

Харківський національний університет радіоелектроніки

Факультет _____ Центр післядипломної освіти _____
 Кафедра _____ Програмної інженерії _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 (код і повна назва спеціальності)
 Тип програми _____ освітньо-наукова програма _____
 Освітня програма _____ Інженерія програмного забезпечення _____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20__ р.

ЗАВДАННЯ**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студенту _____ Сосновському Григорію Ігоровичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів прогнозування при розробці хмарного кросплатформного застосунку для аналізу психологічного стану пацієнтів.
затверджена наказом по університету від 03.04.2023 р. № 83Стз
2. Термін подання студентом роботи до екзаменаційної комісії 01.05.2023 р.
3. Вихідні дані до роботи дослідити методи прогнозування при аналізі психологічного стану пацієнтів та реалізувати їх у вигляді програмного комплексу (система хмарного кросплатформного застосунку), проаналізувати отримані результати.
3. Перелік питань, що потрібно опрацювати в роботі вступний опис питання, аналіз предметної галузі, збір даних, ущільнення даних, побудова моделі, екстраполяція та оцінка прогнозу, наведення моделі наївного баєсового класифікатора, аналіз результатів у висновках.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Визначення методу прогнозування	01.04.2023	Виконано
2	Аналіз предметної галузі	01.04.2023 – 05.04.2023	Виконано
3	Побудова моделі	05.04.2023 - 10.04.2023	Виконано
4	Проектування програмного забезпечення	10.04.2023 - 11.04.2023	Виконано
5	Розробка програмного забезпечення	11.03.2023 - 25.04.2023	Виконано
6	Підготовка пояснювальної записки	25.04.2023	Виконано
7	Підготовка презентації та доповіді	25.04.2023 – 01.05.2023	Виконано
8	Перевірка на плагіат	01.05.2023	Виконано
9	Перевірка на нормоконтроль	02.05.2023	Виконано
10	Оцінка кваліфікаційної роботи керівником	10.05.2023	Виконано
11	Захист кваліфікаційної роботи	10.05.2023	Виконано

Дата видачі завдання «01» квітня 2023 р.

Студент _____
(підпис)

Сосновський Г. І. _____

Керівник роботи _____
(підпис)

доц., к.т.н, Назаров О. С. _____
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить 72 стор., 12 рис., 4 табл., 13 джерел, 7 додатків.

МОДЕЛЬ ПРОГНОЗУВАННЯ, НАЇВНИЙ БАЄСІВ КЛАСИФІКАТОР, ХМАРНЕ СЕРЕДОВИЩЕ, AZURE, BLAZOR WASM, BLOBS STORAGE, C#, .NET 6, SPEECH-TO-TEXT, TEXT ANALYTICS

Об'єктом дослідження є дані, пов'язані з роботою спеціалістів з емоційної реабілітації в Healthcare домені. Предметом є метод прогнозування та розробка хмарної програмної системи для аналізу психологічного стану пацієнтів.

Мета дослідження – створити модель прогнозування у вигляді програмного застосунку «Rehabilife».

Було проведено аналіз предметної галузі, побудована модель наївного баєсівського класифікатора. У результаті створено хмарну програмну систему «Rehabilife».

AZURE, BLAZOR WASM, BLOBS STORAGE, C#, CANVAS, CLOUD ENVIRONMENT, NAIVE BAYES CLASSIFIER, .NET 6, PROBABILISTIC MODEL, SPEECH-TO-TEXT, TEXT ANALYTICS

The object of research is data related to the work of specialists with emotional rehabilitation in the Healthcare domain. The subject is the probabilistic method and the development of a cloud-based software system for the analysis of the psychological state of patients.

The purpose of the research is to create a probabilistic model with the "Rehabilife" software application.

An appropriate was conducted, a naïve Bayes classifier model was built. As a result, the cloud software system "Rehabilife" was introduced.

