

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Системотехніки  
(повна назва)

**АТЕСТАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

ГЮИК 50 5400.014 ПЗ  
(позначення документа)

Проектування системи підтримки прийняття рішень лікарем-терапевтом  
(тема)

Виконав:

студент 2 курсу, групи СПРМ-19-1  
Хорщунова К.Р.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування  
(повна назва спеціалізації)

Керівник проф. Овезгельдієв А.О.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

проф. Гребінник І. В.  
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
(повна назва)  
Кафедра \_\_\_\_\_ Системотехніки \_\_\_\_\_  
(повна назва)  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)  
Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)  
Освітня програма \_\_\_\_\_ Системне проектування (СПР) \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«» \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА АТЕСТАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Хоршуновій Карині Романівні \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Проектування системи підтримки прийняття рішень лікарем-терапевтом \_\_\_\_\_

затверджена наказом університету від 30 жовтня \_\_\_\_\_ 2020 р. № 1497 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 22 грудня \_\_\_\_\_ 2020 р.

3. Вихідні дані до роботи Науково-технічні публікації, Інтернет-ресурси за темою, дані статей, публікації в наукових журналах, \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Аналіз предметної області \_\_\_\_\_

2) Опис системних та функціональних вимог до системи підтримки прийняття рішень \_\_\_\_\_

3) Опис алгоритму прийняття рішення. Метод опорних векторів \_\_\_\_\_

4) Розробка програмного засобу та аналіз результатів \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) **5.1** Архітектура клінічної системи підтримки прийняття рішень (1 аркуш формату А4). **5.2** Контекстна діаграма функції «Автоматизація процесу прийняття рішення лікарем-терапевтом» (1 аркуш формату А4). **5.3** Діаграма декомпозиції функції Автоматизація процесу прийняття рішення лікарем-терапевтом» (1 аркуш формату А4). **5.4** Діаграма декомпозиції функції «Аналіз електронної медичної картки» (1 аркуш формату А4). **5.5** Діаграма декомпозиції функції «Формування рекомендації» (1 аркуш формату А4). **5.6** Діаграма варіантів використання для ролі «Адміністратор» (1 аркуш формату А4). **5.7** Діаграма варіантів використання для ролі «Лікар» (1 аркуш формату А4). **5.8** Діаграма варіантів використання для ролі «Пацієнт» (1 аркуш формату А4). **5.9** Логічна модель бази даних (1 аркуш формату А4). **5.10** Фізична модель бази даних (1 аркуш формату А4). **5.11** Графік залежності точності від функції ядра (1 аркуш формату А4).

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Овезгельдієв А.О.		

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	01.11.2020	виконано
2	Аналіз предметної галузі та постановка задачі	02.11.2020 – 07.11.2020	виконано
3	Дослідження методів прийняття рішень	07.11.2020 – 12.11.2020	виконано
4	Аналіз існуючих проблем даної предметної області	12.11.2020 – 17.11.2020	виконано
5	Визначення вимог до розроблюваної системи	17.11.2020 – 19.11.2020	виконано
6	Проектування системи	19.11.2020 – 23.11.2020	виконано
7	Розробка методу прийняття рішення	23.11.2020 – 25.11.2020	виконано
8	Тестування і аналіз результатів	25.11.2020 – 27.11.2020	виконано
9	Оформлення пояснювальної записки	27.11.2020 – 18.12.2020	виконано
10	Попередній захист	19.12.2020	виконано
11	Захист перед ЕК	22.12.2020	

Дата видачі завдання 01   11   2020 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Овезгельдієв А.О. \_\_\_\_\_  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи: 78 с., 3 табл., 30 рис., 2 дод., 26 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, КЛАСИФІКАЦІЯ, МЕТОД ОПОРНИХ ВЕКТОРІВ, СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, NODE.JS, PYTHON.

Об'єктом дослідження є процес прийняття рішення у медичних системах підтримки прийняття рішень.

Предметом дослідження є система підтримки прийняття рішень лікарем-терапевтом.

Метою атестаційної роботи є розробка системи підтримки прийняття рішень лікарем-терапевтом на основі методу опорних векторів.

Методами досліджень є аналіз існуючих методів прийняття рішень, методології проектування систем, розробка системи підтримки прийняття рішень у контексті підсистеми клінічної інформаційної системи, вирішення практичних завдань.

Результатом атестаційної роботи є розроблена системи підтримки прийняття рішень лікарем-терапевтом на основі методу опорних векторів.

## **ABSTRACT**

Explanatory note: 78 p., 3 tab., 30 fig., 3 app., 21 sources.

**CLASSIFICATION, INFORMATION SYSTEM, NODE.JS, PYTHON, SUPPORT DECISION SYSTEM, SUPPORT VECTOR MACHINE.**

The object of the research is the process of making decision in a clinical support decision system.

The subject of the research is a clinical decision support system.

The purpose of the certification work is to implement a clinical support decision system using support vector machine algorithm.

Research methods include the analysis of existing decision-making methods, systems design methodologies, modeling the system as a part of clinical information systems, solving practical problems.

The result of the certification work is implementation of a clinical support decision system using support vector machine algorithm.

## ЗМІСТ

ПЕРЕЛІК СКОРЕЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1 Клінічні системи підтримки прийняття рішень. Загальна архітектура. Етапи моделювання та основні вимоги до системи. ....	9
1.2 Джерела даних для систем підтримки прийняття лікарських рішень. Обробка даних. ....	16
1.3 Класифікація систем підтримки прийняття лікарських рішень. ....	21
1.4 Постановка задачі .....	29
2 ОПИС СИСТЕМНИХ ТА ФУНКЦІОНАЛЬНИХ ВИМОГ ДО СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ .....	30
2.1 Опис об'єкту проектування та визначення основних системних вимог .....	30
2.2 Визначення функціональних вимог до системи.....	32
3 ОПИС АЛГОРИТМУ ПРИЙНЯТТЯ РІШЕННЯ. МЕТОД ОПОРНИХ ВЕКТОРІВ	42
4 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	52
4.1 Вибір архітектури системи.....	52
4.2 Обґрунтування вибору СУБД .....	54
4.3 Вибір мови програмування.....	62
4.4 Опис обраних технологій розробки та програмних засобів .....	64
4.5 Огляд клієнтського інтерфейсу розроблюваної системи .....	64
4.6 Навчання методу прийняття рішення та аналіз результатів його роботи .....	71
ВИСНОВКИ.....	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....	75
ДОДАТОК А Графічний матеріал атестаційної роботи .....	78
ДОДАТОК Б Текст програми.....	92
ДОДАТОК В Відомість атестаційної роботи .....	97

## ПЕРЕЛІК СКОРЕЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІС – інформаційна система;

СУБД – система управління базами даних;

СППР – система підтримки прийняття рішень;

КСППР – клінічна система підтримки прийняття рішень;

EHR – electronic health record (електронний медичний запис);

ER-модель – entity-relationship model (модель сутність-зв'язок);

FK – foreign key (зовнішній ключ);

IDEF0 – методологія функціонального моделювання для формалізації бізнес-процесів;

PK – primary key (первинний ключ);

SVM – support vector machine.

## ВСТУП

Одним із головних завдань лікаря є правильна постановка діагнозу. Сучасні технології, впроваджені в медичну діагностику, дозволяють істотно вплинути на постановку попереднього діагнозу у пацієнтів. Існують різні підходи створення медичних систем діагностики, наприклад, системи підтримки прийняття рішень (СППР), які допомагають вирішувати неформалізовані задачі. Такі задачі націлені на виявлення захворювання пацієнта. Особливість кожного захворювання обумовлена різним набором симптомів і значеннями коефіцієнтів специфічності. В процесі роботи СППР підсумкові оцінки захворювань, що розглядаються, порівнюються з вхідними факторами прояви симптомів і потім робляться висновки про стан області медичної діагностики [1].

Автоматизація отримання логічних висновків застосовується не тільки в медицині, але і в інших сферах. Так імітаційна модель постановки діагнозу стає все більш широкою темою для вивчення і проектування рішень, пов'язаних з різними видами проблем: недостовірністю та малою кількістю інформації діагностики, аргументацією висунутого припущення, а також точністю представленої гіпотези.

В даній роботі увага приділяється саме медичним експертним системам, тому що завдання підтримки прийняття кваліфікованого рішення з приводу диференціального діагнозу є актуальним. Симптоми прояву захворювань досить різні та залежать від багатьох факторів, в зв'язку з цим існують розбіжності щодо діагностики та інтерпретації результатів обстежень. Використання СППР дозволяє визначити попередній діагноз або ж звузити список можливих захворювань враховуючи усі відомі фактори та симптоми, при цьому кінцевий діагноз визначає лікар.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Клінічні системи підтримки прийняття рішень. Загальна архітектура. Етапи моделювання та основні вимоги до системи.

Система підтримки прийняття рішень (СППР) - це комп'ютерна автоматизована система, метою якої є допомога людям, що приймають рішення в окремих умовах для повного та об'єктивного аналізу предметної діяльності. СППР виникли у результаті злиття управлінських інформаційних систем та систем управління базами даних. СППР - це людино-машинні системи, які дозволяють особі, що приймає рішення використовувати дані, знання, об'єктивні та суб'єктивні моделі для аналізу та розв'язання неструктурованих та слабоформалізованих задач [1].

Процес прийняття рішень - це отримання і вибір оптимальної альтернативи з урахуванням усіх наслідків. При виборі альтернативи потрібно обирати ту, яка повністю відповідає поставленій цілі, але при цьому врахувати велику кількість суперечливих вимог і, відповідно, оцінити обраний варіант рішення за багатьма критеріями [1].

Система підтримки прийняття рішень призначена для підтримки багатокритеріальних рішень у складному інформаційному середовищі. При цьому під багатокритеріальністю розуміється той факт, що результати прийнятих рішень оцінюються не по одному, а за сукупністю багатьох показників (критеріїв), що розглядаються одночасно [1].

СППР характеризується наступними відмінними особливостями [2]:

- орієнтацією на вирішення погано структурованих (формалізованих) задач;
- можливістю поєднання традиційних методів обробки комп'ютерних даних з можливостями математичних моделей та методів вирішення задач, що базуються на цих моделях;

- направленістю на не професійного кінцевого користувача системи;
- високою адаптивністю, що забезпечує можливість задовольняти вимоги користувача та особливості наявного технічного та програмного забезпечення.

Система підтримки прийняття рішень вирішує дві основні задачі:

- вибір найкращого рішення з множини можливих;
- впорядкування можливих рішень за перевагою (ранжування альтернатив).

Лікувально-діагностичний процес є зразком цілеспрямованої діяльності в умовах невизначеності. Лікар не завжди може заздалегідь бути обізнаним про результат процесу. Оцінка об'єктивного статусу пацієнта цілком заснована на ознаках захворювання, як клінічних, так і інструментально-лабораторних [3]. Одним з основних факторів, що впливають на якість наданої медичної допомоги, є кваліфікація лікарів, а саме вміння визначати адекватну тактику діагностики та дослідження, а також лікування пацієнта. Одним з вирішальних факторів у цьому процесі є основи пізнавального, послідовного і логічного мислення, що дозволяють зрозуміти глибинні причини виникнення захворювань та оволодіти способами впливу на патологічний процес. Відповідно, основними завданнями лікаря в даній ситуації є розпізнавання діагнозу на основі первинної інформації про пацієнта, характерних ознаках захворювання згідно з результатами комплексного обстеження і шаблонних уявленнях лікаря про дане захворювання. При цьому сам по собі діагноз представляє собою детальну формалізацію дій, що має імовірнісну природу. На практиці ж клінічна ситуація відображає послідовність ведення хворого і природний плин патологічного процесу. Відповідно, формалізація міркувань лікаря, розпізнавання і використання його суджень лежать в основі складання алгоритмів, покликаних сприяти індивідуальному підходу у вирішенні конкретних клінічних завдань. В даному випадку актуальне значення приймають так звані інтелектуальні клінічних системи підтримки прийняття рішень, покликані забезпечити підвищення якості медичного обслуговування в закладах лікувально-профілактичного профілю [3].

Сучасні системи прийняття рішень в клінічній медицині повинні виконувати наступний набір функцій: диференціальна діагностика і вибір лікування,

ефективність вирішення незалежно від ступеня вираженості клінічних проявів (в тому числі при стертій картині захворювання) з урахуванням фонових станів пацієнта, можливість аналізу динаміки патологічного процесу, можливість оцінки стану в режимі реального часу.

Однак прийняття вірних рішень в медицині зазвичай утруднене у зв'язку з множинністю факторів і ознак більшості захворювань і їх взаємодій. Тоді як обробка великої кількості інформації лікарем часто утруднена через ті чи інші об'єктивні і суб'єктивні причини.

Шаблонна модель інтелектуальної системи складається з послідовних кроків діагностики і лікування, починаючи від аналізу вихідних даних про пацієнта до моменту фіксації необхідних результатів. На сьогоднішній день ринок медичних інформаційних систем знаходиться в стадії активного розвитку. Одним з перспективних напрямків, зокрема, є комп'ютеризовані моделі систем підтримки прийняття лікарських рішень. Даний вид медичних інформаційних систем переслідує одну з найважливіших задач в галузі охорони здоров'я – завдання зниження кількості лікарських помилок і підвищення якості медичної допомоги.

Існуючі клінічні системи підтримки прийняття рішень (КСППР) включають в себе спеціалізовані медичні бази даних, бібліографічні інформаційно-пошукові системи, системи обробки медичних даних і т.д. [4]. Функціонально найбільш затребуваними з них є КсППР, орієнтовані на конкретний «електронно-математичний» образ пацієнта, заснований на сукупності клініко-лабораторних даних. Лікар за допомогою даного інструменту отримує підтримку при прийнятті рішень для конкретного пацієнту і по кожній дії пов'язаній з веденням даного пацієнта.

Розглянемо загальну архітектуру клінічні системи підтримки прийняття рішень наведено на рисунку 1.1:

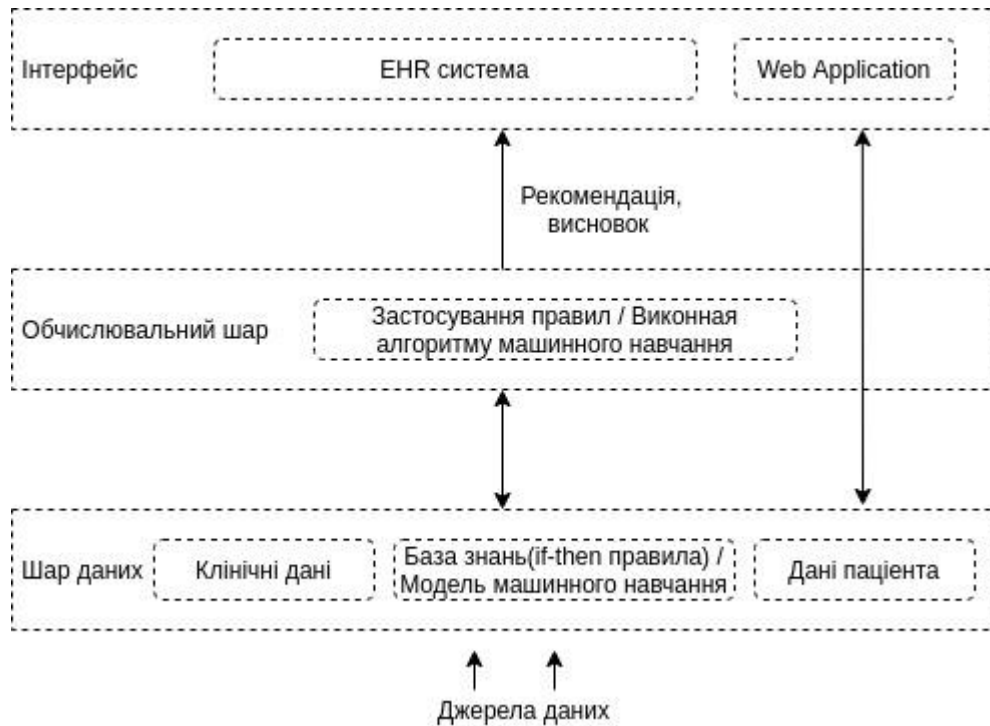


Рисунок 1.1 – Архітектура клінічної системи підтримки прийняття рішень.

Типова архітектура КСППР три основні елементи: базовий або рівень управління даними, механізм обробки або обчислювальний рівень та користувальницький інтерфейс. Рівень управління даними поєднує в собі два джерела даних(клінічні дані та дані пацієнта) і базу знань у вигляді правил, if-then, або моделей машинного навчання. Детальний огляд джерел даних та способів первинної обробки даних наведені у підрозділі 1.2.

Обчислювальний рівень застосовує правила або алгоритми машинного навчання до доступних даних пацієнта та пропонує рішення/рекомендацію. Результати відображаються через рівень користувальницького інтерфейсу - веб додаток або за допомогою інтеграції системи у вже існуючу медичну систему, наприклад, у систему Electronic Health Record (EHR). Електронна медична карта (EHR) - це цифрова версія паперової діаграми пацієнта. EHR - це записи, орієнтовані на пацієнта в режимі реального часу, які надають інформацію миттєво та безпечно авторизованим користувачам [5].

Розглянемо основні етапи розробки системи підтримки прийняття рішень, які наведено на рисунку 1.2. Логічна послідовність дій відображається суцільними

стрілками. Пунктирні стрілки у зворотному напрямку вказують на зміни однієї або декількох фаз, це може стати необхідним під час розробки системи за допомогою механізму зворотного зв'язку.

