприяє підвищенню точності прогнозів і дозволяє уникнути потенційних ускладнень, пов’язаних із неправильною оцінкою пріорітетності.

2.2 Як працюють моделі машинного навчання

2.2.1 Лінійна регресія

Перш ніж переходити до механізмів роботи машинного навчання, варто зупинитися на понятті лінійної регресії, на принципах якої базується машинне навчання. Лінійна регресія передбачає модель, яка намагається знайти лінійний зв’язок між вхідними даними X і результатом y. Саме тому формулою обчислення лінійної регресії є звичайна лінійна функція (2.1):

$$y = w^T x + b, \tag{2.1}$$

де:

- x – вектор вхідних характеристик;
- w – вектор ваг, що визначають вплив кожної характеристики;
- b – зсув (bias), що враховує постійний зсув у даних.

Цільова функція лінійної регресії — мінімізувати функцію втрат, наприклад, середньоквадратичну помилку (2.2):

$$L = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (w^T x_i + b)\right)^2, \quad (2.2)$$

де:

- L – функція втрати;
- n – загальна кількість вхідних параметрів;
- y – значення функції;
- x – вхідні значення;
- w – вектор ваг, що визначають вплив кожної характеристики;
- b – зсув (bias), що враховує постійний зсув у даних.

Тобто для знаходження найбільш точного лінійного зв’язку, значення параметрів w та b ітеративно змінюються, допоки значення функцію втрат не буде мати найменше значення.

Лінійна регресія гарно підходить для знаходження прямих зв’язків між корелюючими даними, що є корисним. Але цей підхід не буде коректно працювати в більш складних ситуаціях, де зв’язки між параметрами не такі очевидні. Тому для вирішення таких комплексних задач були винайдені нейронні мережі, які є ключовою частиною машинного навчання.

2.2.2 Нейронні мережі

Нейронні мережі є однією з основних технологій машинного навчання, що імітують структуру та роботу біологічного мозку. Вони складаються з шарів взаємопов’язаних вузлів або нейронів, які обробляють дані через числові операції. Кожен нейрон виконує прості обчислення, але в сукупності мережа здатна моделювати складні нелінійні залежності в даних.

Зазвичай нейронні мережі мають таку структуру:

Першим шаром нейронів у мережі є її так званий вхідний шар. Цей шар приймає дані у систему у вигляді векторів характеристик. Тобто у нього входять перші параметри, за якими будуть відбуватися усі подальші обчислення. Так наприклад, для задачі медичного тріажу це можуть бути фізіологічні показники пацієнта, такі як пульс, температура тощо.

Наступним шаром у мережі є її приховані шари. Кількість таких шарів може відрізнятися у різних мереж, що визначає глибину мережі та її здатність виділяти більш складні закономірності в даних. Кожен прихований шар складається з нейронів, які виконують обчислення за допомогою ваг та зсувів (biases).

Пройшовши усі приховані шари, дані потрапляють у вихідний шар, який вже генерує результат у вигляді числових значень наприклад, прогноз категорії для задач кластеризації або оцінки пріоритетності як в поставленій задачі.

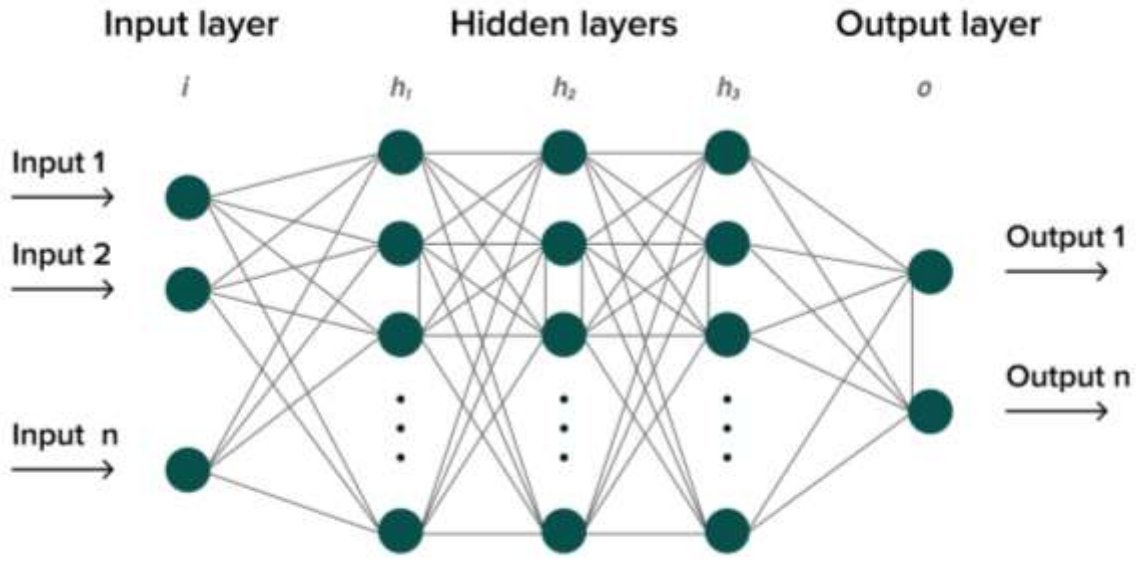


Рис 2.1 Структура нейронної мережі

Дані, проходячи через нейронну мережу від нейрону до нейрону виконують обчислення лінійної функції (2.3):

$$z = w^T x + b, \tag{2.3}$$

де:

- z – значення нейрона, що обчислюється;
- x – вектор вхідних характеристик, тобто значення попередніх нейронів;
- w – вектор ваг, значення яких прив’язані до зв’язків між нейронами;
- b – зсув (bias), значення якого прив’язано до самого нейрона.

Таким чином нейронна мережа під час своєї роботи розраховує численну кількість лінійних функцій, де вихідні результати одних стають вхідними для наступних. Однак навіть такої ускладненої системи недостатньо, щоб виявляти нелінійні патерни, адже це все ще лише серія обчислень лінійних функцій, виконаних один за одним.

Тому, на відміну від звичайної лінійної регресії, у найронних мережах застосовуються так звані активуючі функції, такі як ReLU, Sigmoid або Tanh. Вони представляють собою спеціальні нелінійні функції, які застосовуються при обчисленні кожного нейрона. Це створює нелінійність в системі, що і дозволяє виявляти більш складні патерни даних.

2.3 Методи машинного навчання

Для визначення оптимального методу машинного навчання спершу визначимо необхідні метрики оцінювання, або характеристики методів.

1. Відповідність типу – ступінь відповідності типу методу до необхідного призначення. У випадку розробки моделі оцінювання, нам необхідні

методи, що приймають числові значення характеристик пацієнта та виводять числові значення оцінок;

2. Необхідність у даних із мітками – чи необхідні для цього метода бази даних з прописаними мітками правильних відповідей. Оскільки задача розробити модель з новими метриками, такими як оцінка пріоритетності ми не будемо мати баз даних з подібними оцінками;
3. Інтерпретованість моделі – наскільки легкий для розуміння процес прийняття рішень моделі та кінцевий результат. Медична справа потребує більшої чіткості та обґрунтованості розробленої моделі, оскільки від цього залежать життя людей;
4. Потенціал продуктивності – здатність методу видавати результати з високою точністю та на небачених даних. Висока точність є базовим параметром, який є критично важливий, оскільки неточні прогнози можуть призвести до неправильного пріоритетування та вплинути на якість медичної допомоги;
5. Практичність імплементації – наскільки легко імплементувати метод та підтримувати його роботу на практиці. Легкість імплементації є важливою для швидкого розгортання і підтримки моделі в медичному середовищі, де ресурси можуть бути обмежені, а зміни повинні бути впроваджені швидко.

Для порівняння було проаналізовано 4 основних методи:

1. Навчання з вчителем (Supervised Learning, SL) – цей метод передбачає використання навчального набору даних, який містить вхідні дані та відповідні правильні відповіді, або мітки. Модель навчається на цьому наборі, щоб передбачати мітки для нових, небачених даних.
2. Навчання без вчителя (Unsupervised Learning, UL) – метод використовує навчальний набір даних, що не містить міток із правильними відповідями для виявлення структур або закономірностей у наборі даних. Модель, що використовує цей метод намагається знайти приховані шаблони або

патерни даних, що може бути корисним для виконання задач кластеризації, що є його основним застосуванням.

3. Навчання з підкріпленням (Reinforced Learning, RL) – не схожий на попередні методи, оскільки використовує агента, який взаємодіє з середовищем, отримуючи зворотний зв'язок у вигляді винагород або покарань. Таким чином через ці взаємодії він навчається приймати послідовні оптимальні рішення, намагаючись максимізувати винагороду та мінімізувати покарання.
4. Напівконтрольоване навчання (Semi-supervised Learning, SSL) – використовує комбінацію невеликого обсягу даних із мітками та великого обсягу даних без міток для навчання моделі. Через тренування моделі на невеликому обсязі даних із мітками, модель отримує початкове представлення даних та їх патернів. Далі, моделі можна надавати дані без міток для обробки. Після чого дані, у правильних відповідях яких модель найбільш впевнена додаються до основного набору даних із мітками. Таким чином модель можна тренувати наново, щоразу розширюючи тренувальну базу.

Таблиця 2.1

Порівняння методів машинного навчання

Параметр	**SL**	**UL**	**RL**	**SSL**
Відповідність типу	Відповідна	Частково відповідна	Не відповідна	Відповідна
Необхідність у даних із мітками	Висока	Відсутня	Непряма	Середня
Інтерпретованість моделі	Висока	Помірна	Складна	Висока
Потенціал продуктивності	Висока	Висока	Висока	Висока
Практичність імплементації	Висока	Помірна	Низька	Висока

На основі проведеного аналізу методів машинного навчання обрано напівконтрольоване навчання як оптимальний підхід для створення моделі оцінювання пріоритетності пацієнтів у відділеннях невідкладної допомоги. Метод напівконтрольованого навчання або Semi-supervised Learning є ефективним підходом, який поєднує невелику кількість маркованих даних (даних із відомими відповідями або мітками) із великим обсягом немаркованих даних.