Умови публікації пояснювальної записки

Я, _____ Сосновський Григорій Ігорович _____
(прізвище, ім'я, по батькові)
студент групи ППЗдм-21-1 здобувач вищої освіти на другому (магістерському)
рівні

кафедра _____ програмної інженерії _____
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему «Дослідження методів прогнозування при розробці хмарного кросплатформного застосунку для аналізу психологічного стану пацієнтів», що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	8
1 Опис предметної області	11
1.1 Аналіз моделей прогнозування	11
1.2 Аналіз предметної галузі.....	17
1.3 Аналіз аналогів.....	23
2 Розробка програми для аналізу даних.....	27
2.1 Критерії системи	27
2.2 Технології.....	29
2.3 Ціннісна пропозиція.....	31
2.4 Канвас бізнес-моделі.....	33
2.5 Вихідні форми	35
2.6 Робочий процес застосування.....	38
2.7 Модель прогнозування	41
Висновки	52
Перелік джерел посилання	53
Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії	55
Додаток А (перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії)	56
Додаток Б (посилання на проект)	57
Додаток В (інструкція користувача)	58
Додаток Г (апробація роботи).....	60
Додаток І (слайди презентації).....	64
Додаток Д (звіт результатів перевірки на унікальність тексту)	71

Додаток Е (експертний висновок нормоконтроль).....	72
--	----

ВСТУП

Наразі розширюється кількість ситуацій, в яких постає необхідність та доцільність допомоги з реабілітації з допомогою психологічних знань та методів. Оскільки значно розповсюджуються посттравматичні стресові розлади особистості на фоні пережитої психологічно негативної та ментально травматичної події, це зумовлює допомогу спеціаліста-реабілітолога для фізіологічного й психологічного відновлення людини та повернення її до суспільного життя в післявоєнний період.

Посттравматичний стресовий розлад (ПТСР) – це психіатричний стан, який призводить до значної інвалідності та порушення функцій. Найчастіше виникає у колишніх військових, як так званий «комбатантський синдром». Своєчасне втручання, застосоване невдовзі після травматичної події, може зменшити настання ПТСР та пов'язане з ним довгострокове порушення. Враховуючи складнощі, пов'язані з раннім втручанням, майже неможливо зробити це вчасно для усіх, хто зазнав таких подій, радше рекомендується підхід «перевірити та лікувати», за якого виявляють та лікують осіб із високим ризиком ПТСР. Ключовою перешкодою для надання раннього втручання є нездатність точно ідентифікувати осіб із високим ризиком ПТСР у цьому гострому посттравматичному періоді (<30 днів після події). Проблемна здатність виявляти тих, хто входить до групи ризику, пов'язана з обмеженим розумінням того, як розвиваються симптоми посттравматичного стресового розладу, і, отже, які фактори є найбільш корисними для визначення ризику розладу [1].

Така подія – це шокуючий, страшний та небезпечний досвід, який сильно впливає на людину в емоційному плані. Це підвищує ймовірність розвитку проблем із психічним здоров'ям, тривоги та депресії, а також погіршення життя у дорослому віці [2]. Настрій в цей період також дуже важливий. Як і у будь-якій розумовій діяльності можна відслідковувати його динаміку.

Наприклад (рисунок 1.1):

- відбулося неочікуване та неприємне – шок;
- потім, повертаються базові емоції (відчай, жах, заперечення і т.д.);

- після спаду завжди – ейфорія;
- як в свою чергу не буває постійно.



Рисунок 1.1 – Коливання настрою (джерело [1])

Реабілітація спрямована на створення якіснішого, з соціального погляду, рівня життя особи, яка її і потребує. Важливим у реабілітації є порушені можливості особи, взаємозв'язки, які тимчасово руйнуються між нею і суспільством та потребують відновлення.

Використовуючи методи прогнозування можна вивчати та навіть прогнозувати важливі показники з вищевказаних вище. Зокрема, емоційний стан, що є вкрай важливим складником у даному контексті. Таким чином, методи прогнозування ґрунтуються на трансфері знань та досвіду з минулого до невизначеного майбутнього. Вони створюються на припущенні, що отримані раніше дані не є відмінними від того, що станеться у майбутньому. Виняток становлять тільки ті змінні, які точно розпізнані моделлю прогнозування.