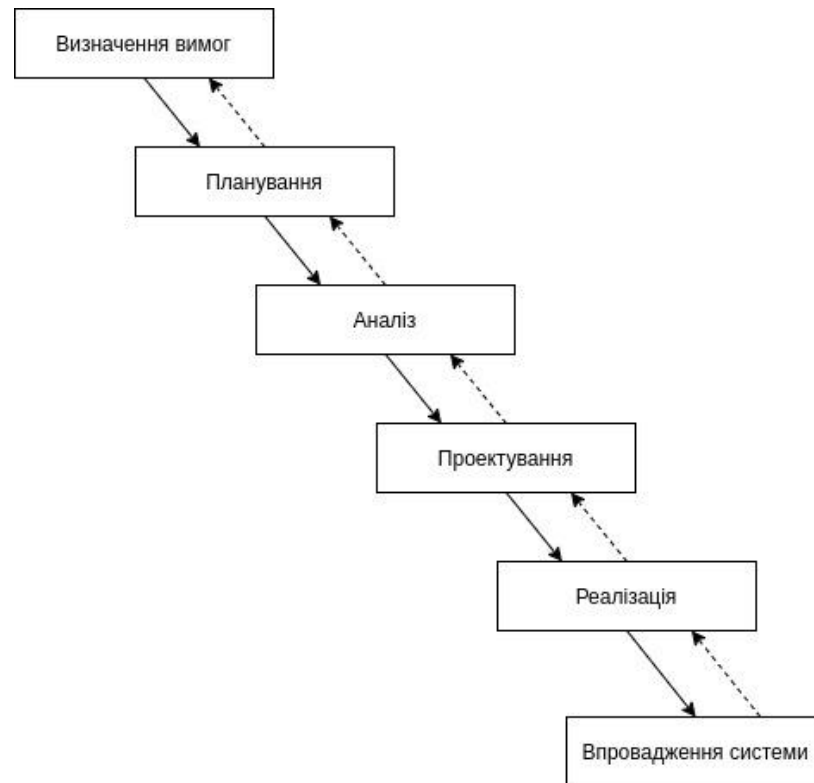


Рисунок 1.2 – Основні етапи розробки СППР

**Планування.** Основна мета етапу планування – зрозуміти потреби та можливості. Зазвичай планування передбачає техніко-економічне обґрунтування, під час якого визначаються загальні та конкретні цілі системи, можливі вигоди, час виконання та витрати. Рішення доцільності продовження розробки системи приймається в кінці етапу планування. У разі негативного рішення розвиток проекту зупиняється, проте рішення може бути змінено після повторення етапу планування.

**Аналіз.** На етапі аналізу визначаються основні функції СППР, які будуть реалізовані, шляхом подальшої розробки та основі попередніх висновків, досягнутих під час техніко-економічного обґрунтування. Для визначення функцій проводиться аналіз процесів прийняття рішень, який дозволяє отримання розуміння всіх взаємозв'язків, що існують між проблемами, що будуть вирішуватися та

навколишнім середовищем. Аналіз також включає в себе відображення фактичних процесів прийняття рішень та уявлення того, як буде виглядати процес прийняття рішень, після створення СППР. Також невід'ємною частиною етапу аналізу є дослідження даних. Виконання такого дослідження дозволяє визначити, скільки і який тип даних вже існує і які дані можна отримати із зовнішніх джерел.

Дизайн. На даному етапі визначається архітектура системи, апаратне забезпечення, структура мережі, база даних та програмні засоби для реалізації системи. Наступним кроком необхідно детально визначити взаємодію системи та користувачів. Основними елементами взаємодії є графічний інтерфейс та друкованих звітів. Також на цьому етапі визначається доцільність залучення третіх сторін до етапу розробки.

Реалізація та впровадження. На даному етапі відбувається безпосередньо реалізація, тестування та впровадження системи. Виконується розробка моделей прийняття рішення, графічного інтерфейсу. Після виконання фази розробки систему необхідно ввести систему в експлуатацію. Послідовність дій на даному етапі залежить від обраного методу розробки. Це можуть бути гнучкі методи розробки, екстремальні методи програмування або метод швидкої розробки прототипів. Швидка розробка прототипів пропонує очевидні переваги. Кожна окрема підсистема може бути розроблена швидше. Таким чином, коли підсистема випускається для користувачів, можна перевірити її відповідність запланованому призначенню та перевірити її функції, навіть якщо вони все ще не повністю розроблені. Крім того, проміжні випробування дозволяють оперативно виправляти більшість помилок проектування. Отже, еволюційний розвиток СППР дозволяє мінімізувати ризик відмови. У ситуації, коли існує явна розбіжність між особливостями розробленого прототипу та сподіваннями користувачів, можливо навіть прийти до раннього переривання проекту.

Подальший аспект фази впровадження, який часто ігнорують, стосується загального впливу на організацію, яка буде користуватися новою системою. Розуміння основних причин, які призводять або до успіху, або до неуспіху, може допомогти підвищити ефективність розробки та впровадження КСППР у

повсякденній практиці. Помилки можуть виникати на різних етапах розробки та впровадження, основні з них: неможливість технічно завершити систему, неможливість отримати систему зрозумілу користувачами; неможливість інтеграції системи в організаційне або користувацьке середовище.

Для того, щоб мати можливість побудувати придатну КСППР, необхідно мати більш широке розуміння прийняття медичних рішень, як це відбувається в природних умовах. Проектування КСППР без розуміння когнітивних процесів, що лежать в основі медичних міркувань та аналізу рішень, призводить до неуспішного впровадження КСППР у повсякденний клінічний робочий процес.

Існує оцінка, що 45% комп'ютеризованих медичних інформаційних систем виходять з ладу через опір користувачів, хоча ці системи є технологічно цілісними. Деякі причини такого високого відсотка відмов можуть бути наслідком недостатньої потужності комп'ютера, зниження професійної автономії, недостатнього усвідомлення довгострокових переваг використання КСППР та відсутності бажання змінювати щоденний робочий процес [6].

З огляду на недавні дослідження можна виділити ряд особливостей та вимог до КСППР, що мають вирішальне значення для успіху цих систем):

– КСППР повинна надавати підтримку прийняття рішень автоматично, тобто бути частиною робочого процесу лікарів. Системи, де спеціалісти повинні були звертатися за порадою вручну, не набувають широкого використання. Підтримка рішень повинна надаватися в реальному часі, без необхідності переривання звичайного режиму допомоги пацієнту. В ідеальному випадку, система підтримки прийняття рішень повинна бути включена у більшу комп'ютерну систему, яка вже є частиною професійної рутини лікарів, роблячи таким чином підтримку прийняття рішень побічним продуктом звичайної практики.

– Система повинна бути зручною для навігації та використання, наприклад із швидким доступом та мінімальними клацаннями миші для отримання потрібної інформації.

– Системи повинна отримувати нові дані автоматично у більшості можливих випадків. Як правило лікарі уникають процесу введення нових даних про пацієнтів,

що призводить до часткового використання системи або до повної відмови від використання

– Час і частота підказок мають велике значення. Наприклад, якщо повідомлень забагато, це може призвести лише до ігнорування всіх повідомлень і, як наслідок, до втрати важливої інформації. Також важливе значення має час - попередження не повинні з'являтися у невідповідний час і переривати робочий процес.

– Результати підтримки прийняття рішень повинні надаватися як клініцистам, так і пацієнтам. Дослідження показали сприятливий ефект таких дій, оскільки вони стимулюють клініцистів обговорювати варіанти лікування з пацієнтами, а отже, змушують останніх почуватись більш залученими до свого лікування.

Підводячи підсумок, можна зазначити, що при розробці КСППР існують і інші фактори, крім програмного забезпечення, які необхідно враховувати. Основні з цих факторів: використання апаратного забезпечення, що задовольняє вимоги системи, достатня та своєчасна технічна підтримка системи, непомітна інтеграція системи у клінічний робочий процес, актуальність та своєчасність надання рішень.

1.2 Джерела даних для систем підтримки прийняття лікарських рішень.  
Обробка даних.

Як було зазначено вище одним з основних елементів КСППР є рівень даних. Розробка кожної системи підтримки прийняття починається з визначення джерел даних. Дані надходять у багатьох формах та наборах, розглянемо основні з них.

Значна частина повсякденних клінічних рішень базується на неструктурованих записах у вільному тексті, що охоплюють, наприклад, історію хворого та спостереження лікарів під час фізичного огляду або регулярні записки медсестер. Незважаючи на те що у сучасних системах електронних медичних карт (EHR), неструктуровані вільні тексти клінічних записів не вважаються найбільш

прийнятною формою даних, вони все ще надають незамінну інформацію та контекст медичним працівникам. Використання вільного тексту створює низку проблем. Окрім очевидних, таких як стиль письма та друкарські помилки, медичний текст є специфічним для конкретної людини і може бути дуже упередженим. Специфічність тексту варіюється від мовних та національних аббревіатур до жаргонних відмінностей між двома відділеннями однієї лікарні, так як для людей притаманно скорочувати слова та придумувати більш легкі для запам'ятовування терміни. Зазвичай такий процес відбувається під час спілкування між медичними працівниками або коли медичні працівники роблять особисті нотатки, щоб зафіксувати хід своїх думок. Це призводить до того, що дана фраза, термін, аббревіатура чи аббревіатура є специфічними для контексту та мають різне значення в різних ситуаціях. Отже, тлумачення вільного тексту в значній мірі залежить від контекстних знань експертів.

Іншим видом даних є структуровані дані. До таких даних можна віднести, числові дані (наприклад, лабораторні вимірювання та артеріальний тиск) та категоріальні дані (наприклад, гіпертонія так / ні), найпростіші в роботі.

Дані, що використовуються у роботі КСППР можна умовно поділити на види – це джерело клінічних знань (clinical knowledge source) та джерело даних (data source). Клінічні знання отримуються з високоякісних джерел таких як рандомізовані контрольовані дослідження, медичні реєстри, систематичні огляди, національні або професійні рекомендації та / або з участі клініцистів, які в кінцевому підсумку будуть використовувати цю систему. Як правило дані, зібрані в ході дослідження, важливі для вирішення заздалегідь обраного питання. Зазвичай ці дані мають високу якість та зберігаються з великою кількістю деталей і часто забезпечені коментарями з використанням експертних знань (діагностичні коди і заздалегідь виділені класифікації тяжкості захворювання). Медичні реєстри розробляються для контролю якості та дослідницьких цілей. Вони використовуються для запису заздалегідь визначеного обмеженого числа ознак для певної групи пацієнтів, часто з наголосом на конкретні стани і діагнози. Ретельне ведення дослідницьких баз даних і реєстрів дозволяє збирати дані про пацієнтів

чистим і систематичним чином відповідно до протоколу, максимально запобігаючи втраті тих чи інших даних для подальшого спостереження. Однак через обмеженість та виділення фіксованого числа ознак дослідницькі дані і медичні реєстри не враховують цінну контекстну інформацію, таку як текстові замітки, для кожного пацієнта, що приймає участь у дослідженні. Таким чином, отримані дані показують обмежений, заздалегідь визначений обсяг ознак стану пацієнта.

Дослідницькі бази даних та медичні реєстри використовують досить штучний, «сухий» спосіб збору даних, який підходить для теоретичних досліджень, тому такі джерела даних не дозволяють створити широко використовувану КСППР. Таким чином для створення робочих КСППР у своїй більшості використовується джерела даних, та як вони містять містять реальні дані з клінічної практики. Джерело даних формується на основі даних конкретного пацієнта, які можна отримати з системи комп'ютеризованого введення призначення лікування (CPOE - Computerized provider order entry systems), медичного інструменту (наприклад, приладу для вимірювання артеріального тиску), EMR (Electronic Medical Record) або іншого сховища даних. Дані також можуть бути зібрані з твердих джерел інформації таких як, паперова медична карта або отримані шляхом збору анамнезу зі слів пацієнта. У цьому випадку дані повинні бути введені в систему за допомогою посередника введення даних. У такому випадку посередником джерела даних виступає лікар, який вводить дані у джерело даних. У деяких випадках посередниками також можуть бути самі пацієнти. Проте кращим варіантом отримання даних вважається автоматичне комп'ютеризоване надання даних системі. На практиці це полегшує процес введення у експлуатацію та подальше використання КСППР. Проте слід зазначити, що системи EMR були розроблені як віртуальна карта пацієнта, що не обов'язково передбачає повторне використання даних. Таким чином, перетворення даних отриманих з EMR в цінні набори даних вимагає ретельної і кваліфікованої обробки даних [7]. У таблиці 1 представлено порівняння різних типів клінічних даних за різними ознаками, важливими для розробки КСППР.

Таблиця 1.1 – Порівняння типів клінічних даних.

	Електронний медичний запис, вільний текст, неструктуровані дані	Медичні реєстри, дослідження	Структуровані дані (числові, категоріальні)
Повнота контексту	За потреби можна включити контекстну інформацію.	Контекст по суті відсутній, обирається тільки обмежено число досліджуваних ознак	Контекст може бути втрачений через наперед визначену категоризацію.
Машиночитабельність	Дані потребують перетворення у формат, що підходить для комп'ютерів	Дані мають однаковий формат і можуть бути проаналізовані комп'ютерам	Дані можуть бути проаналізовані або безпосередньо використані комп'ютерами.
Можливість використання даних, між різними медичними установами	Вільний текст містить специфічні терміни, скорочення притаманні конкретній установі або медичному працівнику	Дані випробувань зазвичай збираються за допомогою стандартизованого протоколу.	Структуровані дані, можуть бути легко переформатовані для сумісності.

## Продовження таблиці 1.1

	Електронний медичний запис, вільний текст, неструктуровані дані	Медичні реєстри, дослідження	Структуровані дані (числові, категоріальні)
Стійкість до шуму	Дані цього типу дуже чутливі до шуму. Шумом для цих даних можна вважати: особисті скорочення, орфографічні помилки, особистий стиль ведення записів.	Дані записуються стандартизованим способом, призначеним для запобігання шуму	Дані часто створюються на комп'ютері або записуються стандартизованим способом.

Не менш важливим етапом після отримання даних є їх попередня обробка. На етапі попередньої обробки з необроблених даних вилучаються змінні(ознаки), які у майбутньому буде використовувати модель. На даному етапі важливо залучати медичних спеціалістів для визначення більш значущих змінних. Також на цьому етапі виконується заповнення пропущених значень, викидів та обробка шуму, це може бути зроблено автоматично використовуючи відомі способи, так як заповнення пропусків нулями, середнім значенням т.д., проте для реальних систем таких вид обробки можливий тільки після визначення значущості ознаки у конкретному випадку, тому іншим способом обробки даних є їх обробка медичним спеціалістом, проте на великих датасетах така обробка займає багато часу. Найбільш розповсюдженим є метод комп'ютерної обробки з залученням експерта, який аналізує ознаки, їх значущість та можливі значення цих ознак, проте основна робота виконується автоматично. Також, якщо це можливо, рекомендується кодування даних з використанням широко відомих схем кодування, наприклад МКБ-10, SNOMED.

### 1.3 Класифікація систем підтримки прийняття лікарських рішень.

Наступним етапом після вилучення та обробки даних є побудова моделі. Розробка моделі зазвичай містить фазу навчання моделі та фазу тестуванням. На етапі навчання розробляється модель, яка найкраще відповідає даним (тобто робить найкращі прогнози на основі даних навчання), а на фазі перевірки проводяться тести, щоб перевірити, чи модель на нових даних, що не входили до тренувального набору.

КСППР можна розділити на системи, що базуються на знаннях та на системи, що базуються на алгоритмах машинного навчання.

Системи, що базуються на знаннях, як правило складаються з трьох частин — бази знань, механізму умови-виведення та механізму зв'язку. Вони містять експертні клінічні знання про цілком конкретні факти та завдання і здатні міркувати із введенням даних із усіх різних джерел, зазначених вище.

Серед таких систем можна виділити два основні типи: системи, засновані на правилах та експертні системи на основі мереж Баяєса. Система, що базується на правилах, використовує різні бази знань експертів у формі виразів, які можна оцінити як правила IF-THEN (виробничі правила). Приклад правил: якщо рівень глюкози більше ніж 11 ммоль / л, то ймовірно у пацієнта цукровий діабет; якщо у пацієнта порушена функція почек, то не дозволяється призначати метформін. Така система є прикладом евристичного підходу, при якому окремі логічні висловлювання отримуються шляхом опитування експертів або формуються на основі їх спостережень. Цей підхід було вперше застосовано в MYCIN (Shortliffe 1976) з метою вибору відповідної антимікробної терапії для пацієнта. Механізм умови-виведення поєднує та співвідносить правила бази знань з даними пацієнта і на основі цих співвідношень робить висновок.

Мережа Баяєса. Типовою системою прийняття рішень, заснованою на знаннях, є мережа Баяєса (відома як байєсівська мережа довіри або причинно-імовірнісна мережа), яка показує імовірнісні взаємозв'язки між наборами змінних -

хворобами та симптомами на основі умовної ймовірності згідно за теоремою Байєса. Це мережа з чіткою вимогою, щоб стосунки були причинними. Така мережа допомагає моделювати прогресування захворювання з часом та взаємодію між ними; Однак великим перешкодою є те, що у медичних знаннях з часом буває важко визначити, що є наслідком, а що - причиною.

Доступність машинного навчання пропонує нові підходи для розробки медичних моделей і оцінок ризику. Машинне навчання відноситься до групи статистичних методів, які можна використовувати для розпізнавання складних закономірностей. Дані моделі використовують ітеративний процес навчання і створюють модель на основі вивчених даних. Основні типи систем, що не базуються на знаннях, використовують генетичні алгоритми та штучні нейронні мережі, незважаючи на це виділяють низку інших алгоритмів, що використовуються для побудови моделей у КСППР.

Алгоритми машинного навчання можна розділити на 2 типи: алгоритми навчання з вчителем (supervised learning) та без вчителя (unsupervised learning).

Навчання з вчителем передбачає, передбачає, що класи заздалегідь відомі та для кожного класу є набір прикладів. Набір даних, який використовується для навчання, називається навчальною вибіркою. Навчальна вибірка складається з залежних або цільових змінних, а також незалежних змінних або вхідних даних. Система налаштовується на основі навчальної вибірки і сигналу помилки (різниці між бажаною реакцією і фактичної реакцією системи). Мета цього підходу - встановити взаємозв'язок або модель прогнозування між залежними і незалежними змінними. При розробці інструментів інтелектуального аналізу даних і систем підтримки клінічних рішень, які використовують статистичні підходи, подібні описаним тут, однією з найважливіших задач є створення моделі класифікації, відомої як класифікатор, яка буде передбачати клас на основі значення вхідних атрибутів. До відомих методів, що широко використовуються в інтелектуальному аналізі даних, належать дерева рішень, логістична регресія, класифікація на основі сусідів, нейронні мережі.