Цей метод забезпечує високу точність прогнозів завдяки комбінації даних із мітками та без міток, що дозволяє моделі ефективно адаптуватися до нових даних і при цьому не залежати від повної бази маркованих даних, яка може бути обмеженою.

Чому саме напівконтрольоване навчання? Цей вибір базується на низці переваг метода, які роблять його гарним варіантом для поставленої задачі. Серед них, до головних переваг відноситься висока точність на рівні навчання з вчителем та в той же час таку модель порівняно легко розгорнути на практиці, оскільки вона вимагає меншої кількості даних із мітками, що спрощує її початкове впровадження та дозволяє адаптуватися до нових даних із часом.

Як і в методі навчання з вчителем, напівконтрольоване навчання починається з тренування моделі на наявному наборі маркованих даних. У випадку з визначенням пріоритетності пацієнтів, вхідними даними можуть бути параметри стану пацієнтів з уже встановленими рівнями терміновості. Модель навчається на цих даних розпізнавати закономірності, що асоціюються з різними рівнями пріоритетності, використовуючи певні характеристики (симптоми, життєві показники, демографічні дані тощо).

Однак через те, що цей метод використовується при невеликій кількості наявних маркованих даних, зазвичай цього буде недостатньо для коректної роботи моделі. Тому після початкового тренування модель починає працювати з великим обсягом немаркованих даних. Тут модель застосовує свій досвід з маркованих даних для передбачення пріоритетності пацієнтів, які не мають встановлених рівнів

терміновості. Цей крок дозволяє значно розширити обсяг даних, на яких модель навчається, підвищуючи її здатність робити точні прогнози на основі нової інформації.

У процесі роботи модель визначає рівень впевненості у власних прогнозах на немаркованих даних. Коли вона досягає високого рівня впевненості для конкретних випадків, ці дані додаються до бази маркованих, створюючи, таким чином, нову навчальну вибірку. Повторне тренування моделі з розширеною базою маркованих даних дозволяє їй поступово вдосконалювати свої прогнози та підвищувати точність у нових випадках.

Таким чином напівконтрольоване навчання є ітеративним процесом, під час якого модель періодично перевчається на основі нових маркованих даних, які вона сама ж і згенерувала з немаркованих. Цей цикл дозволяє моделі накопичувати знання і адаптуватися до різноманітних випадків, що зустрічаються у відділенні невідкладної допомоги. Кожен новий етап тренування робить модель точнішою та надійнішою, що сприяє її готовності до роботи з різними пацієнтами.

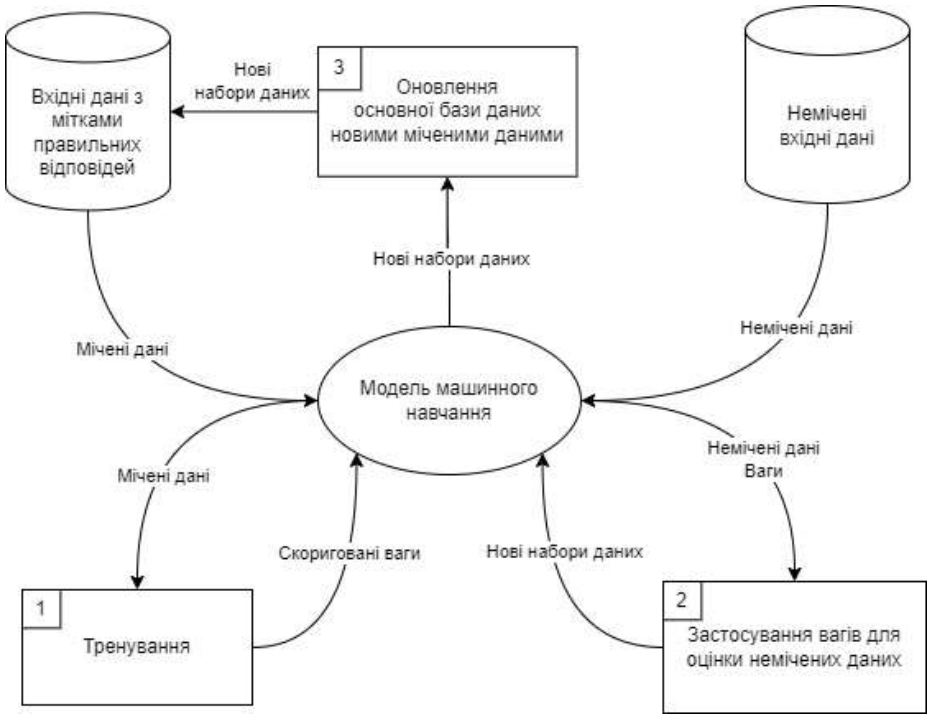


Рис 2.2 Діаграма потоку даних в методі напівконтрольованого навчання

І все ж таки, не зважаючи на свої переваги, у метода є певні обмеження. Так модель може додавати до маркованих даних деякі невірно класифіковані випадки, що може вплинути на точність подальшого навчання. Тому важливо регулярно проводити контроль якості та перевірку прогнозів моделі. Крім того, хоча напівконтрольоване і потребує значно менше маркованих даних ніж звичайне навчання з вчителем, але така база даних все ж таки залишається необхідною та чим більше в ній даних, тим точніше буде модель.

2.4 Висновки до розділу 2

Розробка моделі на основі напівконтрольованого навчання для визначення пріоритетності пацієнтів у відділеннях невідкладної допомоги включає декілька ключових етапів.

Першим етапом, як і в навчанні з вчителем є збір та підготовка початкових даних. Необхідно зібрати початкову базу даних із маркованими даними, що містять характеристики пацієнтів, такі як життєві показники, симптоми, діагнози та інші клінічні параметри і встановлені рівні терміновості.

Такі дані можуть бути зібрані з електронних медичних записів, анонімізованих історій хвороб або з наявних систем тріажу, таких як ESI, MTS або SATS, адаптованих під вимоги дослідження. Подібні бази даних можна знайти на таких сайтах як Kaggle.

Для подальшого тренування знадобиться великий обсяг немаркованих даних, зібраних із систем охорони здоров’я. Це дозволить моделі використовувати ці дані для самонавчання та покращення якості прогнозів на основі прихованих закономірностей.

Щоб ці дані можна було використовувати для тренування моделі, усі зібрані дані мають бути оброблені. Необхідно виконати очищення та стандартизацію даних: видалити дублікати, заповнити пропущені значення, стандартизувати

числові дані. Таким чином після обробки буде наявна чиста база даних одного типу, що дозволить використовувати їх у моделі.

Наступним етапом є навчання на маркованих даних. Початкове навчання моделі проводиться на маркованій вибірці для того, щоб модель змогла засвоїти основні закономірності та зв’язки між характеристиками пацієнтів і рівнями пріоритетності. Це дозволить моделі почати працювати з базовим набором правил, необхідних для подальшого самонавчання.

Далі іде ітеративне навчання з використанням немаркованих даних. Модель спершу використовує засвоєні закономірності для визначення пріоритетності немаркованих даних. На цьому етапі вона застосовує свої початкові знання, щоб прогнозувати рівні терміновості пацієнтів у немаркованій вибірці. Після чого модель автоматично оцінює рівень впевненості в своїх прогнозах. Якщо модель має високий рівень впевненості для певних випадків, ці дані додаються до маркованого набору, що дозволяє покращити якість тренувальної вибірки, яка вже використовується для повторного навчання моделі, що допомагає уточнити її прогнози та підвищити точність у наступних ітераціях.

Після тренування, для оцінки якості та валідації розробленої моделі важливо перевірити, наскільки добре модель прогнозує рівень пріоритетності пацієнтів на тестовій вибірці, що раніше не використовувалася для навчання. Це дозволяє оцінити, наскільки добре модель зможе впоратися з новими, небаченими даними у реальних умовах.

У разі виявлення значних похибок або недостатньої точності, слід скорегувати параметри моделі, або переглянути методологію навчання, зокрема, додатково попрацювати з немаркованими даними.

3 РОЗРОБКА МОДЕЛІ

3.1 Засоби розробки

3.1.1 Мова програмування

Для розробки моделі машинного навчання було обрано мову програмування Python. Ця мова програмування є популярним вибором у сфері машинного навчання та наукових досліджень, що робить його оптимальним варіантом для розробки моделі оцінки пріоритетності пацієнтів у відділеннях невідкладної допомоги. Вибір Python обґрунтовано численними перевагами, які спрощують процес створення та впровадження моделі, а також сприяють її високій продуктивності та гнучкості.

Рис 3.1 Логотип Python

Серед головних переваг можна виділити розширену екосистему бібліотек для машинного навчання та обробки даних. Адже Python має багату екосистему спеціалізованих бібліотек для машинного навчання, серед яких найпопулярніші – scikit-learn, для базових алгоритмів машинного навчання, TensorFlow та PyTorch, для нейронних мереж і глибокого навчання. Ці бібліотеки значно спрощують процес побудови, навчання та тестування моделі, пропонуючи готові інструменти

для роботи з алгоритмами машинного навчання. Крім того, бібліотеки Pandas і NumPy дозволяють легко обробляти та аналізувати дані, що є важливим для роботи з великими масивами медичної інформації.

Іншою перевагою Python є його простота синтаксису та висока читабельність коду, що дозволяє зосередитися на логіці та функціоналі моделі. Крім того, через свою популярність Python має широку підтримку та одну з найбільших спільнот розробників у світі. Це забезпечує широкий доступ до ресурсів, документації, прикладів коду та форумів. Що дозволяє швидко знаходити відповіді на технічні питання, отримувати підтримку та брати участь у розвитку технологій.