У даній роботі зроблено кроки до створення прогностичної моделі, яка допоможе у дослідженні того, які існують кореляції між симптомами ПТСР, що присутні незабаром після травми та через 30 днів після події, та чи можуть ці кореляції бути виявлені за допомогою навчання з учителем методами машинного навчання (МН). Сьогодні дослідження показали, що технології МН є

ефективними для прогностичного моделювання в медичному середовищі, наприклад, для прогнозування прогнозів раку [3]. Однак такі моделі ще не стали регулярно впроваджуватися в психіатричних захворюваннях. Крім того, моделі, створені за допомогою МН, можуть бути ретельно перевірені за допомогою технік перехресної перевірки, що збільшує впевненість у їхній відповідності.

Зокрема, наївний баєсів класифікатор, про який піде річ у даній роботі, рекомендований саме для класифікації та передбачення емоційних станів.

Метою магістерського дослідження є обґрунтування планування реалізації програмного програмної системи для спеціалістів з реабілітації в воєнний та післявоєнний періоди. В контексті даної роботи методи прогнозування допоможуть з алгоритмічною складовою. Зокрема, буде використано наївний баєсів класифікатор.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз методів прогнозування

За допомогою спеціальних алгоритмів прогнозування можна знаходити та передбачати різні закономірності. Одна з таких сфер стосується реабілітаційної психології, а саме емоційних станів.

Тут на допомогу приходить математичне моделювання на основі вибірки окремих прецедентів – результатів спостережень досліджуваного процесу або явищ, що включають значення прогнозованої величини Y і змінних X_1, \dots, X_n . У цих випадках для вирішення завдань діагностики і прогнозування можуть бути використані методи, ґрунтовані на навчанні по прецедентах. Вибірку таких прецедентів також прийнято називати навчальною вибіркою. Така вибірка має вигляд (1.1):

$$S_t = \{(y_1, x_1), \dots, (y_m, x_m)\}, \quad (1.1)$$

де y_j – значення змінної Y для об'єкту,

x_j – значення вектору змінних X_1, \dots, X_n для об'єкту s_j ,

m – число об'єктів в S_t .

Методи, засновані на навчанні по прецедентах, також прийнято називати методами МН. Розглянемо деякі з них та обґрунтуємо свій вибір для цілей цієї роботи.

1.1.1 Метод k-найближчих сусідів

Метод k-найближчих сусідів є одним з простіших та перших методів машинного навчання. Він є непараметричним методом з учителем, який базується на запам'ятовуванні навчальної вибірки та передбаченні класу нового екземпляра за допомогою класів його найближчих сусідів. Цей метод ґрунтується на припущенні, що близькі об'єкти зазвичай належать до одного класу. Таким чином,

він уникає припущень про функціональну форму та попереджає помилки при зміні розподілу даних.

Загальний підхід для визначення класу невідомого екземпляра q полягає в дистанційно зваженому голосуванні (1.2), де кожен сусід має можливість проголосувати на підставі голосів з обернено пропорційною відстанню до q .

$$V(y_j) = \sum_{c=1}^k \frac{1}{d(q, x_c)^n} 1(y_j, y_c) \quad (1.2)$$

Отже, голос, що належить до класу y_j від сусіда x_c , розділяється на відстань до цього сусіда. Таким чином, функція $1(y_j, y_c)$ поверне 1, якщо класи збігаються, а в іншому випадку – 0. Зазвичай, у рівнянні 3^n ця функція буде мати значення 1, але значення менше 1 можуть бути використані для подальшого зменшення впливу більш віддалених сусідів.

Переваги і недоліки:

- доволі простий, легко реалізовується. Для класифікації зчитує цілий набір даних, щоб дізнатися K найближчих сусідів;
- не створює явно будь-яку модель (на основі введення даних);
- якщо стоїть задача з багатьма класами (як у випадку в даній роботі), потребує зусиль для її реалізації;
- використовується як для класифікації, так і для регресії.

Для цілей даного дослідження не підходить, оскільки необхідно додавати емоційні стани в реальному часі. Також, в даній роботі можливе використання багатьох класів.

1.1.2 Наївний баєсів класифікатор

Це імовірнісний класифікатор, заснований на теоремі Баєса (опис ймовірності події, спираючись на обставини, що могли би бути пов'язані з цією подією) з наївними припущеннями про незалежність [4]. Якщо на підставі значень змінних можна достовірно виявити приналежність спостереження до певного класу, то класифікатор Баєса надасть інформацію про імовірність належності до цього класу.