Дерева рішень. Використання дерев рішень є, одним з найпростішим для розуміння і найбільш широко використовуваним методом, який відноситься до категорії навчання з учителем [8]. Дерева рішень використовує стратегію пошуку рішення зверху вниз. На кожному вузлі алгоритм перевіряє всі атрибути та всі значення кожного атрибуту щодо визначення атрибута та значення атрибута, яке «найкраще» поділяє дані на більш однорідні підгрупи щодо цільової змінної. Іншими словами, кожен вузол є питанням класифікації, а гілки дерева є розділами набору даних на різні класи. Цей процес повторюється ітеративним чином, поки подальший поділ даних стане неможливим або до кожного члена похідних підгруп може бути застосована лише одна класифікація. Таким чином термінальні вузли дерева рішень представляють різні класи. Приклад дерева рішень наведено на рисунку 1.3.

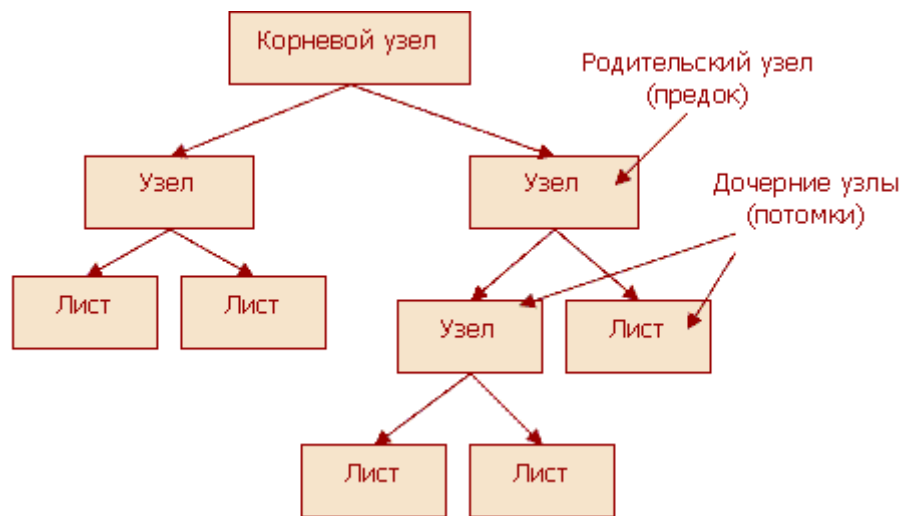


Рисунок 1.3 – Дерево рішень

Логістична регресія. Логістична регресія використовується для моделювання даних, в яких цільова або залежна змінна є бінарною, тобто залежна змінна може приймати значення 1 з імовірністю успіху  $p$ , або значення 0 з імовірністю відмови  $1 - p$ . Основна мета полягає в розробці моделі, що пов'язує бінарну змінну з незалежними змінними. Також логістична регресія може бути використана для вивчення змін залежної змінної, які можна пояснити незалежними змінними, для

ранжирування незалежних змінних на основі їх відносної важливості у прогнозуванні цільової змінної та для визначення ефектів взаємодії між незалежними змінними. Замість прогнозування значень залежної змінної, логістична регресія оцінює ймовірність того, що залежна змінна матиме задане значення. Наприклад, замість того, щоб передбачити, чи страждає пацієнт на певне захворювання, логістична регресія намагається оцінити ймовірність того, що пацієнт має захворювання. Якщо передбачувана ймовірність перевищує 0,5, то ймовірність того, що пацієнт має захворювання, більша за ймовірність відсутності захворювання. Функція, що стосується ймовірності незалежних змінних, є нелінійною функцією, графік цієї функції представлено на рисунку 1.4.

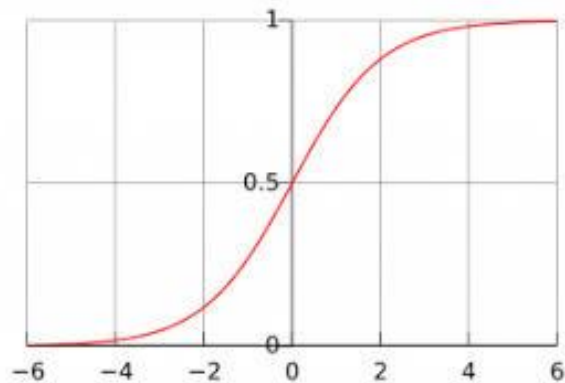


Рисунок 1.4 – Логістична крива

Класифікація на основі сусідів. Класифікація на основі сусідів - це тип навчання на основі екземплярів або не узагальнюючого навчання, він не намагається побудувати загальну модель, а просто зберігає екземпляри навчальних даних. Класифікація проводиться за допомогою розрахунку відстані між прикладами та голосування. Вхідному прикладу присвоюється клас даних, який має найбільшу кількість представників серед найближчих сусідів цього прикладу. k-найближчих сусідів є найбільш часто використовуваною технікою. Оптимальний вибір значення k залежить від вхідних даних. Велике значення k придушує ефекти шуму, проте робить межі класифікації менш чіткими. У випадках, коли дані не мають рівномірної вибірки, кращим вибором може бути класифікація сусідів на основі

радіусу. Класифікатор найближчих сусідів на основі радіуса - це розширення алгоритму класифікації k-найближчих сусідів.

Нейронні мережі. Штучна нейронна мережа - це адаптивна КСППР, що не базується на знаннях і використовує машинне навчання для вивчення досвіду та розпізнавання закономірностей у клінічних даних [9]. Нейронна мережа імітує структуру і властивості організації нервової системи живих організмів. Для цього моделі нейронних мереж використовують мережу зважених орієнтованих графів, в яких вузли є штучними нейронами, а спрямовані ребра є зв'язками між виходами нейрона і входами нейрона. Нейронна мережа отримує на вхід набір сигналів і на виході видає відповідний результат (вихідний сигнал), який описує рішення деякої задачі. Такі системи розпізнавання мають можливість вивчати складні нелінійні відносини введення-виведення, використовувати послідовні процедури навчання і пристосовуватися до даних[10]. У нейронних мережах, які використовують навчання з учителем приклади в формі навчальних даних подаються на вхід мережі по одному. Для кожного прикладу мережа генерує вихідний сигнал, який порівнюється з фактичною цільовою змінною. Після того, як вихідні дані нейронної мережі збігаються з фактичним значенням, навчання вважається завершеним. Якщо вихідне значення відрізняється від фактичного значення, мережа коригує ті ваги. Даний процес є ітеративним і завершується по досягненню конкретної точності[11].

При навчанні без вчителя система представлена набором даних, проте інформація, як згрупувати дані в класи, недоступна. Метою такого навчання є знайти зв'язки між окремими даними, виявити закономірності, підібрати шаблони, упорядкувати дані або описати їх структуру, виконати кластеризацію даних. Головна відмінність кластеризації від класифікації полягає в тому, що перелік груп чітко не заданий і визначається в процесі роботи алгоритму. Алгоритми кластеризації можна розділити на такі групи: *partitional clustering*, ієрархічна кластеризація та *density-based clustering* [12].

*Partitional Clustering* ділить дані на групи, що не перекриваються. Іншими словами, жоден об'єкт не може бути членом більш ніж одного кластера, і кожен кластер повинен мати принаймні один об'єкт. Ці методи вимагають від користувача

вказати кількість кластерів. Багато алгоритмів кластеризації розділів працюють за допомогою ітераційного процесу, щоб розподілити підмножини точок даних у  $k$  кластерів. Найвідомішими алгоритмами даної підгрупи є  $k$ -means and  $k$ -medoids. Алгоритм  $k$ -means - це ітераційний алгоритм, який намагається розділити набір даних на задану кількість окремих кластерів, де кожна точка даних належить лише одній групі. Точка відноситься до кластеру таким чином, що сума квадратної відстані між точками даних та центроїдом кластера (середнє арифметичне всіх точок даних, що належать до цього кластера) є мінімальною.

Ієрархічна кластеризація визначає призначення кластера шляхом побудови ієрархії. Це реалізується або підходом знизу вгору, або зверху вниз: агломеративна кластеризація - це підхід знизу вгору. Даний метод об'єднує дві точки, які є найбільш схожими, поки всі точки не будуть об'єднані в єдиний кластер. Divisive clustering - це підхід зверху вниз. Напочатку усі дані поміщаються у один кластер на кожному кроці кластери розбиваються на менші кластери, поки не залишаються лише окремі точки даних. Ці методи створюють деревоподібну ієрархію точок, що називається дендрограмою. Подібно до роботи *partitional clustering*, в ієрархічній кластеризації кількість кластерів часто визначається користувачем. Кластери призначаються шляхом вирізання дендрограми на заданій глибині, що призводить до  $k$  груп менших дендрограм. На відміну від багатьох методів кластеризації розділів, ієрархічна кластеризація є детермінованим процесом, тобто призначення кластера не змінюватимуться, якщо двічі або більше виконати алгоритм на одних і тих же вхідних даних.

*Density-based clustering* визначає кластери на основі щільності точок даних у регіоні [13]. Кластери формуються там, де є великі щільності точок даних, розділених регіонами низької щільності. На відміну від інших категорій кластеризації, цей підхід не вимагає від користувача вказувати кількість кластерів [14]. Натомість існує параметр, заснований на відстані, який діє як регульований поріг. Цей поріг визначає, наскільки близькими повинні бути точки, щоб вважатися членом кластера. Найвідомішим представником даної підгрупи є алгоритм DBSCAN. В основі методу кластеризації DBSCAN лежить об'єднання деяких

об'єктів відповідно до їх так званих внутрішньогрупових зв'язків. Для проведення коректної процедури кластеризації необхідно вказати критерії, за якими точки будуть об'єднані в кластери. Перш за все, необхідно зазначити, що кластери являють собою щільні області точок у просторі даних, розділених між собою точками, щільність яких значно нижче. Розташування точок в одному кластері обумовлено існуванням зв'язка між ними. Щільність точок визначається двома параметрами: радіусом сусідства та мінімальною кількістю точок, які розташовані найближче до даної точки згідно з визначеним радіусом сусідства.

Новим підходом для реалізації систем підтримки прийняття лікарських рішень є застосування генетичних алгоритмів. Фундаментальна концепція генетичних алгоритмів сягає своїм корінням в еволюційні теорії Дарвіна про природний відбір та адаптацію. Грунтуючись на цій теорії генетичний алгоритм спочатку намагається вирішити загальну проблему випадковими рішеннями. Отримане рішення називають геномом або колекцією генів. Ген являє собою найменшу одиницю інформації для побудови можливого рішення. Наступним кроком є оцінка або кількісне визначення придатності всіх доступних геномів або рішень на основі функції пристосування. Ті рішення, які мають кращі результати, оцінюються вище серед інших і їм дозволено «розмножуватися». Після завершення початкової оцінки генетичні алгоритми вивчають нові рішення, дозволяючи всім отриманим рішенням «еволюціонувати» шляхом взаємного обміну «генетичними матеріалами» і таким чином отримувати нові рішення. Цей процес повторюється доти, поки не буде знайдено оптимальне рішення. Існує багато переваг генетичних алгоритмів. Однією з головних переваг є те, що генетичний алгоритм майже завжди гарантує отримання рішення проблем. Крім того, остаточне рішення часто виходить за рамки початково можливих рішень. Ще однією перевагою є те, що генетичні алгоритми, як правило, знаходять до рішення набагато швидше, ніж інші методи оптимізації. Однак, незважаючи на всі вище описані переваги генетичних алгоритмів, їх ефективність залежить від таких параметрів: розмір популяції рішення та швидкість мутацій. Якщо розмір популяції рішення занадто малий, генетичний алгоритм, може вичерпати всі доступні рішення, до того як буде визначено оптимальне рішення.

Принципово новим підходом у вирішенні клініко-діагностичних завдань відрізняються так звані гібридні системи, засновані на поєднанні методів виведення по прецедентах і правилам [15]. Описувана система інтерактивна і дозволяє здійснювати лікувально-діагностичний процес в умовах дефіциту тимчасових ресурсів, коли відсутня можливість повного опису об'єкта. Новизна даних систем полягає в можливості інтеграції знань про предметну область, отриманих методами класифікації та кластеризації, в механізм вироблення рішення по правилам, об'єднуючи ці два підходи.

Згідно останніх досліджень СППР також поділяють відповідно до методів, що вони використовують [16]:

1. Машинне навчання: ця група включає СППР, що використовують штучні нейронні мережі, support vector machine (SVM) та логістичну регресію.

2. Представлення знань: ці методи стосуються представлення знань та фактів, отриманих в результаті клінічної експертизи, для створення мови опису, що зрозуміла для машин. Ця група включає системи на основі онтології та нечіткі логічні системи.

3. Візуалізація інформації: ці методи використовують алгоритми візуалізації для кодування абстрактних понять та інформації. Такі системи дозволяють операторам візуально перевіряти результати свого рішення.

4. Text mining: ці методи певною мірою використовують логіку контент-аналізу, щоб вилучення важливої інформації з неструктурованих текстів за допомогою машинного навчання, лінгвістичних та статистичних стратегій. Ця група включає пошук інформації та обробку природних мов.

5. Багатоцільові СППР: такі системи об'єднують різні ознаки, параметри, атрибути та характеристики існуючих доменів та категорій для спрощення процесу прийняття рішення. Ця група включає дерева рішень та байєсівську логіку.

## 1.4 Постановка задачі

Завданням атестаційної роботи є моделювання та розробка системи підтримки прийняття рішень лікарем-терапевтом.

Для досягнення поставлено мети необхідно розглянути наступні питання:

- проаналізувати існуючі проблеми в даній області та визначити шляхи їх вирішення;
- розробити системні та функціональні вимоги до системи підтримки прийняття рішень, що розробляється;
- розробити діаграму варіантів використання;
- провести обґрунтування мови програмування що буде використовуватися для створення системи, обґрунтування СУБД та алгоритму прийняття рішень;
- провести логічне та фізичне моделювання даних з реалізацією бази даних.

## 2 ОПИС СИСТЕМНИХ ТА ФУНКЦІОНАЛЬНИХ ВИМОГ ДО СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

### 2.1 Опис об'єкту проектування та визначення основних системних вимог

Об'єктом проектування в межах даної роботи є проектування системи підтримки прийняття рішень лікарем-терапевтом. В даний час задачі в медицині можуть вирішуватися за допомогою інформаційних технологій, що охоплюють практично всі сторони діяльності медичних організацій, включаючи управління ресурсами, лікувальним процесом і надання медичної допомоги. Великі обсяги даних і обмеженість часових ресурсів при виборі лікування може призводити до зниження ефективності роботи, і як наслідок приводити до лікарських помилок. Таким чином основною функцією систем підтримки прийняття рішень є алгоритмізація процесу прийняття рішення, ґрунтуючись на знаннях експертів, що дозволяє створювати принципово нові підходи до вирішення традиційних завдань в охороні здоров'я, таких як спостереження за пацієнтом, консультація, прийняття рішення щодо подальшого лікування. Так як система підтримки рішень зазвичай є частиною більшої інформаційної системи, процес проектування буде розглянутий з точки зору проектування підсистеми підтримки прийняття рішень як частини інформаційної медичної системи та проектування основних функції інформаційної медичної системи, які необхідні для функціонування системи підтримки прийняття рішень.

До системи, що проектується висуваються наступні вимоги:

- а) зрозумілий інтерфейс для користувачів;
- б) легка інтеграція системи у клінічний робочий процес
- в) можливість роботи з неструктурованими даними
- г) здатність підсистеми бути вбудованою у різні інформаційні медичні системи;
- д) точність прийняття рішень;

е) безпека та захист даних (здатність захистити мережу від несанкціонованого доступу). Інформаційної медичні системи зберігають та обробляють важливі дані захист яких, є однією з найважливіших задач системи. Виділяють три основні типи загроз, які можуть виникати у процесі експлуатації системи: розкриття інформації (несанкціонований доступ до конфіденційних даних), порушення цілісності (зміна даних або їх видалення) та відмова в обслуговуванні. Важливо зазначити що для забезпечення нормального функціонування системи та захисту інформації від несанкціонованого доступу, система повинна відповідати таким вимогам:

- 1) кожен користувач отримує доступ до системи тільки з використанням паролю;
- 2) для індивідуальних користувачів або груп користувачів повинні бути встановленні різні рівні доступу;
- 3) кожен користувач, відповідно до рівня доступу, повинен мати певний набір дозволених можливостей для перегляду або зміни даних;
- 4) система повинна мати резервні сховища даних.

Програмний засіб, що розробляється, має три рівні доступу: адміністратор, лікар, пацієнт.

– адміністратор — людина, яка має доступ до всіх функцій та даних системи для контролювання функціонування системи;

– лікар — людина, що працює у медичному закладі та має ліцензію лікаря;

– пацієнт – людина, авторизувалася у системі за допомогою логіна та паролю.

Для будь-якої людини, яка звертається за медичною допомогою має бути створено аккаунт та надано доступ до нього.

Функції розроблюваної системи поділені на дві категорії, категорія «основні функції» містить функції, які відносяться до рекомендаційної системи, а категорія «додаткові функції» описує функції, які має надавати інформаційна медична система для нормального функціонування рекомендаційної системи. Перелік функцій системи наведено нижче.