3.1.2 Середовища розробки

В якості середовища для написання коду, тестування алгоритмів та аналізу результатів були обрані PyCharm і Google Colab, оскільки кожен з цих інструментів надає специфічні переваги, що сприяють продуктивному процесу розробки.

PyCharm є повноцінним середовищем розробки на Python. Воно забезпечує інтеграцію інструментів для написання, відладки та рефакторингу коду, що особливо зручно для великих проектів. Вбудовані функції автозавершення та аналізу коду підвищують продуктивність та знижують кількість помилок. Крім того PyCharm легко інтегрується з популярними бібліотеками Python, такими як TensorFlow, PyTorch, Pandas та Scikit-learn, що спрощує створення та тестування моделі.

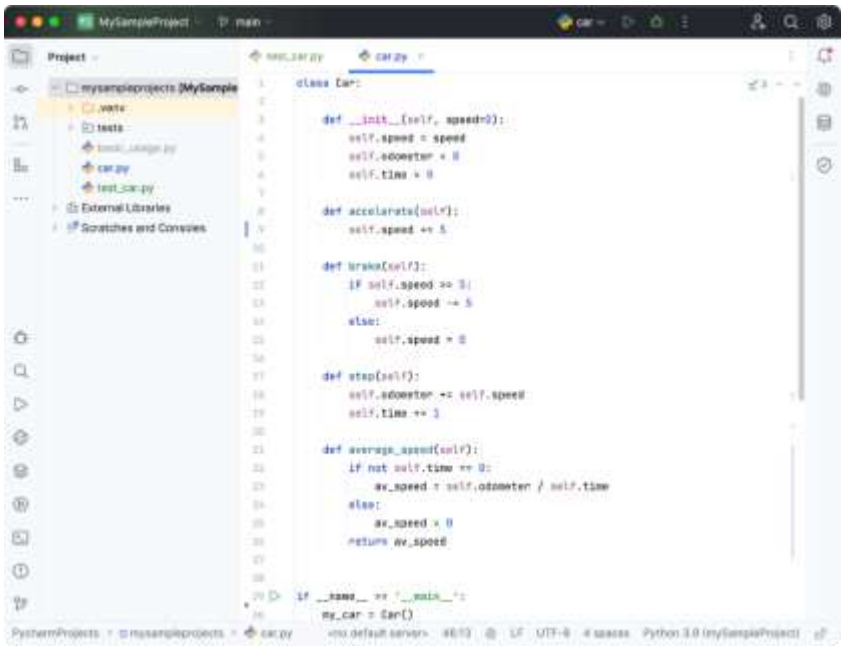


Рис 3.2 Інтерфейс PyCharm

Google Colab є альтернативним хмарним середовищем розробки на Python. Google Colab надає доступ до потужних GPU та TPU для обчислень, що є особливо важливим при тренуванні моделей із великими обсягами даних. Та його головною перевагою є інтерактивне середовище розробки (ноутбуки Jupyter), завдяки якому Colab ідеально підходить для швидкого створення прототипів, тестування гіпотез та загалом аналізу даних.

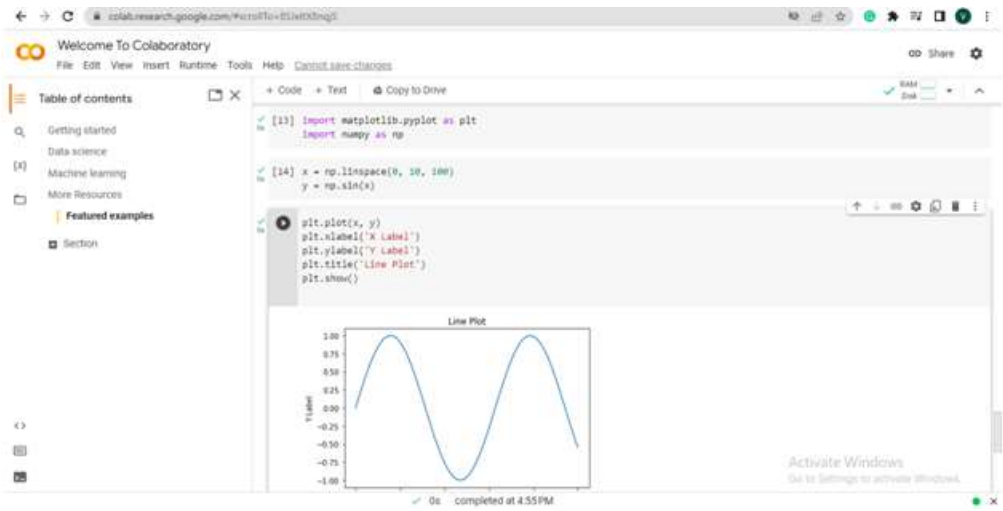


Рис 3.3 Інтерфейс Google Colab

3.1.3 PyTorch

Для розробки моделі машинного навчання було обрано PyTorch – одну з найпопулярніших бібліотек для створення, навчання та тестування нейронних мереж. Це рішення ґрунтується на її перевагах у гнучкості, продуктивності.

Завдяки динамічному обчислювальному графу PyTorch дозволяє швидко адаптувати модель до різних вимог, тестуючи різні архітектури. Для напівконтрольованого навчання це є важливим, оскільки модель потребує налаштування для роботи з маркованими й немаркованими даними. Крім того PyTorch забезпечує підтримку бібліотек, такі як PyTorch Lightning для впорядкування тренування моделей або Hugging Face Transformers для роботи з текстовими даними, що може бути корисно для майбутніх ітерацій проєкту.

3.2 Підбір тренувальних даних

Для побудови ефективної моделі визначення пріоритетності пацієнтів у відділенні невідкладної допомоги необхідно зібрати певний набір медичних даних, який дозволить моделі робити точні прогнози на основі найважливіших факторів, що впливають на стан пацієнтів. Кожен із цих типів даних несе критичну інформацію про пацієнта і забезпечує цілісну картину стану його здоров'я на момент надходження до відділення.

Необхідні такі категорії медичних даних:

1. Демографічні дані та загальна інформація про пацієнта

   Дані про стать і вік є важливими чинниками, оскільки різні групи пацієнтів, зокрема, за віком та статтю, можуть мати різні ризики та реагувати на стани по-різному. Ця інформація дозволяє моделі враховувати специфічні особливості здоров'я в залежності від групи, до якої належить пацієнт, та підвищує точність прогнозу.

2. Обставини надходження до лікарні

Інформація про те, як пацієнт прибув до лікарні: самостійно, з супроводом швидкої допомоги тощо, є важливою, оскільки вона може вказувати на рівень невідкладності стану. Пацієнти, які прибули в екстреному режимі або з супроводом медичних служб, зазвичай потребують більш ретельної уваги з боку лікарів і, можливо, пріоритетної медичної допомоги.

3. Стан свідомості та рівень болю

Стан свідомості пацієнта та рівень болю є показниками тяжкості та невідкладності стану. Зниження рівня свідомості або високий рівень болю можуть свідчити про загрозливі для життя стану, і моделі важливо враховувати ці показники, щоб відповідним чином визначати пріоритетність пацієнта.

4. Фізіологічні параметри

Життєві показники пацієнта, такі як артеріальний тиск, частота серцевих скорочень, частота дихання, температура тіла, тощо, надають інформацію про стан основних життєвих функцій пацієнта. Відхилення у цих показниках є важливими маркерами ризику, і врахування цих даних дозволяє моделі визначати, наскільки невідкладним є випадок.

5. Попередній діагноз та план лікування

Дані про попередній діагноз, встановлений у відділенні невідкладної допомоги є одним із ключових показників для визначення пріоритетності пацієнта. Наприклад вказівки на наявність травм чи інших серйозних пошкоджень дають можливість моделі зрозуміти, чи потребує пацієнт негайної допомоги та чи є ризик для його життя внаслідок цих травм. Також інформація про початкові рекомендації щодо лікування та подальшого перебування у лікарні, як наприклад перебування в палаті, госпіталізація до

відділення інтенсивної терапії, хірургічне втручання чи виписка дають змогу моделі робити точніший прогноз щодо тяжкості стану пацієнта. Крім того оцінка подальшого маршруту пацієнта допомагає визначити необхідність додаткових ресурсів або негайного втручання.

6. Тривалість перебування у відділенні невідкладної допомоги

Час, який пацієнт перебуває у відділенні невідкладної допомоги, може свідчити про складність випадку або про необхідність подальшого нагляду. Довше перебування може свідчити про важчий стан або потребу в додатковій увазі з боку медичного персоналу.

Маючи на увазі необхідні категорії медичних даних, на сайті Kaggle було знайдено базу даних із такими параметрами: «Group, Sex, Age, Patients number per hour, Arrival mode, Injury, Chief_complain, Mental, Pain, NRS_pain, SBP, DBP, HR, RR, BT, Saturation, KTAS_RN, Diagnosis in ED, Disposition, KTAS_expert, Error_group, Length of stay_min, KTAS duration_min, mistriage».

Де, як вказано в описі:

- Type of ED : Group [1 = Local ED 3th Degree, 2 = Regional ED 4tg Degree]
- Sex: Sex of the patient [1 = Famale, 2 = Male]
- Age: Age of the patient
- Arrival mode: Type of transportation to the hospital [1 = Walking, 2 = Public Ambulance, 3 = Private Vehicle, 4 = Private Ambulance, 5,6,7 = Other]
- Injury: Whether the patient is injured or not [1 = No, 2= Yes]
- Chief_complain: The patient's complaint
- Mental: The mental state of the patient [1 = Alert, 2 = Verbol Response, 3 = Pain Response, 4 = Unresponse]
- Pain: Whether the patient has pain [1 = Yes, 0 = No]
- NRS_pain: Nurse's assessment of pain for the patient
- SBP: Systolic Blood Pressure.