Наївний класифікатор Баєса можна порівняти з людиною, яка намагається передбачити погоду на основі вигляду неба, температури та вологості. Людина може дізнатися з минулого досвіду, що червоне небо вранці, висока температура та низька вологість часто призводять до спекотного та сухого дня. Використовуючи ці знання та застосовуючи їх до поточних погодних умов, людина може спрогнозувати погоду з достатньою точністю. Подібним чином наївний класифікатор Байєса вчиться на минулих позначених даних і застосовує ці знання, щоб передбачити клас нових, не позначених даних на основі ймовірності спостереження певних особливостей кожного класу.

Незважаючи на спрощені припущення, цілком добре працює в складних ситуаціях (використовується, наприклад, при аналізі спаму). Перевагою є те, що для оцінки параметрів, необхідних для класифікації, потребує незначну кількість навчальної вибірки.

Наприклад, для класифікації спам повідомлень створимо словник слів та вірогідності їх відношення до певних класів. Якщо класів повідомлень є два, тоді вірогідність класів p буде дорівнювати 0.5. У разі якщо значень n -кількість, тоді вірогідність кожного класу $p(c)$ дорівнює відношенню 1 до n . Вірогідність причетності повідомлення до класу розраховується наступним чином (1.3):

$$p(m) = p(c) \sum_{i=1}^n p(w) \quad (1.3)$$

де $p(m)$ – вірогідність відношення повідомлення до класу,

$p(c)$ – вірогідність класу,

$p(w)$ - вірогідність відношення слова у тестовому реченні до певного класу.

Якщо вірогідність слова у назві рівна 0, тоді і загальна також 0. Для уникнення можна нормувати всі ймовірності у словнику за таким прикладом (1.4):

$$p(nw) = n * p(uw) * p(c) / n + 1 \quad (1.4)$$

де $p(nw)$ - нормована вірогідність слова до класу,

$p(uw)$ - нормована вірогідність слова до класу,

n – кількість повторень слова.

Переваги і недоліки:

- простий, легко реалізовується. Не потребує значних обчислювальних витрат для навчання і класифікації;
- є прогнозуванням в реальному часі, а значить модель швидко масштабується;
- працює з багатьма класами, можна задати багато значень для базової задачі;
- відомий для використання в аналізі настроїв для виявлення позитивних і негативних емоцій.

Підемо далі. Якщо імовірність досягається лише для одного з класів, то класифікований об'єкт повинен бути чітко віднесений тільки до цього класу. Тепер припустимо, що класифікатор використовується для класів K_1, \dots, K_L , які відносять що об'єкти $\nu_1(x), \dots, \nu_L(x)$ до деякої точки x . З припущенням про зміст інформації про класи змінними X_1, \dots, X_n впливає, що існує нескінченна кількість об'єктів з фіксованим x , а розподіл класу не залежить від отриманого прогнозу. Тому, ймовірність віднесення об'єкту до класу K_i , який фактично належить до класу K_i насправді складає (1.4):

$$[1 - \nu_j(x)]P(K_i|x) \quad (1.4)$$

Загальна ймовірність невірних класифікацій в точці x складає (1.5):

$$\sum_{i=1}^L [1 - \nu_i(x)]P(K_i|x) = 1 - \sum_{i=1}^L \nu_i(x)P(K_i|x). \quad (1.5)$$

Задача пошуку мінімуму помилки зводиться до завдання лінійного програмування виду (1.6):

$$\begin{aligned}
\sum_{i=1}^L \nu_i P(K_i|x) &\rightarrow \max_{\nu_1, \dots, \nu_L}, \\
\sum_{i=1}^L \nu_i &= 1, \\
\nu_i &\geq 0, i = 1, \dots, L.
\end{aligned}
\tag{1.6}$$

Для обчислення умовних математичних очікувань $E(Y|x)$ або умовної вірогідності $P(K_i|x)$, $i=1, \dots, L$, потрібні знання конкретного виду імовірнісних розподілів, властивих вирішуваній задачі. Такі знання в принципі можуть бути отримані з використанням методу максимальної правдоподібності.