Основні функції:

- отримання рекомендації для конкретного пацієнта. Дану функцію може виконувати користувач з роллю «Лікар»;
- оцінка рекомендаційної системи. Після отримання рекомендації користувач з роллю «Лікар» має змогу оцінити отримане рішення та залишити коментар;
- заповнення медичних баз даних. Дану функцію може виконувати користувач з роллю «Адміністратор»;

Додаткові функції:

- реєстрація користувачів. Користувач з роллю «Адміністратор» може реєструвати користувачів з роллю «Лікар» та «Пацієнт», користувач з роллю «Лікар» може зареєструвати користувача «Пацієнт». Користувач з роллю «Пацієнт» не має прав для реєстрування інших користувачів;
- управління аккаунтами користувачів. Функції, що стосуються управління аккаунтами користувачів (видалення, блокування, тощо) доступні для користувачів з роллю «Адміністратор». Функція зміни пароля доступна для користувачів з будь-якою роллю.
- авторизація користувачів за допомогою поштової скриньки та паролю;
- пошук пацієнта у системі та перегляд його електронної медичної карти даних. Електронної медична картки створюється автоматично при реєстрації пацієнта;
- редагування електронної медичної картки;
- запис на прийом. Запис на прийом може бути здійснений користувачем з роллю «Лікар» та «Пацієнт»;

## 2.2 Визначення функціональних вимог до системи

При проектуванні інформаційної системи необхідно провести аналіз цілей цієї системи і виявити вимоги до неї користувачів. Інформація для моделі інформаційної

системи береться на основі організації всебічного обстеження, для якої виконується розробка інформаційної системи. Збір даних починається з вивчення сутностей предметної області, що використовують ці суті, і зв'язків між ними.

Процес бізнес-моделювання може бути реалізований за допомогою різних методологій, що відрізняються насамперед своїм підходом. Умовно методології поділені на дві групи: об'єктні і функціональні [17].

Об'єктні методології розглядають систему, що проектується, як набір взаємодіючих об'єктів. Під об'єктом розуміють реальний предмет/явище, яке має чітко визначену поведінку. Метою застосування даної методології є виділення об'єктів та розподіл між ними відповідальності за виконання певних дій. Характерною особливістю об'єктної декомпозиції є виділенням об'єктів (компонентів), що взаємодіють між собою, виконують певні функції (методи) об'єкта. Таким чином при об'єктному моделюванні системи спочатку виділяються класи об'єктів, а далі в залежності від можливих станів об'єктів (життєвого циклу об'єктів) визначаються методи обробки (функціональні процедури), що забезпечує найкращу реалізацію динамічної поведінки інформаційної системи. Для об'єктно-орієнтованого підходу розроблені графічні методи моделювання предметної області, узагальнені в мові уніфікованого моделювання UML.

Функціональні методології, розглядають систему, що проектується, як набір функцій, що перетворює вхідний потік інформації в вихідний потік. Основною відмінністю такого підходу від об'єктної методології є чітке розмежування функцій (методів обробки даних) від даних. У функціональних моделях головними структурними компонентами є функції (операції, дії, роботи), які на діаграмах зв'язуються між собою потоками даних. Перевагою застосування функціональних моделей є реалізація структурного підходу проектування ІС за принципом «зверху-вниз», коли кожен функціональний блок може мати декомпозицію на безліч підфункцій. Для функціональних моделей характерні процедурна строгість декомпозиції ІС і наочність представлення. Найбільш відомим стандартом функціонального підходу є стандартом IDEF0. Проектування інформаційних систем за стандартом IDEF0 зводиться до декомпозиції основних функцій організації на

окремі бізнес-процеси, роботи або дії. В результаті розробляється ієрархічна модель аналізованої організації, при цьому декомпозицію можна проводити багаторазово, до чіткого і детального опису всіх процесів.

Кожна з розглянутих методологій дозволяє проводити моделювання системи, що проектується, проте з певними обмеженнями. Для того щоб застосувати усі переваги вище описаних методологій та звести до мінімуму недоліки підходів рекомендується застосовувати ідею синтетичної методології. Вона полягає в послідовному застосуванні функціонального і об'єктного підходу з урахуванням можливості реінжинірингу існуючої ситуації. Для функціонального аналізу, при проектуванні компонентів системи підтримки прийняття рішень лікарем-терапевтом, будуть розроблені контекстна діаграма за допомогою стандарту IDEF0 та діаграма варіантів використання для основних бізнес-функцій також для визначення ключових сутностей і визначення зв'язків, які можуть встановлюватися між цими сутностями буде розроблено ER-модель.

На рисунку 2.1 представлена контекстна діаграма функції «Автоматизація процесу прийняття рішення лікарем-терапевтом». Контекстна діаграма дозволяє розглянути систему у вигляді «чорного ящика» та визначити основні вхідні та вихідні дані, механізми управління та функціонування системи.

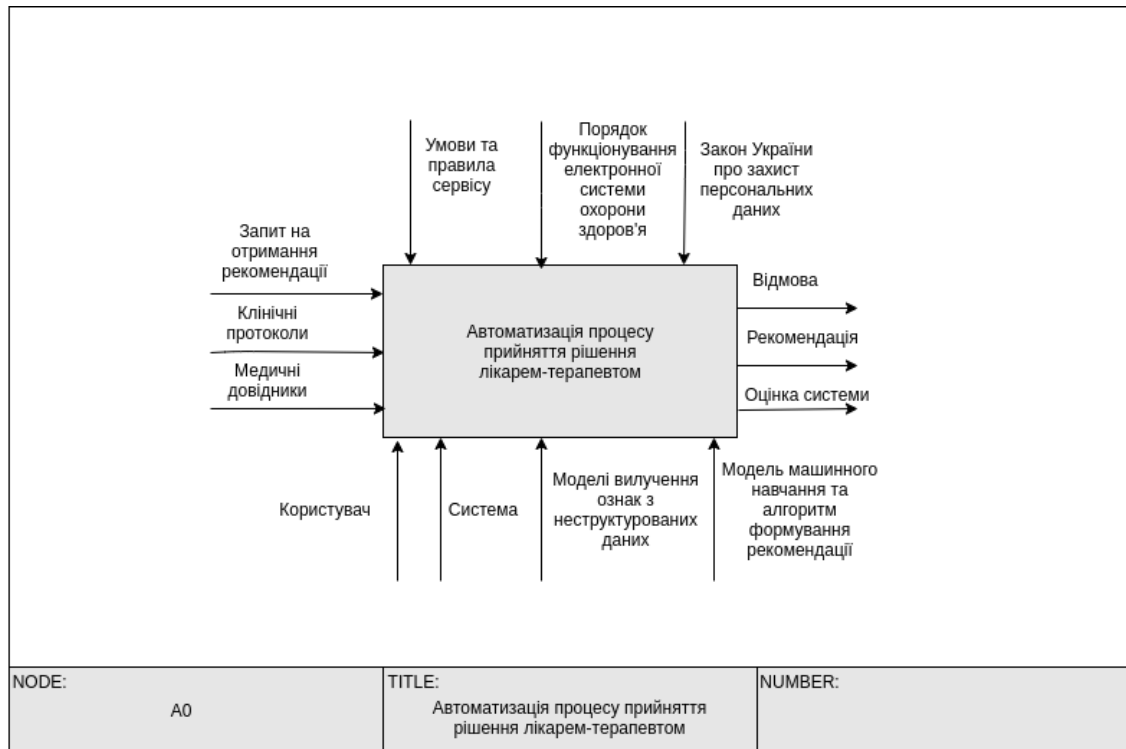


Рисунок 2.1 – Контекстна діаграма функції «Автоматизація процесу прийняття рішення лікарем-терапевтом»

На рисунку 2.2 представлена декомпозиція функції «Автоматизація процесу прийняття рішення лікарем-терапевтом».

Процес автоматизації прийняття рішення лікарем-терапевтом зі сторони користувача(лікаря) відбувається таким чином: лікар відправляє запит на отримання рекомендації до системи, після обробки запиту лікар отримує рекомендацію або відмову у наданні рекомендації.

Процес автоматизації прийняття рішення лікарем-терапевтом зі сторони системи відбувається таким чином: на вхід до системи подається запит на отримання рекомендації, для роботи функції даний запит має включати до себе дані з електронної медичну карти пацієнта для кого робиться запит. Наступним кроком проводиться аналіз даних із запиту. Якщо отриманий вектор ознак не проходить перевірку на достатність та інформативність ознак, результатом роботи системи буде відмова у формування рекомендації. В іншому випадку якщо система успішно вилучає необхідну кількість ознак для отримання рекомендації результатом функції «Аналіз електронної медичної картки» буде вектор ознак, який буде подано на вхід

функції «Застосування моделей машинного навчання». Результатом роботи даної функції буде рішення, яке подається на вхід функції «Формування рекомендацій». Дана функція формує кінцеву рекомендацію, на основі клінічних протоколів та медичних довідників. Після отримання рекомендації користувач має змогу оцінити роботу системи, у подальшому дана оцінка враховується для оновлення моделей машинного навчання, що дозволяє збільшити точність кінцевого рішення, базуючись на знаннях експерта.

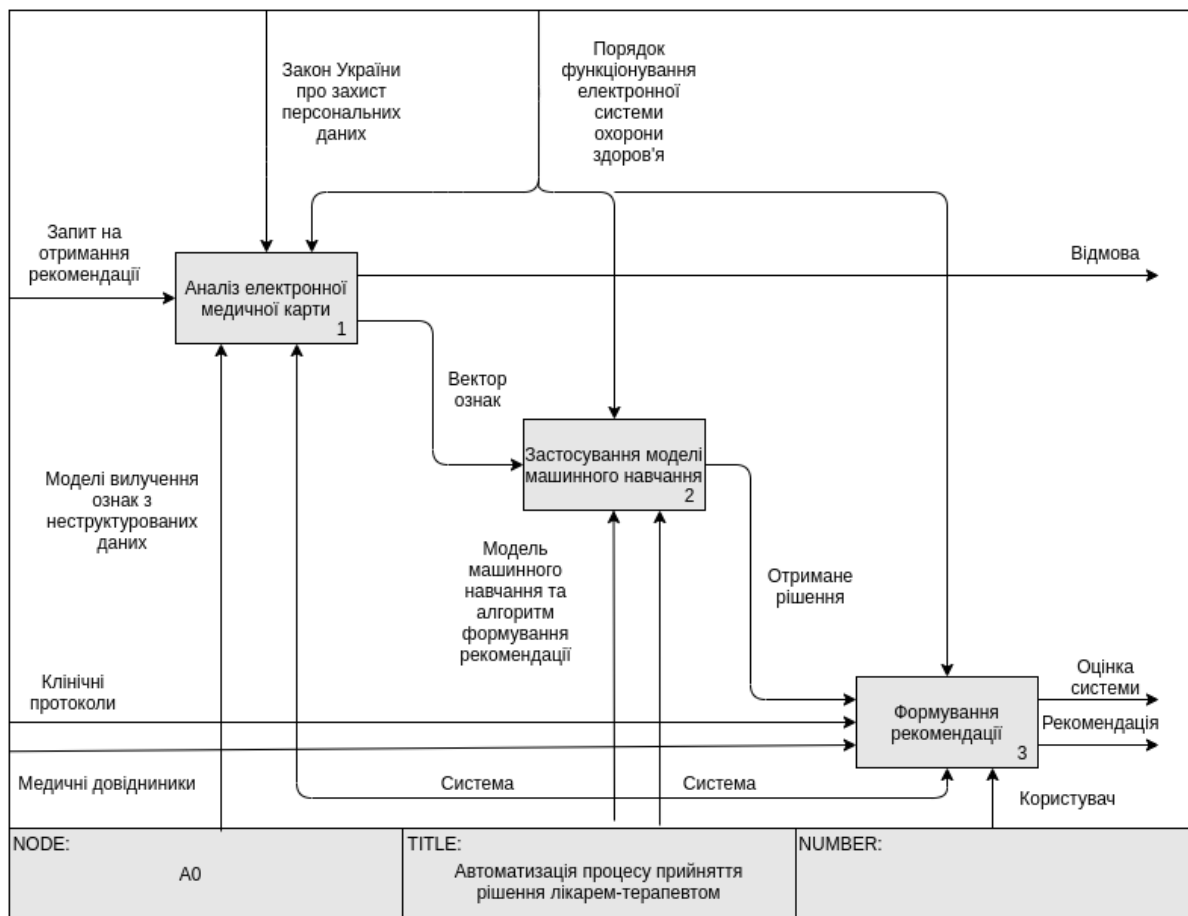


Рисунок 2.2 – Діаграма декомпозиції функції «Автоматизація процесу прийняття рішення лікарем-терапевтом»

На рисунку 2.3 представлена декомпозиція функції «Аналіз електронної медичної карти» на три підфункції: «Знеособлення даних пацієнта», «Вилучення ознак з даних», «Перевірка повноти даних(достатності ознак)». Перша підфункція відповідає за знеособлення даних пацієнта, даний етап обробки даних є важливим,

адже дані пацієнтів є конфіденційною інформацією та можуть зберігатися та використовуватися тільки у межах однієї системи, без права передачі стороннім сервісам. Часто системи підтримки прийняття рішень є сторонніми підсистемами, які використовуються одночасно багатьма іншими системами (Прикладом такої системи є система Webiomed). Такий підхід використовується, адже для навчання моделей прийняття рішень у медицині потрібен великий обсяг актуальних даних, який буде повністю поповнюватися. Системи накопичують неперсоніфіковані медичні дані, з яких формуються датасети для навчання моделей. Таким чином етап знеособлення даних пацієнта дозволяє задовольнити вимогам закону України про захист персональних даних та вимогам порядку функціонування електронних систем охорони здоров'я. Також дані, що дозволяють ідентифікувати людину не є інформативними та важливими для процесу прийняття рішень.

Результатом виконання функції «Знеособлення даних пацієнта» є неперсоніфіковані медичні дані, які подаються на вхід до функції «Вилучення ознак з даних». Під час виконання цієї функції дані обробляються за допомогою моделей вилучення ознак із неструктурованих даних, а результатом роботи функції є вектор ознак, який подається на вхід до функції «Перевірка повноти даних(достатності ознак)». Під час роботи даної функції дані перевіряються на кількість вилучених ознак та їх інформативність та відповідність ознакам, які необхідні для отримання рекомендації. Якщо отриманий вектор ознак не проходить перевірку, результатом роботи системи буде відмова у формуванні рекомендації. В іншому випадку вектор вилучених ознак.

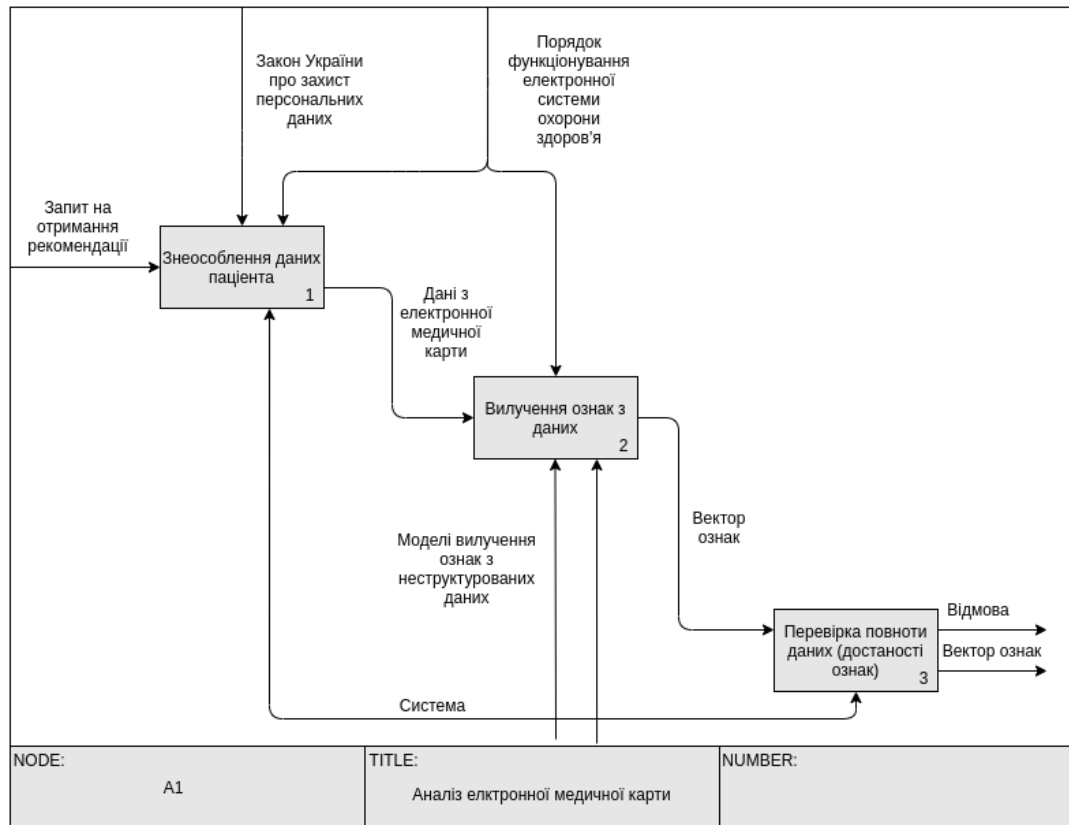


Рисунок 2.3 – Декомпозиція функції «Аналіз електронної медичної картки»

На рисунку 2.4 представлена декомпозиція функції «Формування рекомендації» на дві підфункції: «Формування рекомендації» та «Оцінка системи». На вхід першої підфункції подається отримане рішення, яке є результатом роботи функції «Застосування моделі машинного навчання», медичні довідники та клінічні протоколи. На основі медичних довідників та клінічних протоколів для отриманого рішення формується рекомендація з детальним поясненням. Після отримання рекомендації користувач може оцінити систему. Згодом оцінки користувачів можуть бути проаналізовані для покращення роботи системи.