- DBP: Diastolic Blood Pressure.
- HR: Heat Rate.
- RR: Respiratory rate
- BT: Body Temperature
- Disposition [1 = Dischange, 2 = Admission to ward, 3 = Admission to ICU, 4 = Disharge, 5 = Transfer, 6 = Death, 7 = Surgery]
- KTAS: KTAS… [1,2,3 = Emergency, 4,5 = Non-Emergency]

Серед зазначених даних варто виділити:

1. Демографічні дані пацієнта: Sex, Age;
2. Обставини надходження до лікарні: Arrival mode;
3. Стан свідомості та рівень болю: Mental, NRS_pain;
4. Фізіологічні параметри: SBP, DBP, HR, RR, BT;
5. Попередній діагноз та план лікування: Diagnosis in ED, Disposition;
6. Тривалість перебування у відділенні невідкладної допомоги: Length of stay_min.

Таким чином знайдених набір даних має усі необхідні дані для початку роботи над моделлю машинного навчання.

3.3 Попередня обробка даних

3.3.1 Очищення даних

Попередня обробка даних — це важливий етап у розробці моделей машинного навчання, зокрема нейронних мереж. Правильна підготовка даних забезпечує стабільність, точність і ефективність моделі.

Першим етапом обробки даних зазвичай є очищення даних. Для роботи з базою даних, спершу імпортуємо бібліотеку pandas та переглядаємо дані.

```
import pandas as pd
```

```
file_path = "data.csv"
df = pd.read_csv(file_path, delimiter=';', encoding='latin1')
df
```

	Group	Sex	Age	Patients number per hour	Arrival mode	Injury	Chief_complain	Mental	Pain	NRS_pain	...	BT	Saturation	KTAS_RN	Diagnosis in ED	Disposition
0	2	2	71	3	3	2	right ocular pain	1	1	2	...	36.6	100	2	Corneal abrasion	1
1	1	1	56	12	3	2	right forearm burn	1	1	2	...	36.5	NaN	4	Burn of hand, firts degree dorsum	1
2	2	1	68	8	2	2	arm pain, Lt	1	1	2	...	36.6	98	4	Fracture of surgical neck of humerus, closed	2
3	1	2	71	8	1	1	ascites tapping	1	1	3	...	36.5	NaN	4	Alcoholic liver cirrhosis with ascites	1
4	1	2	58	4	3	1	distension, abd	1	1	3	...	36.5	NaN	4	Ascites	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1262	2	2	79	5	2	1	mental change	1	0	#BOÞ!	...	36.4	95	2	Cerebral infarction due to unspecified occlusi...	2
1263	2	2	81	2	3	1	dysuria	1	0	#BOÞ!	...	36.4	97	4	Dysuria	1
1264	2	2	81	17	2	1	dizziness	1	0	#BOÞ!	...	36.2	99	3	Dizziness and giddiness	1
1265	2	1	81	2	2	2	Sensory, Decreased	1	0	#BOÞ!	...	36.6	98	3	Cord compression, unspecified	7
1266	2	2	87	8	4	1	orthopnea	1	0	#BOÞ!	...	36.2	98	3	Dyspnoea	2

1267 rows × 24 columns

Рис 3.4 Неочищена база даних

Як ми можемо бачити, у базі даних наявні колонки даних, що не будуть використовуватись, тому їх можна прибрати.

```
df = df.drop(columns=['Group', 'Patients number per hour', 'Injury', 'Chief_complain', 'Pain', 'Saturation', 'KTAS_RN', 'KTAS_expert', 'Error_group', 'KTAS duration_min', 'mistriage'], axis=1)
```

Крім того, присутні некоректні значення, такі як NaN, або "#BOÞ!". Переглянувши базу даних, можна побачити, що значення "#BOÞ!" наявне лише у колонці оцінки болю та лише у рядках, де значення наявності самого болю дорівнює нулю. Через це, усі ці значення можна замінити на 0.

```
df['NRS_pain'] = pd.to_numeric(df['NRS_pain'], errors='coerce').fillna(0)
```

	Sex	Age	Arrival mode	Mental	NRS_pain	SBP	DBP	HR	RR	BT	Diagnosis in ED	Disposition	Length of stay_min
0	2	71	3	1	2.0	160	100	84	18	36.6	Corneal abrasion	1	86
1	1	56	3	1	2.0	137	75	60	20	36.5	Burn of hand, firts degree dorsum	1	64
2	1	68	2	1	2.0	130	80	102	20	36.6	Fracture of surgical neck of humerus, closed	2	862
3	2	71	1	1	3.0	139	94	88	20	36.5	Alcoholic liver cirrhosis with ascites	1	108
4	2	58	3	1	3.0	91	67	93	18	36.5	Ascites	1	109
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1262	2	79	2	1	0.0	120	80	86	22	36.4	Cerebral infarction due to unspecified occlusi...	2	1995
1263	2	81	3	1	0.0	120	80	94	20	36.4	Dysuria	1	1000
1264	2	81	2	1	0.0	130	90	80	20	36.2	Dizziness and giddiness	1	310
1265	1	81	2	1	0.0	170	100	78	20	36.6	Cord compression, unspecified	7	475
1266	2	87	4	1	0.0	150	80	62	20	36.2	Dyspnoea	2	887

1267 rows × 13 columns

Рис 3.5 Очищена база даних

3.3.2 Додавання колонки з мітками

Для коректної роботи моделі за методом напівконтрольованого навчання, все ще необхідні дані оцінок пріоритетності для частини наявних даних. Тому варто визначитися за якими принципами проставити ці мітки.

Різні тріажні системи, такі як MTS, ESI, START або SATS використовують логічні правила для визначення рівня пріоритетності пацієнтів. Їхня спільна мета — швидко оцінити стан пацієнтів, спираючись на життєві показники, симптоми та інші критичні фактори, що дозволяє оптимально розподілити медичні ресурси.

Одним варіантом для створення універсальної системи буде взяти готові оцінки інших систем, наприклад MTS, ESI та SATS та скомбінувати їх у єдину оцінку, помноживши їх на відповідні коефіцієнти. Наприклад (3.1):

$$Universal\ rating = MTS * 0.6 + ESI * 0.3 + SATS * 0.1. \quad (3.1)$$

Таким чином можна налаштовувати універсальну оцінку, надаючи переваги тим чи іншим системам оцінювання. Однак цей підхід має свої недоліки. Так як він передбачає наявність декількох оцінок за іншими системами, то він не дуже підходить через відсутність цих даних, а спроба спочатку надавати оцінки за іншими системами тільки підвищить ймовірність помилок.

Тому альтернативним варіантом є виставлення оцінок пріоритетності за певним набором правил. Бо хоча підходи до встановлення пріоритетів можуть варіюватися, існує кілька загальних принципів, на яких базуються всі системи.

Серед яких:

- Відхилення життєвих показників, таких як артеріальний тиск, пульс, частота дихання або температура, автоматично підвищують рівень пріоритетності. Вимірювання життєвих показників є центральним елементом у системах, таких як MTS та SATS, де ці параметри використовуються для встановлення негайності стану пацієнта.
- Системи, такі як ESI, враховують рівень болю, використовуючи шкалу Numeric Rating Scale. Значення ≥ 7 вказує на високий рівень пріоритетності, незалежно від інших показників.
- Системи можуть підвищувати пріоритет пацієнтам із серйозними діагнозами, такими як інфаркт міокарда, інсульт або сепсис. MTS забезпечує такий підхід через свої деталізовані діагностичні графіки.

Тому пропонується такий алгоритм оцінки пріоритетності:

1. Знаходимо первинні бали, оцінюючи вхідні параметри за такими критеріями:

Життєві показники:

Систолічний артеріальний тиск (САТ): ≥

$90 \leq CAT \leq 140$: 0 балів

$141 \leq CAT \leq 160$ або $80 \leq CAT < 90$: 1 бал

САТ > 160 або САТ < 80: 2 бали

Діастолічний артеріальний тиск (ДАТ):

60 ≤ ДАТ ≤ 90: 0 балів

91 ≤ ДАТ ≤ 110 або 50 ≤ ДАТ < 60: 1 бал

ДАТ > 110 або ДАТ < 50: 2 бали

Частота серцевих скорочень (ЧСС):

60 ≤ ЧСС ≤ 100: 0 балів

101 ≤ ЧСС ≤ 120: 1 бал

ЧСС > 120 або ЧСС < 60: 2 бали

Частота дихання (ЧД):

12 ≤ ЧД ≤ 20: 0 балів

21 ≤ ЧД ≤ 30: 1 бал

ЧД > 30 або ЧД < 12: 2 бали

Температура тіла (Т):

36.1 ≤ Т ≤ 37.5: 0 балів

37.6 ≤ Т ≤ 38.5 або 35.5 ≤ Т ≤ 36.1: 1 бал

Т > 38.5 або Т < 35.5: 2 бали

Рівень болю:

0-10 балів за шкалою Numeric Rating Scale

Критичність діагнозу:

0-10 балів за All Patient Refined Diagnosis Related Groups

2. Знаходимо зважені бали, перемножуючи первинні на ваги відповідних груп (3.2):

$$\text{Зважені бали} = \text{Первинні бали} * \frac{\text{Вага категорії}}{100}. \quad (3.2)$$

Критичність діагнозу – 55%

Життєві показники – 30%

Рівень болю – 15%

3. Додаємо усі зважені бали та знаходимо рівень пріоритетності за діапазонами:

0–2 балів – Пріоритет 5 (стабільний).

2–4 балів – Пріоритет 4 (помірний).

4–6 балів – Пріоритет 3 (терміновий).

6–8 балів – Пріоритет 2 (високий ризик).

8–10 балів – Пріоритет 1 (критичний).

Саме за такими правилами було проставлено рівні пріоритетності пацієнтів у перших 170 рядках бази данних. Таким чином утворюючи 2 необхідні набори даних для напівконтрольованого навчання: з мітками та без.