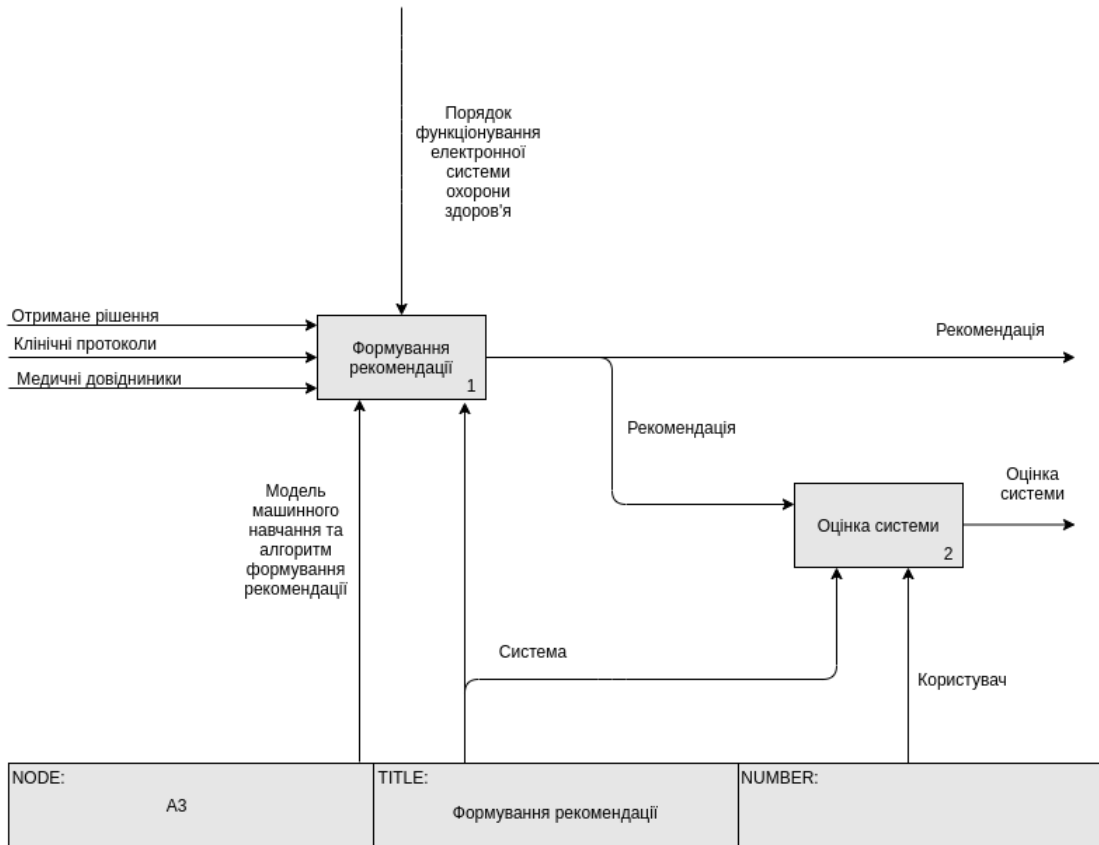


Рисунок 2.4 – Декомпозиція функції «Формування рекомендації»

Для системи, що проектується було розроблено три діаграми варіантів використання. Кожна із діаграм відображає функції та набір дозволених можливостей для перегляду або зміни даних, які може виконувати користувач системи в залежності від рівня доступу.

Користувач з роллю «Адміністратор» може виконувати наступний перелік функцій:

- авторизація;
- заповнення медичних баз даних;
- реєстрація користувачів;
- перегляд даних системи;
- управління аккаунтами користувачів.

На рисунку 2.5 зображена діаграма варіантів використання для ролі «Адміністратор».



Рисунок 2.5 – Діаграма варіантів використання для ролі «Адміністратор»

Користувач з роллю «Лікар» може виконувати наступний перелік функцій:

- а) авторизація;
- б) реєстрація пацієнта;
- в) перегляд даних пацієнта:
  - 1) пошук пацієнта у системі.
- г) запис пацієнта на прийом;
- г) редагування електронної медичної карти:
  - 1) редагування персональних даних;
  - 2) редагування клінічних даних;
  - 3) завантаження лабораторних досліджень.
- д) отримання рекомендації:
  - 1) перегляд отриманої рекомендації;
  - 2) аналіз отриманого рішення:
    - формування кінцевого рішення та збереження результатів;
    - оцінка системи підтримки прийняття рішення.

На рисунку 2.6 зображена діаграма варіантів використання для ролі «Лікар».

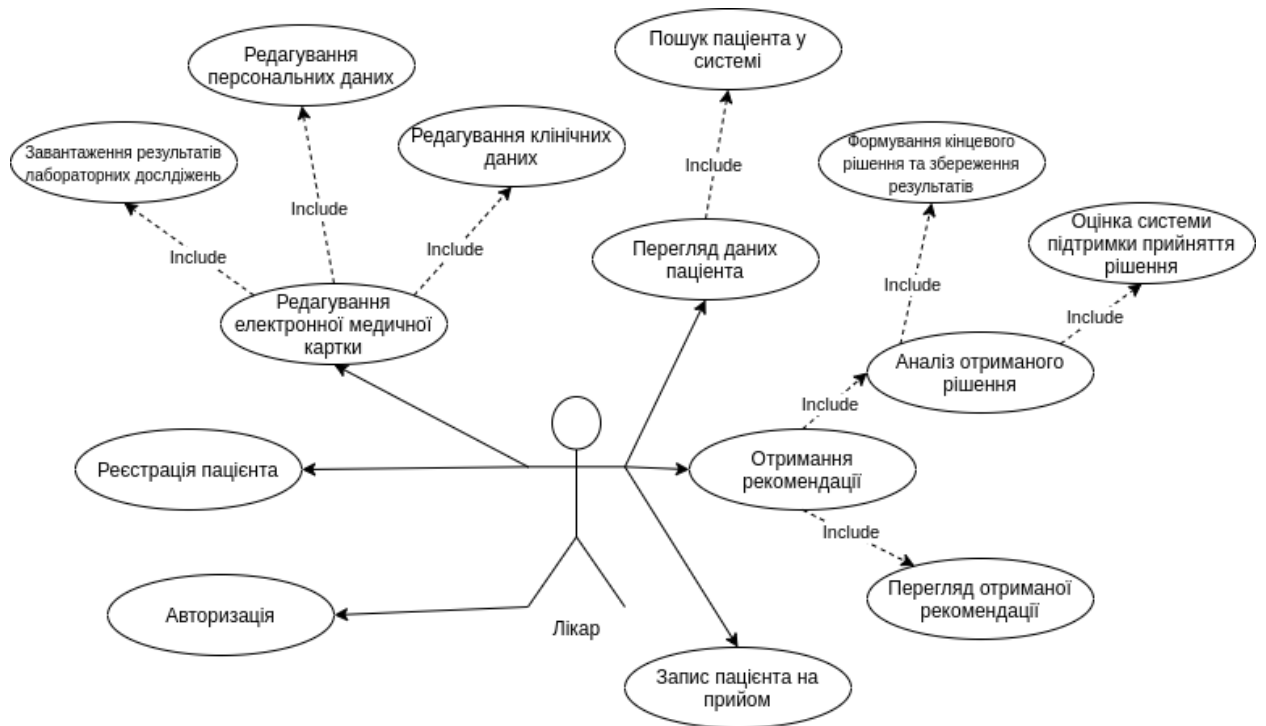


Рисунок 2.6 – Діаграма варіантів використання для ролі «Лікар»

Користувач з роллю «Пацієнт» може виконувати наступний перелік функцій:

- авторизація;
- перегляд особистих даних;
- запис на прийом;

На рисунку 2.7 зображена діаграма варіантів використання для ролі «Пацієнт».

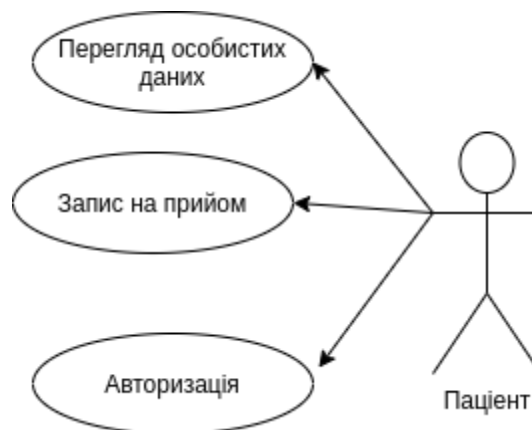


Рисунок 2.7 – Діаграма варіантів використання для ролі «Пацієнт»

### 3 ОПИС АЛГОРИТМУ ПРИЙНЯТТЯ РІШЕННЯ. МЕТОД ОПОРНИХ ВЕКТОРІВ

Метод опорних векторів був розроблений радянськими математиками В. Н. Вапніком і А. Я. Червоненкісом у 1974 році [18].

Основна ідея класифікатора за допомогою метода опорних векторів полягає в відображенні вихідних векторів у простір більш високої розмірності та пошуку розділяючої гіперплощини даних на два класи. Під час навчання класифікатор SVM знаходить гіперплощину у просторі ознак, яка розділяє об'єкти даних на два класи. Задача оптимізації полягає в тому, щоб знайти гіперплощину, яка максимально віддалена (запас) між двома класами. Таким чином під найкращою гіперплощиною, розуміють площину, яка розділяє два класи та має найбільшу відстань від гіперплощин обох класів. Виділяють два типи методу опорних векторів [19]:

– Лінійний метод опорних векторів: Лінійний SVM використовується для лінійно відокремлюваних даних, це означає, що якщо набір даних можна класифікувати на два класи за допомогою однієї прямої лінії, тоді такі дані називаються лінійно відокремлюваними даними, а класифікатор використовується як лінійний класифікатор SVM. На рисунку 3.1 наведено приклад лінійно роздільних даних.

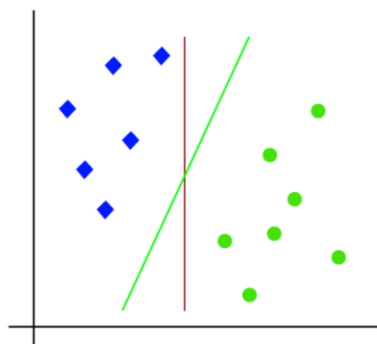


Рисунок 3.1 – Приклад лінійно роздільних даних

– Нелінійний SVM: нелінійний SVM використовується для нелінійно відокремлених даних, що означає, що якщо набір даних неможливо класифікувати за

допомогою прямої лінії, тоді такі дані називаються нелінійними даними, а класифікатор, що використовується, називається нелінійний класифікатор SVM. На рисунку 3.2 наведено приклад лінійно нероздільних даних.

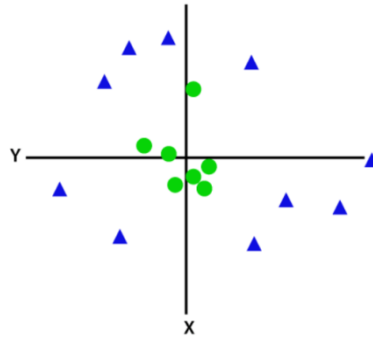


Рисунок 3.2 – Приклад лінійно нероздільних даних

SVM розділяє всі об'єкти даних у просторі об'єктів на два класи. Об'єкти даних повинні мати ознаки  $\{x_1 \dots x_n\}$  та мітку класу  $y_i$ . SVM відображає кожен об'єкт даних як точку в просторі об'єктів, така що об'єкт належить до того чи іншого класу. Зокрема, об'єкт даних (що характеризується своїм вектором ознак) або належить до класу, у цьому випадку мітка класу має значення  $y_i = 1$ , або він не належить до класу (маючи на увазі, що він належить до іншого класу), у такому випадку мітка класу –  $y_i = -1$ . Таким чином дані визначаються за формулою (3.1):

$$Data = \{(x_i, y_i) | x_i \in R^p, y_i \in (-1, 1)\}_{i=1}^n, \quad (3.1)$$

де  $p$  – розмірність вектора ознак;

$n$  – кількість точок даних.

У просторі  $R^n$  рівняння  $\langle \vec{\omega}, \vec{x} \rangle - b = 0$  при заданих  $\vec{\omega}$  і  $b$  визначає гіперплощину – безліч векторів  $\vec{x} = (x_1 \dots x_n)$ , що належать простору меншої розмірності  $R^{n-1}$ . Наприклад, для  $R^1$  гіперплощиною є точка, для  $R^2$ - пряма, для  $R^3$ - площина і т. д. Параметр  $\vec{\omega}$  визначає вектор нормалі до гіперплощини. Таким чином

гіперплощина ділить дані  $R^n$  на дві підплощини, що обмежуються наступними рівняннями (3.2 – 3.3):

$$\langle \vec{\omega}, \vec{x} \rangle - b > 0, \quad (3.2)$$

$$\langle \vec{\omega}, \vec{x} \rangle - b < 0, \quad (3.3)$$

де  $\omega$  – перпендикуляр до розділяючої гіперплощини.

Параметр  $\frac{b}{\|\omega\|}$ , визначає відстань від гіперплощини до початку координат. Якщо параметр  $b$  дорівнює нулю, то гіперплощина проходить через початок координат.

Для двох лінійно розділених класів можливі різні варіанти побудови розділених гіперплощин. Метод опорних векторів обирає ту гіперплощину, яка максимізує відступ між класами: Відступ (англ. Margin) – характеристика, що оцінює, наскільки об'єкт є типовим представником свого класу. Чим менше значення відступу, тим ближче точка знаходиться до межі класів і тем вище стає вірогідність помилки класифікації.

Так як необхідно знайти оптимальне розділення даних, необхідно описати рівняння опорних векторів та гіперплощини, що є паралельні до гіперплощини та найближчі до опорних векторів класів.

Гіперплощини можна задати наступними рівняннями (3.4 – 3.5):

$$\langle \vec{\omega}, \vec{x} \rangle - b = 1, \quad (3.4)$$

$$\langle \vec{\omega}, \vec{x} \rangle - b = -1. \quad (3.5)$$

Для лінійно розділеної вибірки можна обрати гіперплощини таким чином, між ними не лежала жодна точка навчального вибору, а наступним кроком максимізувати відступ між гіперплощинами. Щоб гіперплощина, що розділяє класи була якомога

далі від точок вибірки, ширина смуги повинна бути максимальною. Нехай  $x_-$  і  $x_+$  – дві довільні точки для класів  $-1$  і  $1$  відповідно, що лежать на кордоні смуги. Тоді ширину смуги можна знайти за формулою (3.6):

$$\frac{\langle \omega x_+ \rangle - \langle \omega x_- \rangle}{\|\omega\|} = \frac{(b+1) - (b-1)}{\|\omega\|} = \frac{2}{\|\omega\|}. \quad (3.6)$$

Так як ширина між гіперплощинами визначається як  $\frac{2}{\|\omega\|}$  для максимізації відступу необхідно мінімізувати  $\omega$ . Для того щоб жодна точка не лежала між гіперплощинами, необхідно виконання наступної умови (3.7):

$$\begin{aligned} \langle \vec{\omega}, \vec{x} \rangle - b &\geq 1, y_i = 1, \\ \langle \vec{\omega}, \vec{x} \rangle - b &\leq -1, y_i = -1. \end{aligned} \quad (3.7)$$

Таким чином проблема побудови оптимальної роздільної гіперплощини зводиться до мінімізації  $\|\omega\|$ , при умові 1. Задача квадратичної оптимізації має вигляд (3.8):

$$\begin{cases} \|\omega\|^2 \rightarrow \min \\ y_i(\omega * x_i - b) \geq 1, 1 \leq i \leq n \end{cases} \quad (3.8)$$

За теоремою Куна-Таккера ця задача еквівалентна задачі пошуку сідлової точки функції Лагранжа. Загальний вигляд функції Лагранжа (3.9):

$$L(x, \lambda) = f(x) + \sum_{i=1}^n \lambda_i \varphi_i(x), \quad (3.9)$$

де  $\lambda = (\lambda_1, \dots, \lambda_n)$  – множиники Лагранжа;

$n$  – кількість обмежень;

$\varphi_i$  – обмеження.

Функція Лагранжа для задачі квадратичної оптимізації побудови роздільної гіперплощини має вигляд (3.10):

$$\begin{cases} L(\omega, b; \lambda) = \frac{1}{2} \|\omega\|^2 - \sum_{i=1}^n \lambda_i (y_i (\omega * x_i - b) - 1) \rightarrow \min_{\omega, b} \max_{\lambda} \\ \lambda_i \geq 0, \quad 1 \leq i \leq n \end{cases} \quad (3.10)$$

Після розв'язання отриманої задачі можна визначити формули (3.11 – 3.12) для знаходження значень  $\omega$  та  $b$  формула:

$$\omega = \sum_{i=1}^n \lambda_i y_i x_i, \quad (3.11)$$

$$b = \omega * x_i - y_i, \lambda_i > 0. \quad (3.12)$$

З вище описаних формул можна визначити алгоритм класифікації для лінійно роздільних даних за формулою (3.13):

$$\alpha(x) = \text{sign}\left(\sum_{i=1}^n \lambda_i y_i x_i * x - b\right), \quad (3.13)$$

де  $\sum_{i=1}^n \lambda_i y_i x_i * x - b$  – лінійна комбінація векторів для задання  $n$ -вимірної гіперплощини.

При цьому додавання відбувається для опорних векторів, для яких  $\lambda_i \neq 0$ .

На рисунку 3.3 наведено приклад виконання алгоритму методу опорних векторів.

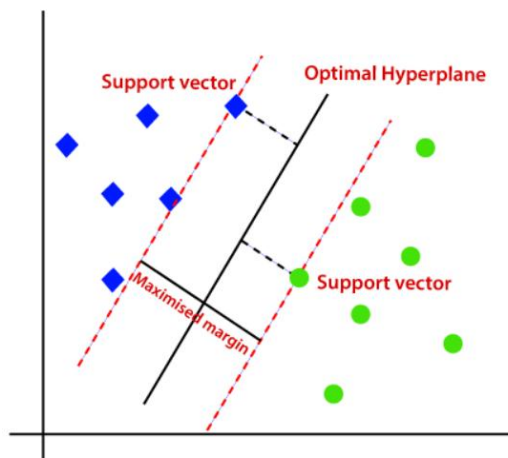


Рисунок 3.3 – Результат виконання алгоритму методу опорних векторів.

Розглянутий вище алгоритм описує класифікацію з жорстким відступом (hard margin classification). Класифікація називається класифікацією з жорстким відступом, якщо всі зразки знаходяться поза смугою. Така класифікація має ряд недоліків:

- можливість проведення класифікації тільки для лінійно роздільних даних;
- чутливість до викидів.

Для ситуацій коли застосування класифікації з жорстким відступом є неможливим використовують класифікацією з м'яким відступом (soft margin classification). Ідея методу полягає в тому, щоб відшукати баланс між утриманням смуги якомога ширшою і обмеженням кількості порушень відступу (тобто появи прикладів, які відображаються у середині смуги або навіть по іншу сторону гіперплощини). Для застосування класифікацією з м'яким відступом до умови, якій повинна відповідати оптимальна гіперплощина вводять додатковий параметр  $\xi$ . Тоді умова оптимальна гіперплощина набуває вигляду за формулою (3.14):

$$y_i(\omega * x_i - b) \geq 1 - \xi_i, 1 \leq i \leq n. \quad (3.14)$$

А задача квадратичної оптимізації набуває наступний вигляд за формулою (3.15):

$$\begin{cases} \|\omega\|^2 + C \sum_{i=1}^n \xi_i \rightarrow \min \\ y_i(\omega * x_i - b) \geq 1 - \xi_i, \xi_i > 0 \quad 1 \leq i \leq n \end{cases}, \quad (3.15)$$

де  $C$  – параметр регуляризації, який контролює компроміс між збільшенням відступу та мінімізацією помилки навчання.

На рисунку 3.4 наведено приклад виконання алгоритму методу опорних векторів для класифікації з м'яким відступом.

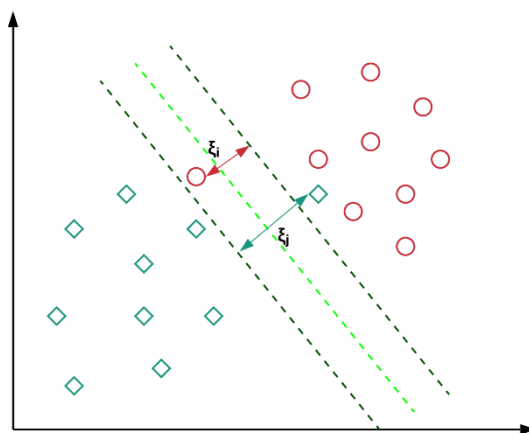


Рисунок 3.4 – Результат виконання алгоритму методу опорних векторів для класифікації з м'яким відступом.

У реальному житті дані як правило є лінійно нероздільними, для вирішення задачі класифікації для лінійно нероздільних даних застосовують прийом Kernel Trick [20]. Ідея методу полягає у відображенні нелінійного набору даних у більший розмірний простір, де можна знайти гіперплощину, яка розділить вибірку. Як видно із формули 3.13 для обчислення вхідного значення  $x$  необхідно обчислити скалярні добутки з всіма опорними векторами, які використовуються у вирішальній функції. Таким чином скалярний добуток можна замінити іншою функцією виду

$K(x_1, x_2)$ , яка буде скалярним добутком в іншому просторі з більшою розмірністю. Ядром називається  $K(x_1, x_2)$ , яка з урахуванням двох векторів у вхідному просторі повертає точковий добуток їх відображення у просторі об'єктів. Вигляд ядра визначається за формулою (3.16):

$$K(x_1, x_2) = \langle \varphi(x_1) * \varphi(x_2) \rangle, \quad (3.16)$$

де  $\varphi(x)$  – функція перетворення.

За теоремою Мерсера функція  $K(x_1, x_2)$  є ядром тоді і тільки тоді, коли вона симетрична за рівнянням (3.17) и невід'ємно визначена за рівнянням (3.18)

$$K(x_1, x_2) = K(x_2, x_1), \quad (3.17)$$

$$\int \int K(x_1, x_2) g(x_1) g(x_2) dx_1 dx_2 \geq 0, \quad (3.18)$$

де  $g$  – будь-яка функція  $X \rightarrow R$ .

Якщо взяти рівняння 3.13, яке є функцією прийняття рішення для оптимального класифікатора гіперплощин у подвійній формі, і застосувати відображення  $\varphi$  до кожного використовуваного ним вектора, ми отримаємо формулу (3.19):

$$\alpha(x) = \text{sign}\left(\sum_{i=1}^n \lambda_i y_i K(x_i, x) - b\right) \quad (3.19)$$

На рисунку 3.5 представлено приклад відображення даних у простір іншої розмірності:

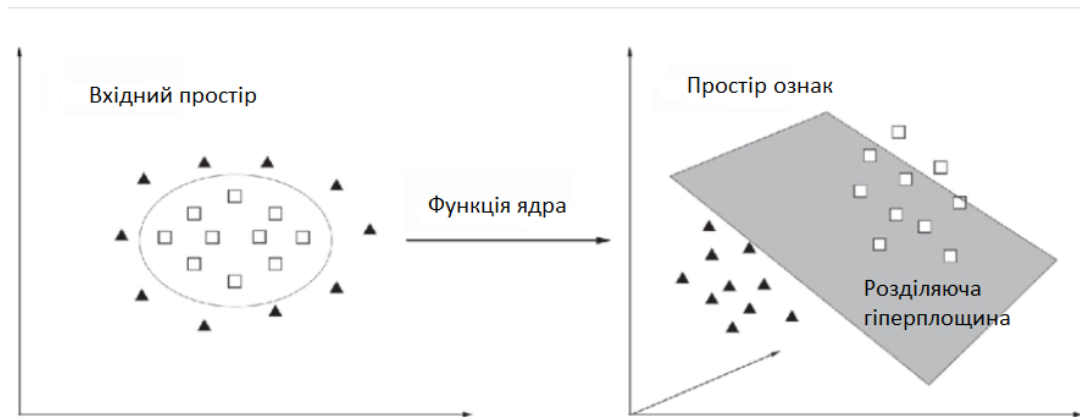


Рисунок 3.5 – Відображення даних у простір іншої розмірності

Виділять наступний перелік найпопулярніших ядер для методу опорних векторів: поліноміальна функція, радіально-базисна функція, радіально-базисна функція Гауса, сигмоїд.

Поліноміальна функція визначається за формулою (3.20):

$$K(x_1, x_2) = (x_1 * x_2 + 1)^d, \quad (3.20)$$

де  $d$  – ступінь поліноміального ядра. Даний ступінь контролює гнучкість результуючого класифікатора.

Радіально базисна функція визначається за формулою (3.21):

$$K(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|^2), \quad (3.21)$$

Радіально базисна функція Гауса визначається за формулою (3.22):

$$K(x_1, x_2) = \exp\left(\frac{-\|x_1 - x_2\|^2}{2\sigma^2}\right), \quad (3.22)$$

де  $\sigma$  – параметр, який контролює ширину Гауса. Даний параметр відіграє подібну роль, як ступінь поліноміального ядра.

Сигмоїд визначається за формулою (3.23):

$$K(x_1, x_2) = \tanh(k * x_1 * x_2 + c) \quad (3.23)$$

Підходяще ядро для методу опорних векторів визначається шляхом послідовного використання різних ядер, починаючи з лінійного ядра.

## 4 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

### 4.1 Вибір архітектури системи

Архітектура інформаційної системи – це концепція, яка визначає модель, структуру, функції і взаємозв'язок компонентів інформаційної системи. Компоненти інформаційної системи по виконуваних функцій можна розділити на три шари: рівень представлення, рівень бізнес-логіки та рівень доступу до даних.

Рівень представлення – все, що пов'язано з взаємодією користувача та системи.

Рівень бізнес-логіки – правила, алгоритми та реакції програми на дії користувача або на внутрішні події, правила обробки даних.

Рівень доступу до даних – зберігання, редагування та видалення даних.

З точки зору програмно-апаратної реалізації можна виділити ряд типових архітектур ІС: файл-серверна архітектура, клієнт-серверна архітектура, трирівнева клієнт-серверна архітектура [21].

Додатки з файл-серверною архітектурою схожі за своєю структурою з локальними програмами, що використовують мережевий ресурс для зберігання програми і даних. До функції сервера відносяться: зберігання даних і коду програми. До функції клієнта – обробка даних (вона відбувається виключно на стороні клієнта). До головних переваг такої системи можна віднести: система розрахована на одночасну роботу з даними (кількість клієнтів обмежена десятками), зручність централізованого управління доступом, низька вартість розробки. Незважаючи на описані вище переваги системи з такою архітектурою мають ряд суттєвих недоліків: низька продуктивність, низька надійність, слабкі можливості масштабування. Всі ці недоліки виникають через те, що дані зберігаються в одному місці, а обробляються в іншому, що породжує необхідність передачі даних за допомогою мережі. Це призводить до високих навантажень на мережу і, внаслідок цього, різкого зниження продуктивності додатки при збільшенні числа одночасно працюючих клієнтів.

Ключовою відмінністю архітектури клієнт-сервер від архітектури файл-сервер є абстрагування від внутрішнього представлення даних (фізичної схеми даних). У такій архітектурі клієнтські програми маніпулюють даними на рівні логічної схеми. Клієнтська програма працює з даними за допомогою запитів до серверного програмного забезпечення, а базові функції програми розділені між клієнтом і сервером. Основними перевагами використання такої архітектури є підтримка багатокористувацької роботи та гарантія цілісності даних. Серед основних недоліків можна виділити: необхідність використовувати потужні ПК на клієнтських місцях, високу складність розробки системи через необхідність виконувати бізнеслогіку і забезпечувати користувальницький інтерфейс в одній програмі, високі вимоги до пропускну здатності комунікаційних каналів з сервером

Трирівнева клієнт-серверна архітектура. Основною її відмінністю є фізичне розділення програм, що відповідають за зберігання даних (СУБД) від програм ці дані обробляють (сервер системи). Таке розділення програмних компонентів дозволяє оптимізувати навантаження як на мережеве, так і на обчислювальне обладнання комплексу. До основних переваг такої архітектури відносять: зниження навантаження на сервер даних, що дозволяє в свою чергу підвищити швидкість роботи системи в цілому, дешевий трафік між сервером додатків і СУБД, можливість роботи сервера на одному або декількох комп'ютерах, що дозволяє використовувати обчислювальні потужності організації настільки ефективно і безпечно як цього забажає адміністратор ІС

Також важливо зазначити, що при створенні ІС доцільно використовувати технології «хмарних обчислень» (cloud computing), які швидко розвиваються в даний час і широко використовуються для реалізації складних територіально розподілених інформаційних систем [22].

Хмарні обчислення – модель надання сервісу, при якій користувач має можливість отримати за вимогою, зручний доступ до пулу розділених ресурсів (наприклад мереж, серверів, пам'яті, додатків і т.д.). Головною перевагою використання технологій хмарних обчислень є суттєве підвищення ефективності

автоматизованих процесів і скорочення витрат на створення, підтримку і розвиток інформаційних систем.

Проаналізувавши основні переваги та недоліки описаних вище архітектур, було обрано трирівневу клієнт-серверна архітектура, яка задовольняє всім вимогам, що висуваються до системи. Схему трирівневої клієнт-серверної архітектури наведено на рисунку 4.1.



Рисунок 4.1 — Схема трирівневої клієнт-серверної архітектури

## 4.2 Обґрунтування вибору СУБД

На даний час ринок СУБД досить великий, є багато як платних, так і безкоштовних продуктів. Кожна з цих систем реалізує основні механізми управління базами даних та пропонує свої унікальні функції для роботи з даними. При проведенні порівняльного аналізу різних СУБД, слід враховувати зручність системи у роботі, її масштабованість та сумісність з іншими продуктами, які вже використовуються. Крім того, під час вибору слід взяти до уваги вартість системи та якість підтримки, що надається розробником. Враховуючи специфіку розроблюваної системи та дані з якими система бути працювати було вирішено використовувати

реляційні бази даних, то ми нижче буде наведено порівняльний аналіз СУБД для реляційних баз даних.

MySQL – реляційна СУБД, що має відкритий вихідний код та дозволяє підтримувати табличні БД з простою структурою і складними умовами запитів [23]. Вона відрізняється гнучкістю і високою швидкістю обробки інформації, простотою інтерфейсу і здатністю синхронізації з іншими БД.

Microsoft SQL Server. Ця розробка Microsoft, що може використовуватися в ОС Windows і Linux. Її характеризують простий інтерфейс, надійність збереження даних і сумісність з різними програмними продуктами Windows [24]. В інтелектуальному аналізі даних ці СУБД використовуються головним чином для обробки даних з Microsoft Excel.

Oracle – об'єктно-реляційна СУБД, якій притаманні висока надійність, практичність і швидкодія [25]. Дана СУБД дозволяє автоматизувати задачі адміністрування, забезпечує безпеку і відповідність нормативно-правовим актам захисту інформації, містить функції управління і самодіагностики. До характеристик системи відноситься управління великими обсягами даних за допомогою використання компресії і розподілених таблиць. Проте ця СУБД має суттєвий недолік – вартість.

Для розробки предметної області обрано систему управління базами даних MySQL Workbench. MySQL Workbench – це інструмент для візуального проектування баз даних, в складі якої є інструменти проектування, моделювання, створення та експлуатація БД. Дана СУБД була використана для розробки предметної області, тому що вона дозволяє наочно представити модель бази даних, надає наочний механізм установки зв'язків та дозволяє редагувати таблиці у візуальному режимі. Основними перевагами даної СУБД є: простота використання та легкість інтеграції з програмним забезпеченням.

Логічна модель бази даних представлена у вигляді діаграми сутність-зв'язок на рисунку 4.2. ER-діаграми використовуються для моделювання схеми структури бази даних і являють собою стандартний спосіб визначення даних і відносин між ними. ER-діаграма містить інформацію про сутності системи і способи їх взаємодії,

включає ідентифікацію об'єктів, важливих для предметної області (сутностей), властивостей цих об'єктів (атрибутів) і їх відносин з іншими об'єктами (зв'язків). Сутність зображується у вигляді прямокутника, вгорі якого розташовується ім'я сутності. В прямокутнику можуть бути перераховані атрибути сутності; атрибути ER-діаграм, відзначені малюнком ключа, є ключовими, інші атрибути ключовими не є. Відношення зображується лінією між двома сутностями типом зв'язку «1 до багатьох» та «1 до 1-го».

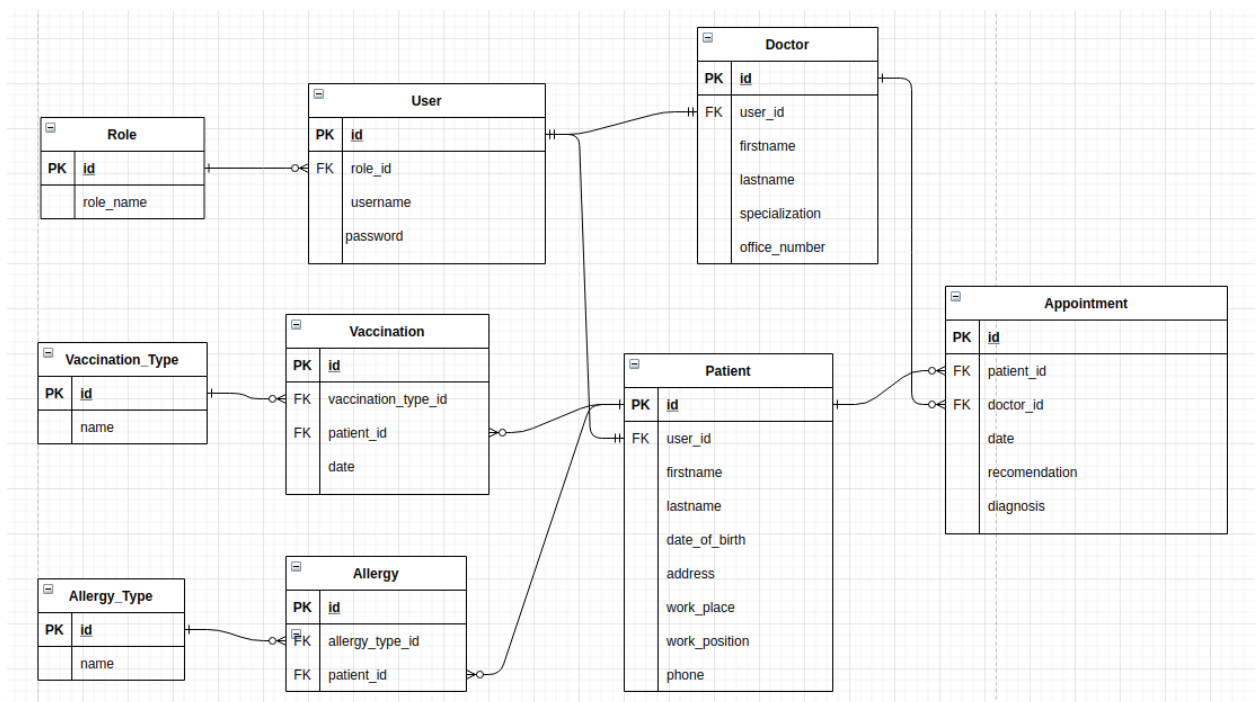


Рисунок 4.2 – Логічна модель бази даних

Для перелік сутностей бази даних з детальним описом наведено нижче:

- **User** – це сутність, що зберігає дані користувачі системи. Користувачі мають різні рівні доступу у системі(ролі). Якщо користувач має роль відмінну від «Адміністратор», він має бути зв'язаний з обліковим записом пацієнта або лікаря;
- **Role** – сутність, яка зберігає перелік можливих рівнів у систем;
- **Patient** – це сутність, що зберігає усі дані користувача з роллю «Пацієнт». Всі записи про щеплення, прийому у лікаря та алергії повинні мати посилання на запис пацієнта у базі даних;

- Doctor – це сутність, що зберігає усі дані користувача з роллю «Лікар»;
- Vaccination – це сутність, яка зберігає щеплення пацієнтів;
- Vaccination\_Type – це сутність, яка зберігає перелік можливих щеплень згідно медичних довідників;
- Allergy – це сутність, яка зберігає алергії пацієнтів;
- Allergy\_Type – це сутність, яка зберігає перелік можливих алергій згідно медичних довідників;
- Appointment – це сутність, що зберігає дані прийому пацієнта лікарем. Після закінчення прийому лікар зобов'язаний занести результати прийому до бази даних

Створена база даних представляє собою реляційну базу даних – відповідну методам і вимогам до інформаційної бази сукупність взаємопов'язаних таблиць, кожна з яких містить інформацію про об'єкти певного типу [26]. Реляційні бази даних дозволяють зберігати інформацію в декількох двомірних таблицях, пов'язаних між собою за допомогою спільно використовуваних полів даних, які називаються ключами. Записи складаються з полів, що зберігають атрибути об'єкта. Кожен стовпець, описує одну характеристику об'єкта і має певний тип даних. Всі записи мають одні і ті ж атрибути, тільки в них відображаються різні інформаційні властивості об'єкта.