3.3.3 Стандартизація

Маючи усі необхідні числові значення для моделі, необхідно пройти їх стандартизацію. Стандартизація є ключовим кроком у процесі підготовки даних для моделей машинного навчання. Цей процес передбачає перетворення даних до уніфікованого масштабу, що дозволяє алгоритмам краще працювати з числовими значеннями, зменшує вплив масштабних відмінностей між характеристиками та покращує стабільність і точність моделей. Так як в наших даних є параметри, як наприклад пульс, артеріальний тиск, або час проведений у лікарні у хвилинах, то такі значення, через свій розмір будуть мати завеликий вплив, порівняно до характеристик, які лежать в діапазоні 1 – 5.

Тому стандартизація перетворює усі характеристики так, щоб їх розподіл мав середнє значення μ=0 і стандартне відхилення σ=1. Основна формула стандартизації виглядає так (3.3):

$$z = \frac{x - \mu}{\sigma}, \tag{3.3}$$

де:

- z – стандартизоване значення;
- x – початкове значення;
- μ – середнє значення для характеристики;
- σ – стандартне відхилення для характеристики.

Однак пропонується інший популярний підхід – перетворення значень так, щоб вони потрапили у визначений діапазон, наприклад, від -1 до 1 (3.4):

$$z = 2 * \frac{x - x_{min}}{x_{max} - x_{min}} - 1, \tag{3.4}$$

де:

- x_{min} – мінімальне значення у характеристиці;
- x_{max} – максимальне значення у характеристиці.

Вибір такого підходу буде пояснено у наступному підрозділі.

Для його реалізації можемо використати MinMaxScaler від sklearn:

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

df = pd.read_csv('formatted_data.csv')

columns_to_scale = df.columns.difference(['Diagnosis in ED'])

scaler = MinMaxScaler(feature_range=(-1, 1))
df[columns_to_scale] = scaler.fit_transform(df[columns_to_scale])

print(df)
```

```
df.to_csv('scaled_data.csv', index=False)
```

3.3.4 Word embedding

Таким чином дані вже майже готові до використання в моделі машинного навчання. Однак залишається одна проблема – один із головних параметрів має значення, записані у словах. І оскільки для роботи моделі, кожен параметр необхідно множити на відповідні ваги, це не буде працювати. Для такої ситуації існує word embedding, або вкладання слів – техніка перетворення текстових даних у числові, багатовимірні векторні представлення, які можна використовувати в моделях машинного навчання. Вона дозволяє алгоритмам розуміти зв'язки між словами та контекст у тексті.

Поширеними методами цієї техніки є:

- Word2Vec: Алгоритм, який навчається знаходити зв’язки між словами на основі їхнього контексту.
- GloVe: Побудова векторів на основі глобальної статистики тексту.
- BERT (Bidirectional Encoder Representations from Transformers): Інноваційний підхід, що враховує контекст слова в обох напрямках (до та після нього).

Для поставленої задачі варто звернути увагу на спеціальну адаптацію методу BERT для медицини:

ClinicalBERT – це спеціальна версія BERT, яка була навчена на текстах із медичної сфери, таких як електронні медичні записи та клінічні звіти. Цей підхід дозволяє забезпечити точніше моделювання медичних термінів, скорочень та контексту, який не завжди розуміють загальні моделі обробки тексту.

Особливості ClinicalBERT:

- ClinicalBERT був навчений на текстах, специфічних для медицини, наприклад, з MIMIC-III, великої бази клінічних даних. Це дозволяє йому розуміти складні медичні терміни та взаємозв’язки між ними.
- Як і стандартний BERT, ClinicalBERT враховує контекст слів у реченні. Наприклад, слово "суб’єкт" може мати різне значення залежно від медичного контексту.

Застосування ClinicalBERT для кодування колонки 'Diagnosis in ED' в векторні значення:

```
import pandas as pd
import torch
from transformers import AutoTokenizer, AutoModel

df = pd.read_csv('scaled_data.csv')

model_name = "emilyalsentzer/Bio_ClinicalBERT"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)

def get_batch_embeddings(diagnoses_texts):
    inputs = tokenizer(diagnoses_texts, return_tensors="pt", truncation=True,
padding=True)
    with torch.no_grad():
        outputs = model(**inputs)
        embeddings = outputs.last_hidden_state[:, 0, :]
    return embeddings.numpy()

diagnoses = df['Diagnosis in ED'].tolist()
embeddings_batch = get_batch_embeddings(diagnoses)

df['Diagnosis in ED'] = list(embeddings_batch)

print(df)

df.to_csv('embedded_data.csv', index=False)
```

Таким чином ми отримаємо базу даних, де в колонці із попередніми діагнозами замість слів записані вектори. Кожен з яких складається із 768 значень від -1 до 1. Що і є причиною обрання такого діапазону стандартизації в попередньому підрозділі.

3.3.5 Вирівнювання даних

Однак ми ще не можемо використовувати цю базу, оскільки ми маємо 12 звичайних числових значень та 1 векторне значення, яке не може бути переданим як одне в нейронній мережі. Для вирішення цієї проблеми є варіант розділити вектор на усі його 768 окремих значень. Таким чином ми отримаємо 768 + 12 = 780 вхідних параметрів, кожен із яких є одним числовим значенням від -1 до 1.

```
import pandas as pd
import numpy as np
import re
import ast

df = pd.read_csv('embedded_data.csv')

def preprocess_embedding_string(embedding_str):
    embedding_str = re.sub(r"\s+", ", ", embedding_str.strip())
    embedding_str = embedding_str.replace("[,", "[").replace(",]", "]")
    return embedding_str

df["Diagnosis in ED"] = df["Diagnosis in ED"].apply(
    lambda x: np.array(ast.literal_eval(preprocess_embedding_string(x)))
)

embedded_df = pd.DataFrame(
    df["Diagnosis in ED"].tolist(),
    columns=[f"embedding_{i}" for i in range(len(df["Diagnosis in ED"].iloc[0]))]
)

df_flattened = pd.concat([df.drop(columns=["Diagnosis in ED"]), embedded_df],
axis=1)

df_flattened = df_flattened[[col for col in df_flattened.columns if col != 'Priority
rating'] + ['Priority rating']]
```

```
df_flattened.to_csv('flattened_data.csv', index=False)
```

Після виконання написаного коду ми маємо повноцінно готову, оброблену базу даних 'flattened_data.csv', що означає, що можна вже переходити до написання самої моделі машинного навчання.

3.4 Написання моделі

Перш за все, імпортуємо усі необхідні бібліотеки, що будуть використовуватись в роботі програми:

```
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, Dataset
from sklearn.model_selection import train_test_split
import pandas as pd
import os
```

Створюємо клас CustomDataset призначений для роботи з даними у форматі, сумісному з PyTorch. Він дозволяє завантажувати дані, обробляти їх і передавати у вигляді тензорів до моделі. Цей клас спрощує роботу з тренувальними та тестовими наборами даних, зокрема для маркованих і немаркованих прикладів.

```
class CustomDataset(Dataset):
    def __init__(self, features, labels=None):
        self.features = torch.tensor(features, dtype=torch.float32)
        self.labels = torch.tensor(labels, dtype=torch.float32).view(-1, 1) if labels is
not None else None

    def __len__(self):
        return len(self.features)

    def __getitem__(self, idx):
        if self.labels is not None:
            return self.features[idx], self.labels[idx]
        return self.features[idx]
```

Метод __init__ ініціалізує набір даних, перетворюючи ознаки (features) та мітки (labels) на тензори з типом float32. Мітки додаються тільки за наявності.

Метод __len__ повертає кількість зразків у наборі даних.

Метод __getitem__ дозволяє отримати зразки за індексом – або пару "ознаки-мітка", або лише ознаки, якщо міток немає.

А також створюємо клас NeuralNetwork, що визначає архітектуру нейронної мережі, яка виконує основну задачу прогнозування на основі вхідних даних. Мережа будується з використанням повнозв'язних шарів (nn.Linear) і функцій активації LeakyReLU. Кількість шарів і нейронів у них визначається списком hidden_dims. Останній шар виконує регресійне передбачення одного числового значення оцінки пріоритетності.

Метод forward реалізує переднє проходження через шари моделі, де на кожному шарі до вхідного тензора застосовуються лінійне перетворення і функція активації.

```
class NeuralNetwork(nn.Module):
    def __init__(self, input_dim, hidden_dims, output_dim):
        super(NeuralNetwork, self).__init__()
        self.layers = nn.ModuleList()
        dims = [input_dim] + hidden_dims
        for i in range(len(dims) - 1):
            self.layers.append(nn.Linear(dims[i], dims[i + 1]))
            self.layers.append(nn.LeakyReLU(negative_slope=0.01))
        self.layers.append(nn.Linear(dims[-1], output_dim))

    def forward(self, x):
        for layer in self.layers:
            x = layer(x)
        return x
```

Метод train_semi_supervised відповідає за процес напівконтрольованого навчання. Він використовує марковані та немарковані дані для навчання моделі.

У кожній епосі модель проходить марковані дані, обчислює функцію втрат (criterion) і оновлює ваги за допомогою градієнтного спуску.

Модель прогнозує результати для немаркованих даних. Зразки з високою впевненістю, тобто вище порогу confidence_threshold, отримують псевдомітки, які додаються до маркованого набору. Новостворені пари "ознаки-псевдомітки" додаються до маркованих даних, і модель продовжує навчання на оновленому наборі.