Для усунення дублювання і несуперечності інформації, що зберігається, реляційна модель даної ІС була приведена до 3 нормальній формі.

Фізичну модель бази даних для медичної системи представлено на рисунку 4.3

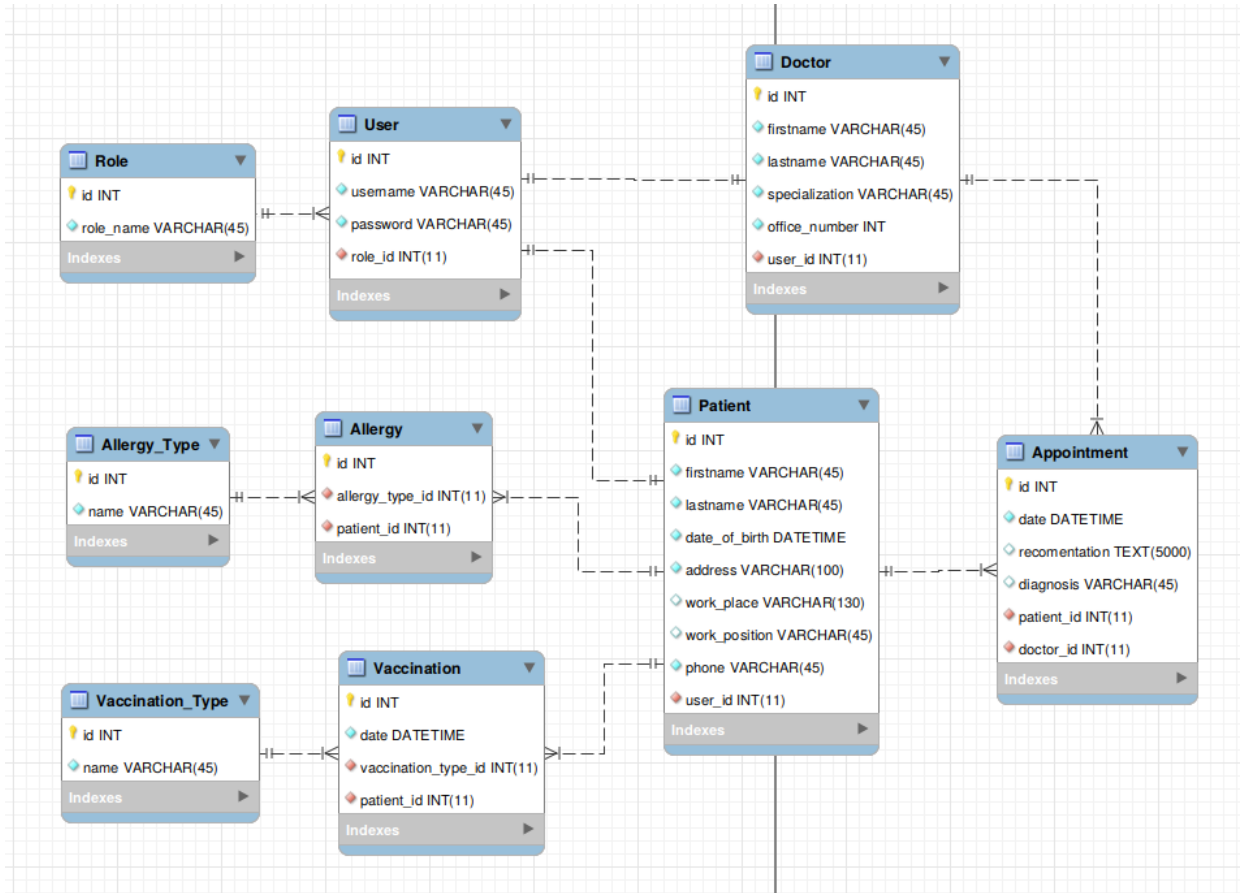


Рисунок 4.3 – Фізична модель бази даних

Таблиця 4.1 – Опис атрибутів бази даних

Тип сутності	Атрибут	Опис	Тип даних	Обмеження	Допустимість на null
User	id	Унікальний ідентифікатор користувача	Int	PK	Ні
	username	Ім'я користувача	Varchar(45)		Ні
	password	Пароль користувача	Varchar(45)		Ні
	role_id	Унікальний ідентифікатор ролі	Int(11)	FK	Ні

Продовження таблиці 4.1

Тип сутності	Атрибут	Опис	Тип даних	Обмеження	Допустимість null
Patient	id	Унікальний ідентифікатор пацієнта	Int	PK	Ні
	firstname	Ім'я пацієнта	Varchar(45)		Ні
	lastname	Прізвище пацієнта	Varchar(45)		Ні
	date_of_birth	Дата народження	Datetime		Ні
	address	Адреса проживання	Varchar(100)		Ні
	work_place	Місце роботи	Varchar(130)		Так
	work_position	Посада	Varchar(45)		Так
	phone	Телефонний номер	Varchar(45)		Ні
	user_id	Унікальний ідентифікатор користувача	Int(11)	FK	Ні
Doctor	id	Унікальний ідентифікатор лікаря	Int	PK	Ні
	firstname	Ім'я лікаря	Varchar(45)		Ні
	lastname	Прізвище лікаря	Varchar(45)		Ні
	specialization	Спеціалізація лікаря	Varchar(45)		Ні
	office_number	Номер кабінету	Int		Ні
		user_id	Унікальний ідентифікатор користувача	Int(11)	FK

Продовження таблиці 4.1

Тип сутності	Атрибут	Опис	Тип даних	Обмеження	Допустимість null
Role	id	Унікальний ідентифікатор ролі	Int	PK	Ні
	role_name	Роль користувача	Varchar(45)		Ні
Allergy_Type	id	Унікальний ідентифікатор типу алергії	Int	PK	Ні
	name	Назва алергії	Varchar(45)		Ні
Allergy	id	Унікальний ідентифікатор алергії	Int	PK	Ні
	allergy_type_id	Унікальний ідентифікатор типу алергії	Int (11)	FK	Ні
	patient_id	Унікальний ідентифікатор типу пацієнта	Int (11)	FK	Ні
Vaccination_Type	id	Унікальний ідентифікатор типу щеплення	Int	PK	Ні
	name	Назва щеплення	Varchar(45)		Ні

Продовження таблиці 4.1

Тип сутності	Атрибут	Опис	Тип даних	Обмеження	Допустимість null
Vaccination	id	Унікальний ідентифікатор щеплення	Int	PK	Hi
	vaccination_type_id	Унікальний ідентифікатор типу щеплення	Int (11)	FK	Hi
	patient_id	Унікальний ідентифікатор типу пацієнта	Int (11)	FK	Hi
	date	Дата проведення щеплення	Datetime		Hi
Appointment	id	Унікальний ідентифікатор запису	Int	PK	Hi
	date	Дата проведення прийому	Datetime		Hi
	recommendation	Рекомендація КСППР	Text(5000)		Так
	diagnosis	Діагноз	Varchar(45)		Так
	doctor_id	Унікальний ідентифікатор типу лікаря	Int (11)	FK	Hi
	patient_id	Унікальний ідентифікатор типу пацієнта	Int (11)	FK	Hi

### 4.3 Вибір мови програмування

Вибір мови програмування є важливим етапом розробки будь-якого продукту. Так як у роботі розглядається інформаційна медична система та підсистема підтримки прийняття рішень важливо визначити мову програмування для основного функціоналу системи та алгоритму прийняття рішень. До основних вимог, що висуваються до основної мови програмування можна віднести: кросплатформеність, легка інтеграція з базами даних, можливість написання графічного інтерфейсу.

На сьогоднішній день можна виділити такі найбільш широко використовувані мови програмування: JavaScript, C#, Java.

– JavaScript. Від самого початку дана мова розроблялась для написання клієнтської частини для веб-додатків, проте з появою Node.js вона широко використовується для написання серверної частини додатків. Також донедавна вважалося, що ця мова не є зручною для написання серверної частини, адже це не типізована мова програмування, на даний момент ця проблема вирішується за допомогою використання Typescript. До основних переваг мови можна віднести кросплатформеність, легку інтеграція з будь-якою базою даних, можливість написання графічної частини, шикороке різноманіття бібліотек та фреймворків для front-end частини (React, AngularJs, Vue т.д)

– Java. Java є широко використовуваною об'єктно-орієнтованою мовою для великих, складних ентерпрайз проектів. До основних переваг можна віднести: зручність розробки, кросплатформеність, легка взаємодія з будь-якою СУБД, багатопоточність. Для побудови графічного інтерфейсу можна використовувати Swing або JavaFX, проте тільки для десктопних додатків.

– C# – це об'єктно-орієнтована мова програмування, яка відноситься до сімейства мов C. Використання даної мови програмування дозволяє створювати як десктопні додатки так і веб-додатки, до особливостей можна віднести LINQ запити, створюють дуже зручну мовну конструкцію для строго типізованих запитів. Проте

використання даної мови є доволі складним, через велику кількість модулів та можливостей.

Проаналізувавши основні переваги та недоліки описаних вище мов програмування, у якості основної мови програмування було обрано мову Javascript. Для написання серверної частини буде використано Node.js з Typescript, для клієнтської частини - React. Дана бібліотека дозволяє будувати складні графічні інтерфейси з можливістю розширення та перевикористання компонентів інтерфейсу.

Для програмування алгоритмів машинного навчання можна виділити такий список мов програмування: R, Matlab, Python.

– R є і мовою програмування і програмним забезпеченням. В основному R використовується для статистичних обчислень. Він має набір алгоритмів, які поглиблено застосовуються в області машинного навчання, зокрема в аналізі часових рядів, класифікації, кластеризації та лінійному. До переваг можна віднести: велика кількість графічних інтерфейсів та модулів полегшення роботи, широкий набір операторів для обробки масивів, матриць, і інших складних конструкцій, низький поріг входу.

– Matlab Мова MATLAB є високорівневою інтерпретованою мовою програмування, що включає в себе засновані на матрицях структури даних, широкий спектр функцій, інтегроване середовище розробки, об'єктно-орієнтовані можливості. Особливостями мови є орієнтація на роботу з матрицями, великий набір готових реалізацій алгоритмів, багаті можливості візуалізації даних і взаємодії з іншими мовами.

– Python – це широко використовувана мова програмування загального призначення. Python є високорівневою мовою програмування з динамічною семантикою. Вона має великий набір спеціально розроблених модулів та дуже проста у вивченні. Однією із основних причин, широкого використання цієї мови у машинного навчання є велика кількість бібліотек і фреймворків, які спрощують процес написання коду і скорочують час на розробку. Найпопулярнішими з них є numpy, pandas, sciPy, sciKit-Learn. Ці бібліотеки працюють в таких фреймворків, як TensorFlow, CNTK і Apache Spark

Проаналізувавши основні переваги та недоліки описаних вище мов програмування, для створення алгоритму прийняття рішення було обрано мову програмування Python.

#### 4.4 Опис обраних технологій розробки та програмних засобів

Ґрунтуючись на виборі технологій розробки, які були визначені у попередніх підрозділах, можна визначити повний перелік програмних засобів необхідних для розробки продукту та версій технологій, що будуть використовуватися.

Для розробки основного функціоналу та серверної частини буде використовуватися мова програмування Node.js 14.15 з Typescript 3.9.3 та середовище розробки IDE WebStorm 2020.3. IDE WebStorm 2020.3 є комерційним засобом, ліцензія для доступу надається навчальним закладом. Для написання клієнтської частини буде використано мову Javascript з бібліотекою React 16.13.1.

Для бази даних обрано MySQL 5.7.32 та система управління базами даних MySQL Workbench 8.0.17.

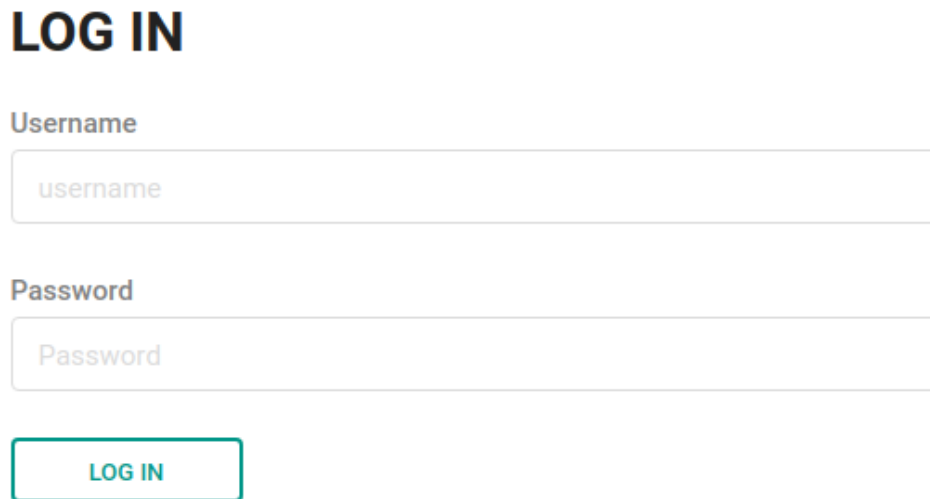
Для розробки алгоритму прийняття рішення обрано мову програмування Python 3.8 та середовище розробки IDE PyCharm 2020.3. IDE PyCharm 2020.3 є комерційним засобом, ліцензія для доступу надається навчальним закладом.

#### 4.5 Огляд клієнтського інтерфейсу розроблюваної системи

Розроблювана система представлена у вигляді веб сайту. Для огляду клієнтського інтерфейсу буде розглянута основна функція системи - проведення прийому пацієнта та отримання рекомендації для нього. Наразі система має одну

модель для підтримки прийняття рішень, яка дозволяє отримати заключення про наявність цукрового діабету у пацієнта.

Для початку роботи користувачу необхідно авторизуватися у системі. Використовуючи юзернейм та пароль отримані від адміністратора користувач може авторизуватися у системі на сторінці «Log in», що представлена на рисунку 4.4:



**LOG IN**

Username

Password

**LOG IN**

Рисунок 4.4 – Сторінка авторизації

Після успішної авторизації користувач отримує доступ до головної сторінки в залежності від ролі. Для користувача з роллю «Лікар» відображається сторінка зі списком всіх зареєстрованих пацієнтів, що представлена на рисунку 4.5. На цій сторінці користувач може знайти пацієнта, зареєструвати нового та переглянути історію змін даних пацієнтів.

healthcare

Olga Mironenko

Patients

PATIENTS

NEW

Search...

Full Name	Phone	Gender	
<a href="#">Ivan Maximov</a>	0935462853	Male	<a href="#">+</a> <a href="#">-</a>
<a href="#">Karyna Khorshun...</a>	0954236527	Female	<a href="#">+</a> <a href="#">-</a>
<a href="#">Roman Ponomar...</a>	0950361525	Female	<a href="#">+</a> <a href="#">-</a>

Previous Page 1 of 1 10 rows Next

[Roman Ponomarenko](#)  
Updated today by Olga Mironenko

[Roman Ponomarenko](#)  
Updated today by Olga Mironenko

[Roman Ponomarenko](#)  
Updated today by Olga Mironenko

[Roman Ponomarenko](#)  
Updated today by Olga Mironenko

[Karyna Khorshunova](#)  
Updated today by Olga Mironenko

[Karyna Khorshunova](#)  
Updated today by Olga Mironenko

[Karyna Khorshunova](#)  
Updated today by Olga Mironenko

[Roman Ponomarenko](#)  
Updated today by Olga Mironenko

[Karyna Khorshunova](#)  
Updated today by Olga Mironenko

[Ivan Maximov](#)  
Updated today by Olga Mironenko

VIEW HISTORY

Рисунок 4.5 – Головна сторінка для користувача з роллю «Лікар»

Для користувача з роллю «Пацієнт» після авторизації відображається сторінка особистого профілю, представлена на рисунку 4.6. Також ця сторінка є доступною для користувач з роллю «Лікар», якщо лікар хоче переглянути дані пацієнта. На цій сторінці представлені головні персональні дані та перелік усіх візитів пацієнта, історія змін даних, також є можливість редагування електронної медичної карти пацієнта за допомогою форми, що представлена на рисунку 4.7. Описана форма має декілька сторінок для заповнення: personal info, chronic diseases, allergy, vaccination, laboratory tests, documents. Дані на цій формі необхідно заповнювати згідно форми 025 медичної карти пацієнта.










**healthcare** Olga Mironenko

Patients

**Ivan Maximov** EDIT PROFILE

Email: -  
Phone: 0935462853

**VISITS** NEW

Comment	Date	Doctor	Status	
<a href="#">Visit to ophth...</a>	14-12-2020 ...	In progress	In progress	  
<a href="#">Visit to card...</a>	14-12-2020 ...	In progress	In progress	  
<a href="#">No comment</a>	14-12-2020 ...	In progress	In progress	  

Previous Page 1 of 1 5 rows Next

Updated today by Olga Mironenko  
Updated 17 days ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko

Рисунку 4.6 – Сторінка профілю пацієнта

**healthcare** Olga Mironenko

Patients

Ivan Maximov

**Personal Info**

Chronic diseases

Allergy

Vaccination

Laboratory tests

Documents

**PERSONAL INFO**

First Name\*  
Ivan

Last Name\*  
Maximov

Age\*  
25

Gender\*  
 Male  Female

Email  
johnsmith@gmail.com

Phone Number\*  
0935462853

Location\*  
New York, NY

Work Place  
Medical Hospital

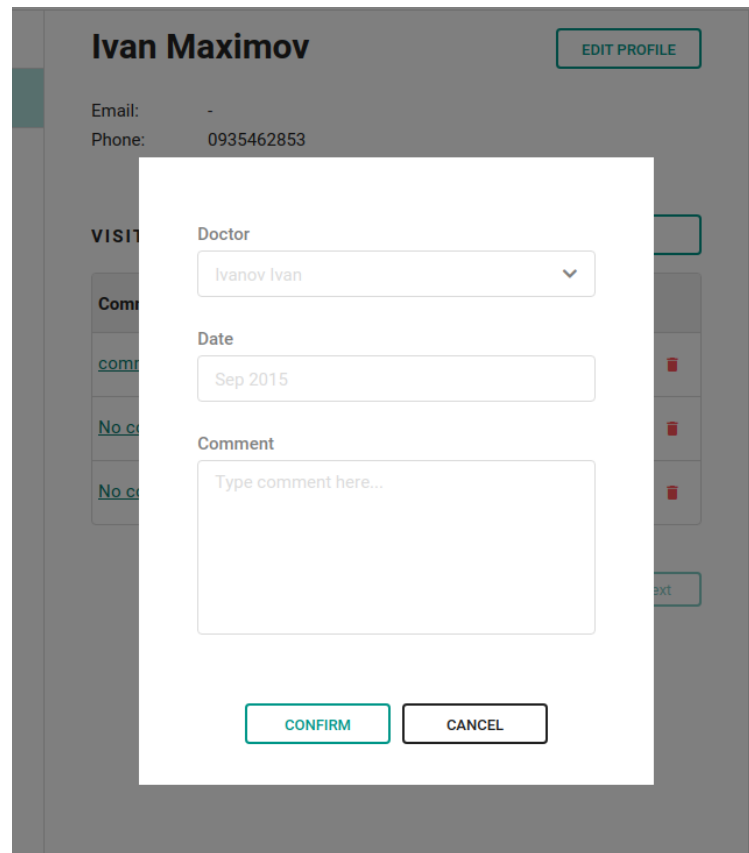
Work Position

RESET CHANGES UPDATE VIEW HISTORY

Updated today by Olga Mironenko  
Updated 17 days ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko  
Updated a month ago by Olga Mironenko

Рисунок 4.7 – Форма редагування персональних даних медичної карти пацієнта

Також на сторінці профіля пацієнта можна створювати нового запису до лікаря за допомогою форми, що представлена на рисунку 4.8.



The image shows a user interface for a patient's profile. At the top, the name "Ivan Maximov" is displayed next to an "EDIT PROFILE" button. Below the name, contact information is listed: "Email: -" and "Phone: 0935462853". A modal window is open in the center, titled "VISIT" (partially visible). The modal contains three input fields: "Doctor" with a dropdown menu showing "Ivanov Ivan", "Date" with a text input showing "Sep 2015", and "Comment" with a text area containing the placeholder "Type comment here...". At the bottom of the modal are two buttons: "CONFIRM" and "CANCEL".

Рисунок 4.8 – Форма створення запису до лікаря

Після створення запису до лікаря та початку прийому відкривається форма внесення даних прийому. Лікар може ввести нові симптоми, завантажити результати лабораторних досліджень та ввести результуюче заключення по завершенню прийому. Уся інформація, що стосується конкретного прийому має бути внесена на формі, що представлена на рисунку 4.9.

The screenshot displays a web interface for a healthcare system. At the top left is the 'healthcare' logo. At the top right, the user 'Olga Mironenko' is logged in, indicated by a profile icon and a name. On the left side, there is a navigation menu with the following items: 'Patients', 'Ivan Maximov', 'Visit to ophthalmologis...', 'Visit Info' (highlighted in teal), 'Laboratory tests', and 'Documents'. The main content area is titled 'VISIT INFO' and contains several input fields: 'Patient\*' with the value 'Ivan Maximov', 'Doctor\*' with the value 'Olga Mironenko', 'Status' with a dropdown menu showing 'In progress', and 'Summary\*' with the value 'Summary'. Below these is a 'Comment' field containing the text 'Visit to ophthalmologist'. At the bottom of the form, there are two buttons: 'REQUEST RECOMMENDATION' and 'FINISH VISIT'.

Рисунок 4.9 – Форма внесення даних прийому

Також на цій формі лікар має змогу скористатися рекомендаційною системою натиснувши кнопку «Request recommendation». На сторінці запити рекомендації лікар повинен обрати тип рекомендації. Результати роботи системи підтримки прийняття рішень надаються у вигляді pdf файлу, що описує заключення системи, дані на основі яких заключення було зроблено та рекомендації для пацієнта та лікаря. Сторінка запити рекомендації наведена на рисунку 4.10. Після аналізу отриманих результатів, лікарю необхідно оцінити роботи системи та підтвердити або відхилити отриману рекомендацію на формі, що наведено на рисунку 4.11. Після аналізу результатів, оцінки системи прийом вважається завершеним, а користувач повертається на сторінку профіля пацієнта

healthcare

Olga Mironenko

Patients

Ivan Maximov

Visit to ophthalmologis...

Recommendation

REQUEST RECOMMENDATION

Diabets

PDF LaTeX JSON

1 of 1

Звіт №16 від 14.12.2020 16:40

**Заключення:**  
Підозра на цукровий діабет

**Виявлені фактори ризику:**

- надлишкова вага: ІМТ > 25 кг/м<sup>2</sup>;
- концентрація глюкози у крові: 7 ммоль/л;
- вік: старше 45;
- спадковість.

**Рекомендації лікарю:**

- призначення додаткового обстеження лікарем-дієтологом;
- спостереження пацієнта;
- призначення пероральних цукрознижуючих препаратів;
- призначення інсулінотерапії;
- призначення обстеження гіперкалемії;
- додаткове обстеження для визначення типу цукрового діабету.

**Рекомендації пацієнту:**

- активний спосіб життя. Фізичні навантаження 150 хв на тиждень;
- моніторинг артеріального тиску;
- дотримання дієти. Дотримання ІМТ у межах від 17 до 25 кг/м<sup>2</sup>;
- повторне обстеження.

**Заключення та рекомендації валідні згідно наступних протоколів:**

1. Наказ Міністерства охорони здоров'я України 29 грудня 2014 р. № 1021 "Уніфікований клінічний протокол первинної, екстреної, вторинної (спеціалізованої) та третинної високоспеціалізованої медичної допомоги в галузі цукрового діабету 1 типу у молодих людей та дорослих"
2. Міністерство охорони здоров'я України Наказ від 21.12.2012 р. № 1118 "Про затвердження та оприлюднення медико-методичних документів зі стандартизації медичної допомоги при цукровому діабеті 2 типу"

CONFIRM RESULTS

Рисунок 4.10 – Форма запити рекомендації

1 of 1

Rate

5\*

Comment

Type comment here...

ACCEPT RESULTS DISCARD RESULTS

Рисунок 4.11 – Форма оцінки СППР та аналізу результатів

#### 4.6 Навчання методу прийняття рішення та аналіз результатів його роботи

Для навчання та тестування мережі використовувався набір даних Diabetes 130-US hospitals for years 1999-2008 Data Set. Набір знаходиться у вільному доступі на [UCI Machine Learning Repository](#). Дані про факти здоров'я, що використовуються у наборі – це витяг даних із 130 закладів медичної допомоги по всій території США за 10 років (1999–2008). Набір даних формувався на основі бази даних, яка складається з 41 таблиці База даних включає 74 036 643 унікальних відвідувань, які відповідають 17 880 231 унікальним пацієнтам. Оскільки ці дані представлялися інтегрованими системами охорони здоров'я, дані містять як стаціонарні, так і амбулаторні дані.

Усього набір містить 100000 прикладів та 55 ознак.

Набір даних був розділений на 2 частини у співвідношенні 30% на 70%: навчальний набір – складається з 70000 прикладів, тестовий набір – з 30000 прикладів.

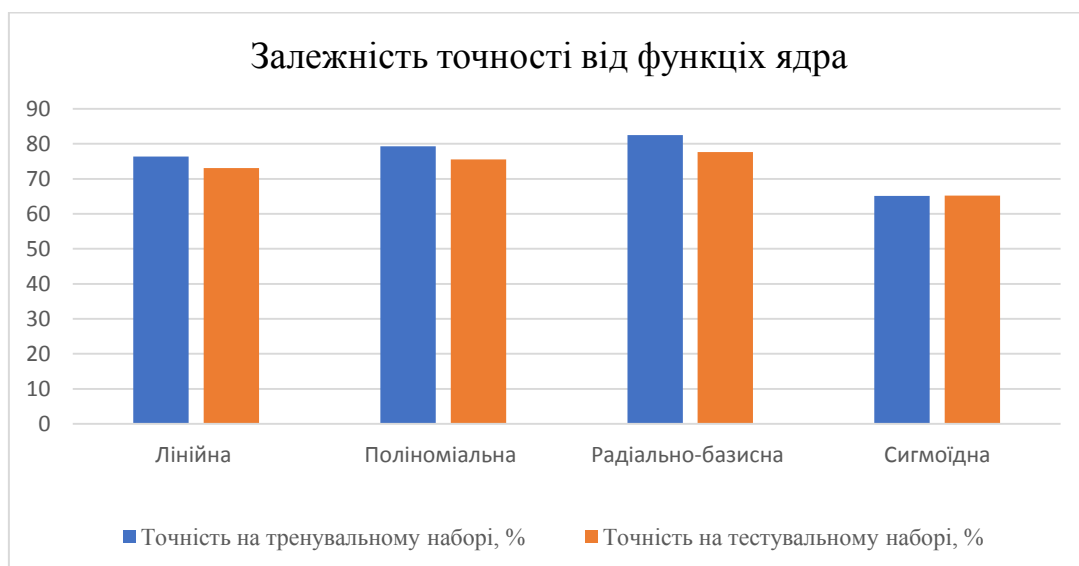
Для навчання та тестування алгоритму використовувалась мова програмування Python та бібліотека Scikit-learn. Дана бібліотека надає широкий вибір алгоритмів для вирішення класичних задач машинного навчання. Найбільшою перевагою бібліотеки є те, що вона працює на основі широковідомих математичних бібліотекою, які легко інтегруються між собою. Перелік математичних бібліотек, що використовувалися для навчання алгоритму: numPy, pandas.

Для навчання алгоритму використовувалися чотири функції ядра: лінійна, поліноміальна, радіально-базисна та сигмоїдна функції. У таблиці 4.2 наведено результати отриманої точності класифікації на тренувальному та тестувальному наборах.

Таблиця 4.2 – Точності навчання алгоритму в залежності від функції ядра

	Точність на тренувальному наборі, %	Точність на тестувальному наборі, %
Лінійна	76,4	73,06
Поліноміальна	79,3	75,52
Радіально-базисна	82,5	77,6
Сигмоїдна	65,1	65,2

На рисунку 4.12 представлено графік залежності точності від функції ядра.



Рисунку 4.12 – Графік залежності точності від функції ядра.

Як видно з таблиці 4.2 та рисунку 4.12 найвища точність та тренувальному та тестувальному наборах було отримана при застосування радіально-базисної функції ядра. Таким чином для навчання методу прийняття рішень було обрано радіально-базисну функцію ядра. Класифікаційний звіт наведено на рисунку 4.13:

	precision	recall	f1-score
0	0.80	0.90	0.84
1	0.71	0.52	0.60
accuracy			0.78
macro avg	0.75	0.71	0.72
weighted avg	0.77	0.78	0.77

Рисунок 4.13 – Класифікаційний звіт для методу опорних векторів з радіально-базисною функцією ядра

## ВИСНОВКИ

У атестаційній роботі представлені результати, котрі відповідають поставленій меті і є вирішенням задачі проектування системи підтримки прийняття рішень лікарем-терапевтом. Для вирішення задачі було розроблено медичні інформаційну систему до складу якої входить підсистема підтримки прийняття рішень на основі методу опорних векторів.

Проведені дослідження дозволили зробити наступні висновки:

– під час формування вимог до системи були визначені основні фактори, що впливають на процес розробки системи та успішність її введення в експлуатацію. Було проведено порівняльний аналіз можливих джерел даних та визначені особливості використання кожного з них. Використання апаратного забезпечення, що задовольняє вимоги системи, достатня та своєчасна технічна підтримка системи, непомітна інтеграція системи у клінічний робочий процес, актуальність та своєчасність надання рішень, достовірність, актуальність та достатня кількість даних, обраний алгоритм прийняття рішень є основними з факторів які впливають на успішність розробки та введення в експлуатацію системи.

– в рамках атестаційної роботи була сформовано основні функціональні та системні вимоги до систем підтримки прийняття рішень та обрано метод прийняття рішення, який відповідає вимогам предметної області. А саме дозволяє працювати з лінійно нероздільними даними та даними з великою кількістю атрибутів та є стійким до викидів.

– проведено навчання та тестування методу опорних векторів з різними функціями ядра та обрано функцію ядра, яка дозволяє отримати найвищу точність.

– - проведено дослідження мережі на реальних даних. Результати досліджень свідчать про нездатність системи до розпізнавання дуже схожих товарів, проте мережа показує високу точність на товарах, що мають яскраво вираженні ознаки та форми

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Система поддержки принятия врачебных решений ./ О. Атьков, Ю. Кудряшов, А. Прохоров, О. Касимов// Системы поддержки принятия врачебных решений.– 2013– №6.- С. 67-75.
2. P. Casimir.. Role of Clinical Decision Support Systems in Improving Clinical Practice // MOJ Clinical & Medical Case Reports – New York.: 2015. – № 2(6), P. 10.
3. Z. Omididan The role of clinical decision support systems in healthcare // A systematic review study. Jentashapir Sceintific-Research Quarterly. – 2011 – № 2(3). P. 125-34.
4. Berner ES Clinical Decision Support Systems: Theory and Practice 3rd. – New York: Springer, 2016 – 458p.
5. A. Nasser, H.Zaied, M. Elmogy and S. Abd Elkader Electronic health records: applications, techniques and challenges // International Journal of Computer Applications. – 2015 – №119(14). – P. 38-49.
6. M. Khalifa, O. Alswailem, Clinical Decision Support Knowledge Management: Strategies for Success // Studies in Health Technology and Informatics.– 2015.– №213. – P. 67-70.
7. Перова И.Г. Адаптивная обработка данных медико-биологических исследований методами вычислительного интеллекта // Восточно-европейский журнал передовых технологий. – Х.: 2014. – №1(67), с. 24-28
8. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.
9. I. Pliss, I. Perova Diagnostic Neuro-Fuzzy System and Its Learning in Medical Data Mining Tasks in Conditions of Uncertainty about Numbers of Attributes and Diagnoses // Automatic Control and Computer Sciences . – 2017. – 51 (6). – P. 391-398, DOI: 10.3103/ S0146411617060062
10. Perova, Ye. Bodyanskiy Fast medical diagnostics using autoassociative neuro-fuzzy memory // International Journal of Computing. – 2017. – 16 (1). – P. 34-40.

11. O. Turuta, I. Perova, A. Deineko Evolving flexible neuro-fuzzy system for medical diagnostic tasks // International Journal of Computer Science and Mobile Computing IJCSMC, Vol.4, Issue 8, 2015, P.475-480
12. Zhernova, P., Deyneko, A., Deyneko, Z., Pliss, I., Ahafonov, V. Data stream clustering in conditions of an unknown amount of classes / Zhernova, P., Deyneko, A., Deyneko, Z., Pliss, I., Ahafonov, V. // Advances in Intelligent Systems and Computing. – 2019. – Vol. 754. – P. 410–418.
13. Zhernova, P.Y. Adaptive Kernel Data Streams Clustering Based on Neural Networks Ensembles in Conditions of Uncertainty about Amount and Shapes of Clusters / Zhernova, P.Y., Deineko, A.O., Bodyanskiy, Y.V., Riepin, V.O. // Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018. – 2018. – Vol. 8478616. – P. 7–12.
14. Zhernova, P.Y. Data Stream Online Clustering Based on Fuzzy Expectation-Maximization Approach / Deineko, A.O., Zhernova, P.Y., Gordon, B., Zayika, O.O., Pliss, I., Pabyrivska, N. // Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018. – 2018. – Vol. 8478517. – P. 171–176.
15. L. Karpov Hybrid approach to building decision support system // Institute for System Programming of the Russian Academy of Sciences. – 2013. – P. 45-56.
16. Fraccaro P, O’Sullivan D, Plastiras P, O’Sullivan H, Dentone C, Di Biago A, Weller P. Behind the screens: Clinical decision support methodologies – A review. Health Policy and Technology. 2014;
17. Гвоздева В. А. Основы построения автоматизированных информационных систем / В. А. Гвоздева, И. Ю. Лаврентьева. – М.: ФОРУМ: ИНФРА-М, 2007. – 320 с.
18. V.Vapnik. The nature of Statistical Learning. – New York: Springer, 1995 – 452p.
19. Raj Bridgelall Introduction to Support Vector Machines // International Journal of Computer Science and Mobile Computing, 2017– №6.- P. 67-75
20. O. Turuta, A. Deineko, I. Perova , Y. Kutsenko M. Shalamov Evolving Neural Network for Kernel Principal Component Analysis // International Journal of Computer

Science and Mobile Computing IJCSMC, Vol.4, Issue 9, 2015, P.356-363  
Scheinberg K. An efficient implementation of an active set method for svms // J. Mach. Learn. Res. — 2006. — Vol. 7. — Pp. 2237–2257

21. И.Ю. Коцюба. Основы проектирования информационных систем Учебное пособие / И.Ю. Коцюба, Чунаев А.В., А.Н. Шиков. – Санкт-Петербург, 2015. – 206 с.

22. Cloud Computing [Электронный ресурс] – Режим доступа до ресурсу: <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/>

23. MySQL [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mysql.com/>

24. MSSQL Server [Электронный ресурс] – Режим доступа до ресурсу: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019>

25. Oracle Database [Электронный ресурс] – <https://www.oracle.com/database/index.html>

26. Кузнецов С. Д. Основы баз даних : навчань. посібник / Сергій Дмитрович Кузнецов. - 2-е видавництво, испр. - М.: Інтернет-університет інформаційних технологій : Біном. Лабораторія знань, 2007. – 484 с.