```
def train_semi_supervised(model, labeled_loader, unlabeled_loader, criterion,
optimizer, device, epochs=50,
                          confidence_threshold=0.9):
    model.train()

    for epoch in range(epochs):
        total_loss = 0.0

        for features, labels in labeled_loader:
            features, labels = features.to(device), labels.to(device)
            optimizer.zero_grad()
            outputs = model(features)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()
            total_loss += loss.item()

        pseudo_features = []
        pseudo_labels = []

        for features in unlabeled_loader:
            features = features.to(device)
            with torch.no_grad():
                outputs = model(features)
                probabilities = torch.sigmoid(outputs).squeeze()
                high_confidence_mask = probabilities >= confidence_threshold
                if high_confidence_mask.any():
                    pseudo_features.append(features[high_confidence_mask].cpu())
                    pseudo_labels.append(probabilities[high_confidence_mask].cpu())

        if pseudo_features:
            pseudo_features = torch.cat(pseudo_features, dim=0)
            pseudo_labels = torch.cat(pseudo_labels, dim=0).view(-1, 1)
```

```
            labeled_features = torch.cat([labeled_loader.dataset.features,
pseudo_features], dim=0)
            labeled_labels = torch.cat([labeled_loader.dataset.labels, pseudo_labels],
dim=0)

            augmented_dataset = CustomDataset(labeled_features.numpy(),
labeled_labels.numpy())
            labeled_loader = DataLoader(augmented_dataset, batch_size=32,
shuffle=True)

        print(f"Epoch {epoch + 1}/{epochs}, Loss: {total_loss:.4f}")

    torch.save(model.state_dict(), "model.pth",
_use_new_zipfile_serialization=False)
    model_dir = "C:/Users/Andva/OneDrive/Рабочий стол/saved_models"
    os.makedirs(model_dir, exist_ok=True)

    torch.save(model.state_dict(), os.path.join(model_dir, "model.pth"))

    print(f"Model saved as model.pth")
```

Основна програма, або __main__ відповідає за підготовку даних, ініціалізацію моделі та запуск процесу навчання.

Дані зчитуються з CSV-файлу з обробленими даними flattened_data.csv. Рядки з маркованими та немаркованими мітками розділяються на основі наявності значень у стовпці міток. Крім того, марковані дані діляться на тренувальну та валідаційну вибірки з використанням train_test_split.

Створюються об’єкти DataLoader для передачі даних до моделі пакетами (batch_size=32).

Ініціалізується модель NeuralNetwork із параметрами: 780 вхідних ознак, приховані шари з 512 і 256 нейронами та одним виходом. Вибирається функція втрат MSELoss і оптимізатор Adam.

Та викликається функція train_semi_supervised, яка виконує навчання моделі.

```
if __name__ == "__main__":
```

```
    df = pd.read_csv("flattened_data.csv")

    features = df.iloc[:, :780].values
    labels = df.iloc[:, 780].values

    labeled_mask = ~pd.isnull(labels) & (labels != '??')
    unlabeled_mask = ~labeled_mask

    labeled_features = features[labeled_mask]
    labeled_labels = labels[labeled_mask].astype(float)
    unlabeled_features = features[unlabeled_mask]

    train_features, val_features, train_labels, val_labels = train_test_split(
        labeled_features, labeled_labels, test_size=0.2, random_state=42
    )

    labeled_train_dataset = CustomDataset(train_features, train_labels)
    labeled_val_dataset = CustomDataset(val_features, val_labels)
    unlabeled_dataset = CustomDataset(unlabeled_features)
    labeled_train_loader = DataLoader(labeled_train_dataset, batch_size=32,
shuffle=True)
    labeled_val_loader = DataLoader(labeled_val_dataset, batch_size=32,
shuffle=False)
    unlabeled_loader = DataLoader(unlabeled_dataset, batch_size=32,
shuffle=True)

    input_dim = 780
    hidden_dims = [512, 256]
    output_dim = 1
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    model = NeuralNetwork(input_dim, hidden_dims, output_dim).to(device)
    criterion = nn.MSELoss()
    optimizer = optim.Adam(model.parameters(), lr=0.001)

    train_semi_supervised(model, labeled_train_loader, unlabeled_loader,
criterion, optimizer, device)
```

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Результати роботи моделі

Після виконання програми моделі, вона почне тренування, по завершенні якого збереже модель для подальшого тестування. Для перевірки натренованої моделі, надаємо їй 3 набори вхідних даних пацієнтів:

```
Input data:
  Sex  Age  Arrival mode  Mental  NRS_pain  SBP  DBP  HR  RR   BT                Diagnosis in ED
   1   54             4       1         3  140   90  94  20 38.1              Fever, unspecified
   1   56             3       1         2  137   75  60  20 36.5 Burn of hand, firts degree dorsum
   2   78             3       1         3  169   86  80  20 36.0          Acute coronary syndrome
```

Рис 4.1 Вхідні дані пацієнтів

Після чого система надає наступний вивід:

```
Priority prediction #1:  [3.6718473]
Priority prediction #2:  [5.0061874]
Priority prediction #3:  [2.1630125]
```

Рис 4.2 Надані оцінки пріоритетності

Як можна побачити, система оцінила перший випадок у 3.67, другий – у 5 та третій – у 2.16. Такі оцінки є коректними, оскільки:

Перший пацієнт має температуру, що дає 2 бали, рівень болю дорівнює 3 балам та діагнозом є лихоманка, яка оцінюється в 6 балів. Сума зважених балів дорівнює 4.35, що підпадає у діапазон пріоритетності 3.

Другий пацієнт має стабільні життєві показники, рівень болю дорівнює 2 балам та діагнозом є опік руки першого ступеня, що оцінюється в 2 бали. Сума зважених балів дорівнює 1.4, що підпадає у діапазон пріоритетності 5.

Третій пацієнт має високий систолічний артеріальний тиск та трохи занизьку температуру, що дає 3 бали, рівень болю дорівнює 3 балам та діагнозом гострий коронарний синдром, що є критичним діагнозом та оцінюється у 10 балів. Сума зважених балів дорівнює 6.85, що підпадає у діапазон пріоритетності 2.

4.2 Аналіз результатів

Оцінка ефективності розробленої моделі машинного навчання для тріажу пацієнтів у відділеннях невідкладної допомоги є важливим етапом її впровадження. Щоб зрозуміти, наскільки нова система перевершує або відповідає існуючим методам, необхідно порівняти їх за чіткими і вимірюваними параметрами. Нижче наведені два основних параметри, за якими було здійснено порівняння результатів систем:

1. Час оцінки пацієнта

- Суть параметра: Час, необхідний для визначення пріоритетності одного пацієнта у хвилинах.
- Чому це важливо: У відділенні невідкладної допомоги кожна секунда може мати критичне значення, особливо для пацієнтів у важкому стані.

2. Точність

- Суть параметра: Відсоток оцінок, що були коректно визначені згідно з правилами досліджуваної системи. Тобто у традиційних систем оцінки пріоритетності цей параметр фактично є відсотком людської помилки. В той час як у розробленій системі це є невідповідність отриманої оцінки

оригінальним правилам, за якими модель машинного навчання була натренована.

– Чому це важливо: Точність є критичною для забезпечення відповідної медичної допомоги у встановлені часові рамки.

Оцінивши правильність 200 оцінок пріоритетності, наданих розробленою системою, маємо наступні результати:

Таблиця 4.1

Порівняльна таблиця систем оцінки пріоритетності пацієнтів

Система	Час оцінки пацієнта (хв)	Точність (%)
MTS	2-5	70-84
ESI	3-6	77-80
START	0.5-1	50-62
SATS	5-8	74-78
СМНОСП	1-3	80-84

Результати порівняння показують значні переваги розробленої системи у швидкості оцінки пацієнтів та точності визначення пріоритетності. Нижче наведено аналіз отриманих результатів та обґрунтування, чому розроблена система перевершує існуючі підходи.

1. Швидкість оцінки пацієнта

Розроблена система демонструє середній час оцінки 1–3 хвилини, що швидше за більшість традиційних систем, таких як SATS (5–8 хв) та ESI (3–6 хв), і порівняно з MTS (2–5 хв). Цей результат досягається завдяки автоматизації процесу. Так як її використання позбавляє необхідності проходити процес оцінювання пацієнта вручну, співставляючи його показники за відповідними алгоритмами як в звичайних системах. Єдиним сповільнюючим фактором

розробленої системи є необхідність у первинному діагнозі як в одному з вхідних параметрів моделі машинного навчання.

2. Точність визначення пріоритетності

Розроблена система досягає точності 80–84%, що перевершує традиційні системи, такі як ESI (77–80%) і SATS (74–78%). Основною причиною цього є здатність алгоритмів машинного навчання враховувати складні нелінійні залежності між характеристиками пацієнта, що важко реалізувати вручну. Крім того, у деяких традиційних систем сортування пацієнтів алгоритми прописані недостатньо точно, залишаючи місце для інтерпретації людини, що проводить оцінку. Це підвищує ймовірність помилок через людський фактор на брак досвіду.

ВИСНОВКИ

Тріаж, або сортування пацієнтів у відділеннях невідкладної допомоги є критично важливим процесом, який забезпечує своєчасну медичну допомогу тим, хто її найбільше потребує. Цей процес передбачає сортування пацієнтів за рівнем терміновості, що допомагає ефективно розподілити обмежені ресурси медичного персоналу та обладнання. У сучасних реаліях, зокрема через перевантаження лікарень, зростання кількості звернень та глобальні виклики, такі як пандемії, автоматизація цього процесу стає дедалі більш актуальною.

Автоматизація тріажу дозволяє підвищити швидкість і точність прийняття рішень, зменшуючи ризик людських помилок. Традиційні системи сортування часто залежать від людського фактору, що робить їх уразливими до суб’єктивних рішень, втоми персоналу чи високого навантаження. Використання моделей машинного навчання дає можливість стандартизувати процес, мінімізувати часові витрати та забезпечити високу точність оцінювання.

У роботі проаналізовано системи оцінки пріоритетності пацієнтів: MTS, ESI, START, SATS. Основна увага приділялась ключовим параметрам, таким як час, необхідний для оцінки пацієнта, та точність визначення пріоритетності, для порівняння із розробленою системою;

Було досліджено основні методи машинного навчання: з учителем, без учителя, з підкріпленням та напівконтрольоване навчання. Для реалізації автоматизованої системи обрано напівконтрольоване навчання, оскільки цей метод забезпечує високу точність навіть за умови обмеженої кількості маркованих даних. Такий підхід виявився оптимальним у контексті доступності медичних даних, де часто є велика кількість немаркованих записів.

Реалізовано систему для оцінювання пріоритетності пацієнтів на основі моделі машинного навчання. За результатами тестування системи було виявлено, що вона є більш ефективною ніж більшість проаналізованих систем. А саме:

- Швидкість оцінки пацієнта за розробленою системою становить 1–3 хв, що швидше за більшість проаналізованих систем. Єдиним винятком є START, яка має час оцінки 0.5–1 хв, але її простота робить її менш точною.
- Точність розробленої системи становить 80–84%, що в середньому на 7% вище за показники такої точної системи, як MTS.

Розроблена система не лише забезпечує швидке й точне сортування пацієнтів, але й може бути інтегрована в існуючі інформаційні системи медичних закладів для покращення роботи відділень невідкладної допомоги. Це робить її перспективним рішенням для оптимізації медичних процесів у сучасних лікарнях.

ПЕРЕЛІК ПОСИЛАНЬ

1. ElGammal M. E. Emergency Department triage Why and How?. Saudi Med J. 2014. № 35.
2. Performance of triage systems in emergency care: a systematic review and meta-analysis / J. M. Zachariasse та ін. BMJ Open. 2019. Т. 9, № 5. С. e026471. URL: https://doi.org/10.1136/bmjopen-2018-026471 (дата звернення: 22.10.2024).
3. Roberts D. C., McKay M. P., Shaffer A. Increasing Rates of Emergency Department Visits for Elderly Patients in the United States, 1993 to 2003. Annals of Emergency Medicine. 2008. Т. 51, № 6. С. 769–774. URL: https://doi.org/10.1016/j.annemergmed.2007.09.011 (дата звернення: 18.10.2024).
4. Suamchaiyaphum K., Jones A. R., Markaki A. Triage Accuracy of Emergency Nurses: An Evidence-Based Review. Journal of Emergency Nursing. 2023. URL: https://doi.org/10.1016/j.jen.2023.10.001 (дата звернення: 18.10.2024).
5. Tam H. L., Chung S. F., Lou C. K. A review of triage accuracy and future direction. BMC Emergency Medicine. 2018. Т. 18, № 1. URL: https://doi.org/10.1186/s12873-018-0215-0 (дата звернення: 19.10.2024).
6. Triage Performance in Emergency Medicine: A Systematic Review / J. S. Hinson та ін. Annals of Emergency Medicine. 2019. Т. 74, № 1. С. 140–152. URL: https://doi.org/10.1016/j.annemergmed.2018.09.022 (дата звернення: 24.10.2024).
7. Yancey C. C., O'Rourke M. C. Emergency Department Triage. Treasure Island (FL): StatPearls Publishing. 2023. URL: https://www.ncbi.nlm.nih.gov/books/NBK557583/ (дата звернення: 29.10.2024).
8. Validity of the Manchester Triage System in emergency care: A prospective observational study / J. M. Zachariasse та ін. PLOS ONE. 2017. Т. 12, № 2.

C. e0170811. URL: https://doi.org/10.1371/journal.pone.0170811 (дата звернення: 22.10.2024).

9. Nusi T., Lestari Y., Suryanto S. Overview of the efficiency of using the triage Emergency Severity Index (ESI) in emergency installations: A systematic review. Jurnal Aisyah : Jurnal Ilmu Kesehatan. 2023. Т. 8, № 3. URL: https://doi.org/10.30604/jika.v8i3.2070 (дата звернення: 29.10.2024).
10. Accuracy of Triage Systems in Disasters and Mass Casualty Incidents; a Systematic Review / J. Bazyar та ін. Arch Acad Emerg Med. 2022.
11. The effectiveness of the South African Triage Score (SATS) in a rural emergency department / K. Rosedale та ін. PubMed. 2011.
12. Sarker I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science. 2021. Т. 2, № 3. URL: https://doi.org/10.1007/s42979-021-00592-x (дата звернення: 18.10.2024).
13. Study On Machine Learning Algorithms / P. R та ін. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2021. С. 67–72. URL: https://doi.org/10.32628/cseit2173105 (дата звернення: 18.10.2024).
14. An Introduction to Machine Learning Methods for Survey Researchers / T. D. Buskirk та ін. Survey Practice. 2018. Т. 11, № 1. С. 1–10. URL: https://doi.org/10.29115/sp-2018-0004 (дата звернення: 25.05.2024).
15. Muhamedyev R. I. Machine learning methods: An overview. COMPUTER MODELLING & NEW TECHNOLOGIES 2015. 2015. Т. 19, № 6. С. 14–29.
16. Mitchell T. M. Machine Learning. McGraw-Hill Science/Engineering/Math, 1997. 432 с.
17. Nasteski V. An overview of the supervised machine learning methods. HORIZONS.B. 2017. Т. 4. С. 51–62. URL: https://doi.org/10.20544/horizons.b.04.1.17.p05 (дата звернення: 30.10.2024).
18. Eltouny K., Gomaa M., Liang X. Unsupervised Learning Methods for Data-Driven Vibration-Based Structural Health Monitoring: A Review. Sensors. 2023. Т. 23, № 6. С. 3290. URL: https://doi.org/10.3390/s23063290 (дата звернення: 30.10.2024).

19. Deep learning, reinforcement learning, and world models / Y. Matsuo та ін. Neural Networks. 2022. URL: https://doi.org/10.1016/j.neunet.2022.03.037 (дата звернення: 30.10.2024).
20. Bergmann D. What Is Semi-Supervised Learning? | IBM. IBM - United States. URL: https://www.ibm.com/topics/semi-supervised-learning#:~:text=Semi-supervised%20learning%20is%20a,for%20classification%20and%20regression%20tasks. (дата звернення: 18.10.2024).
21. Risk of mortality and cardiopulmonary arrest in critical patients presenting to the emergency department using machine learning and natural language processing / M. Fernandes та ін. PLOS ONE. 2020. Т. 15, № 4. С. e0230876. URL: https://doi.org/10.1371/journal.pone.0230876 (дата звернення: 25.05.2024).
22. Machine learning-based models to support decision-making in emergency department triage for patients with suspected cardiovascular disease / H. Jiang та ін. International Journal of Medical Informatics. 2021. Т. 145. С. 104326. URL: https://doi.org/10.1016/j.ijmedinf.2020.104326 (дата звернення: 18.10.2024).
23. Predictive Models for Emergency Department Triage using Machine Learning: A Systematic Review / F. Gao та ін. Obstetrics and Gynecology Research. 2022. Т. 05, № 02. URL: https://doi.org/10.26502/ogr085 (дата звернення: 25.05.2024).
24. AP - DRGs all patient diagnosis related groups: Definitions manual. Wallingford, CT : 3M Health Information Systems, 3M Health Care, 2003.
25. GitHub - kexinhuang12345/clinicalBERT: ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission (CHIL 2020 Workshop). GitHub. URL: https://github.com/kexinhuang12345/clinicalBERT (дата звернення: 21.10.2024).
26. Releases - huggingface/transformers. GitHub. URL: https://github.com/huggingface/transformers/releases (дата звернення: 21.10.2024).
27. Emergency Service - Triage Application. Kaggle. URL: https://www.kaggle.com/datasets/ilkeryildiz/emergency-service-triage-application (дата звернення: 02.10.2024).

28. A review on utilizing machine learning technology in the fields of electronic emergency triage and patient priority systems in telemedicine: Coherent taxonomy, motivations, open research challenges and recommendations for intelligent future work / O. H. Salman та ін. Computer Methods and Programs in Biomedicine. 2021. Т. 209. С. 106357. URL: https://doi.org/10.1016/j.cmpb.2021.106357 (дата звернення: 09.10.2024).
29. A machine learning and explainable artificial intelligence triage-prediction system for COVID-19 / V. V. Khanna та ін. Decision Analytics Journal. 2023. С. 100246. URL: https://doi.org/10.1016/j.dajour.2023.100246 (дата звернення: 09.10.2024).
30. Vântu A., Vasilescu A., Băicoianu A. Medical emergency department triage data processing using a machine-learning solution. Heliyon. 2023. Т. 9, № 8. С. e18402. URL: https://doi.org/10.1016/j.heliyon.2023.e18402 (дата звернення: 09.10.2024).

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

РОЗРОБКА МЕТОДИКИ ОЦІНЮВАННЯ СТАНУ ПАЦІЄНТІВ ВІДДІЛЕННЯ НЕВІДКЛАДНОЇ ДОПОМОГИ НА ОСНОВІ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Виконав: студент групи ПДМ-61 Спіцин Андрій Ярославович

Керівник: к.ф.-м.н., доц. Садовенко Володимир Сергійович

Київ - 2025

МЕТА, ОБ’ЄКТ, ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета дослідження: підвищити точність та зменшити час оцінки пріоритетності пацієнтів за рахунок застосування моделі машинного навчання.

Об'єкт дослідження: процес оцінювання пріоритетності пацієнтів.

Предмет дослідження: модель машинного навчання для визначення пріоритетності пацієнтів.

ПОРІВНЯЛЬНА ТАБЛИЦЯ СИСТЕМ СОРТУВАННЯ

Критерії	MTS (Manchester Triage System)	ESI (Emergency Severity Index)	START (Simple triage and rapid treatment)	SATS (South African Triage Scale)	СМНОСП (Система машинного навчання для оцінки стану пацієнтів)
Точність	Висока	Помірна	Низька	Помірна	Висока
Швидкість	Помірна	Помірна	Дуже висока	Низька	Висока
Простота використання	Складна	Помірна	Дуже проста	Проста	Проста
Складність навчання	Висока	Помірна	Низька	Низька	Низька

3

ЗАСОБИ РОЗРОБКИ

4

АЛГОРИТМ ОЦІНКИ ПРІОРИТЕТНОСТІ

1. Первинні бали знаходяться, оцінюючи вхідні дані пацієнта за відповідними параметрами. Наприклад:
$60 \leq$ Пульс ≤ 100: 0 балів (норма).
$101 \leq$ Пульс ≤ 120: 1 бали (помірне відхилення).
Пульс > 120 або Пульс < 60: 2 бали (критичне).

2. Далі бали з кожної категорії сумуються та перемножуються з відповідними вагами:

$$\text{Зважені бали} = \text{Первинні бали} * \frac{\text{Вага категорії}}{100}$$

Ваги категорій:
Критичність діагнозу – 55%
Життєві показники – 30%
Рівень болю – 15%

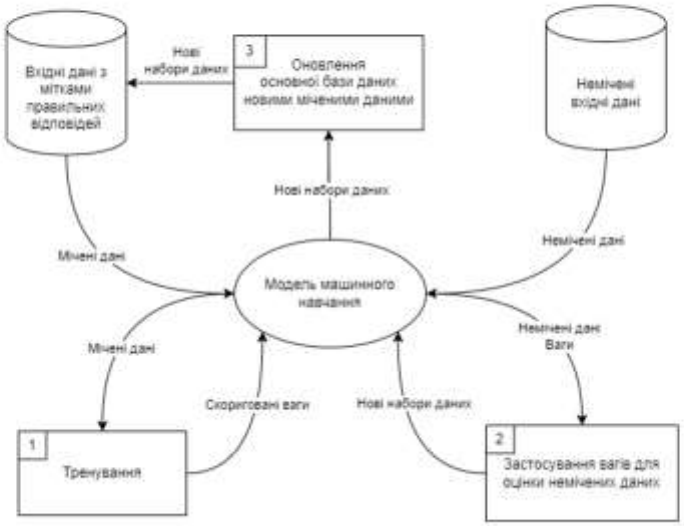
3. Знаходимо фінальну кількість балів, сумуючи зважені бали кожної категорії:

$$\text{Фінальна кількість балів} = \text{Критичність діагнозу} + \text{Життєві показники} + \text{Рівень болю}$$

Фінальна оцінка пріоритетності:
0–2 балів – Пріоритет 5 (стабільний).
2–4 балів – Пріоритет 4 (помірний).
4–6 балів – Пріоритет 3 (терміновий).
6–8 балів – Пріоритет 2 (високий ризик).
8–10 балів – Пріоритет 1 (критичний).

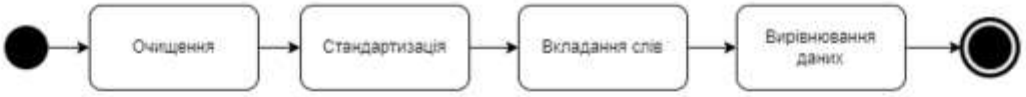
5

ДІАГРАМА ПОТОКУ ДАНИХ В МЕТОДІ НАПІВКОНТРОЛЬОВАНОГО НАВЧАННЯ



6

ДІАГРАМА ПРОЦЕСУ ОБРОБКИ ДАНИХ



7

РЕЗУЛЬТАТ ОБРОБКИ ВХІДНИХ ДАНИХ ПАЦІЄНТІВ

Sex	Age	Arrival mode	Mental	NRS_pain	SBP	DBP	HR	RR	BT	Diagnosis in ED	Disposition	Length of stay_min
2	71	3	1	2	160	100	84	18	36.6	Corneal abrasion	1	86
1	56	3	1	2	137	75	80	20	36.5	Burn of hand, firts degree dorsum	1	64
1	66	2	1	2	130	80	102	20	36.6	Fracture of surgical neck of humerus	2	862
2	71	1	1	3	139	94	88	20	36.5	Alcoholic liver cirrhosis with ascites	1	108
2	58	3	1	3	91	67	93	18	36.5	Ascites	1	109
1	54	4	1	3	140	80	94	20	38.1	Fever, unspecified	2	9248
2	49	3	1	3	110	70	70	20	36.2	Angina pectoris, unspecified	1	400
2	78	3	1	3	169	86	80	20	36	Acute coronary syndrome	1	247
2	32	3	1	3	140	75	91	20	36.6	Herniated disc disease of lumbar spir	1	59
1	38	3	1	3	130	80	80	20	36.3	Ocular pain	1	185
2	43	3	1	3	148	104	72	16	36.5	Acute gastritis	1	176
2	45	3	1	3	141	81	88	18	36.6	Gout site unspecified	1	45

Sex	Age	Arrival mode	Mental	NRS_pain	SBP	DBP	HR	RR	BT	Disposition	Length of s	embedding_0
1	0.375	-0.333333333	-1	-0.6	-0.09524	0.069767	-0.10345	-0.5	-0.46667	-1	-0.99976	0.20889321
-1	2.22E-16	-0.333333333	-1	-0.6	-0.31429	-0.31783	-0.51724	-0.25	-0.5	-1	-0.99982	0.548816919
-1	0.3	-0.666666667	-1	-0.6	-0.38095	-0.24031	0.206897	-0.25	-0.46667	-0.66667	-0.99757	0.687682271
1	0.375	-1	-1	-0.4	-0.29524	-0.02326	-0.03448	-0.25	-0.5	-1	-0.9997	0.362955242
1	0.05	-0.333333333	-1	-0.4	-0.75238	-0.44186	0.051724	-0.5	-0.5	-1	-0.99969	0.419940889
-1	-0.05	0	-1	-0.4	-0.28571	-0.08527	0.068966	-0.25	0.033333	-0.66667	-0.97394	0.8370592
1	-0.175	-0.333333333	-1	-0.4	-0.57143	-0.39535	-0.34483	-0.25	-0.6	-1	-0.99887	0.468864739
1	0.55	-0.333333333	-1	-0.4	-0.00952	-0.14729	-0.17241	-0.25	-0.66667	-1	-0.9993	-0.068866514
1	-0.6	-0.333333333	-1	-0.4	-0.28571	-0.31783	0.017241	-0.25	-0.46667	-1	-0.99983	0.296072453
-1	-0.45	-0.333333333	-1	-0.4	-0.38095	-0.24031	-0.17241	-0.25	-0.56667	-1	-0.99948	0.203708523
1	-0.325	-0.333333333	-1	-0.4	-0.20952	0.131783	-0.31034	-0.75	-0.5	-1	-0.9995	0.296505302
1	-0.275	-0.333333333	-1	-0.4	-0.27619	-0.22481	-0.06897	-0.5	-0.46667	-1	-0.99987	0.858306289
-1	-0.075	-0.333333333	-1	-0.4	-0.47619	-0.24031	0.275862	-0.25	-0.6	-1	-0.97726	0.316215783
1	-0.025	-0.333333333	-1	-0.4	-0.46667	-0.27132	-0.17241	-0.25	-0.53333	-0.66667	-0.99939	0.381089121

8

РЕЗУЛЬТАТ РОБОТИ СИСТЕМИ

Вхідні дані пацієнтів:

```
Input data:
  Sex  Age  Arrival mode  Mental  NRS_pain  SBP  DBP  HR  RR   BT                Diagnosis in ED
    1   54            4       1         3  140   90  94  20 38.1             Fever, unspecified
    1   56            3       1         2  137   75  68  20 36.5 Burn of hand, firts degree dorsum
    2   78            3       1         3  169   86  88  20 36.8          Acute coronary syndrome
```

Відповідні оцінки пріоритетності:

```
Priority prediction #1:  [3.6718473]
Priority prediction #2:  [5.0061874]
Priority prediction #3:  [2.1630125]
```

9

ПОРІВНЯЛЬНИЙ АНАЛІЗ РЕЗУЛЬТАТІВ

Система	Час оцінки пацієнта (хв)	Точність (%)
MTS	2 – 5	70 – 84
ESI	3 – 6	77 – 80
START	0.5 – 1	50 – 62
SATS	5 – 8	74 – 78
СМНОСП	1 – 3	80 – 84

10

ВИСНОВКИ

1. Проаналізовано системи оцінки пріоритетності пацієнтів: MTS, ESI, START, SATS. В тому числі такі їх параметри як час на оцінку пацієнтів та точність цих оцінок, для порівняння із розробленою системою;
2. Проаналізовано методи машинного навчання: з вчителем, без вчителя, з підкріпленням та напівконтрольоване. Для реалізації системи було обрано метод напівконтрольованого навчання через свою здатність видавати високу точність при невеликій кількості даних для навчання;
3. Реалізовано систему для оцінювання пріоритетності пацієнтів. За результатами тестування системи було виявлено, що вона є швидшою ніж більшість проаналізованих систем. Оцінка пацієнта за розробленою системою займає 1-3 хв, єдина система, що має кращі результати є START зі значеннями 0.5-1 хв. Також система показує високі значення точності у 80-84%, що в середньому на 7% вище за таку точну систему як MTS.

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Спіцин А.Я. Визначення пріоритетності пацієнтів у відділенні невідкладної допомоги як задача машинного навчання / Спіцин А.Я., Садовенко В.С. // Виклики та рішення в програмній інженерії: Матеріали всеукраїнської науково-технічної конференції. Збірник тез 26.11.2024, ДУІКТ, м. Київ, 2024. – подано до друку.

2. Спіцин А.Я. Використання моделі на основі напівконтрольованого машинного навчання для визначення пріоритетності пацієнтів лікарні / Спіцин А.Я., Садовенко В.С. // Виклики та рішення в програмній інженерії: Матеріали всеукраїнської науково-технічної конференції. Збірник тез 26.11.2024, ДУІКТ, м. Київ, 2024. – подано до друку.