1.1.3 Метод максимальної правдоподібності

Цей метод використовується для опису вірогідності розподілу по вибірках даних. У загальному випадку вимагає припущень про тип розподілів. Значення параметрів, що визначають конкретний вид розподілів, аналізуються шляхом максимізації функціонала правдоподібності (1.7):

$$L(\tilde{S}_t, \theta_1, \dots, \theta_r) = \prod_{j=1}^m p(y_j, \mathbf{x}_j, \theta_1, \dots, \theta_r).
\tag{1.7}$$

В деяких випадках можна аналітично розв'язати умови першого порядку функції ймовірності, наприклад, метод найменших квадратів для моделі лінійної регресії максимізує ймовірність, припускаючи, що всі результати мають нормальний розподіл з однаковою дисперсією. З точки зору баєсівського підходу, метод максимальної правдоподібності еквівалентний максимальному оцінюванню з рівномірним або нормальним пріоритетним розподілом.

У випадку пошуку параметрів, метод максимальної правдоподібності використовується різними алгоритмами МН. Оптимальні коефіцієнти можуть бути використані для створення моделі, яка може передбачати значення близькі до 1 для класу за замовчуванням та значення, близькі до 0 для інших. Максимальна

вірогідність регресії полягає в пошуку значень коефіцієнтів, що мінімізують похибку ймовірностей, передбачених моделлю.

Для цілей дослідження не підходить, оскільки необхідно додавати емоційні стани в реальному часі.

1.1.4 Древа рішень

Наступний алгоритм називається деревом рішень (decision tree). Він є одним з найпростіших і найбільш поширених алгоритмів машинного навчання для задач класифікації та регресії. Древа рішень є часто використовуваними у бізнесі та науці, оскільки вони дозволяють відносно легко інтерпретувати результати та пояснити, які атрибути були використані для прийняття рішення.

Процес побудови дерева рішень починається з кореневого вузла, який включає всі приклади з тренувального датасету. На кожному кроці процесу вибирається атрибут, який найкраще розділяє приклади на підмножини, щоб краще відрізнити класи. Для визначення кращого атрибута можна використовувати різні метрики, такі як ентропія, джінні-індекс або помилка класифікації.

Після вибору атрибута вузол розбивається на дочірні вузли, в які приклади розподіляються в залежності від значення обраного атрибута. Процес рекурсивно повторюється для кожного дочірнього вузла до тих пір, поки не буде досягнуто критерій зупинки. Критерій зупинки може бути різним, наприклад, досягнення максимальної глибини дерева, досягнення мінімальної кількості прикладів у вузлі або досягнення однорідності класів у вузлі. Після побудови дерева рішень можна використовувати його для класифікації нових прикладів. Для цього новий приклад пройде по дереву від кореневого вузла до листового вузла, пропустивши його.

Індекс неоднорідності обчислюється для довільної вибірки $\gamma_e(\tilde{S})$, що містить об'єкти з класів K_1, \dots, K_L . Разом з тим приміняються наступні типи видів індексів:

- індекс ентропії неоднорідності;
- індекс Джини;
- індекс помилкової класифікації.

Індекс ентропії неоднорідності обчислюється за формулою (1.8):

$$\gamma_e(\tilde{S}) = - \sum_{i=1}^L P_i \ln P_i, \quad (1.8)$$

де P_i - доля об'єктів класу K_i у вибірці.

При цьому приймається, що $0 \ln(0) = 0$. Найбільше значення приймає при рівності вірогідності класів. Найменше значення відповідно, коли всі об'єкти належать до одного класу.

При побудові класифікатора дерева рішень не потрібно знати предметну область, і тому він підходить для дослідницького виявлення знань. Дерева рішень можуть обробляти дані великого розміру. Якщо використовувати складне дерево, то рішень класифікатор має високу точність, оскільки складність дерева прямо пов'язана з цим. Тому даний класифікатор було відкинуто.

Висновок: після аналізу вищевказаної інформації, шляхом експериментального дослідження визначено, що баєсів класифікатор є найбільш підходящим варіантом у цьому разі оскільки доведеться працювати з багатьма класами. Перевага наївного класифікатора Баєса полягає в тому, що для оцінки параметрів, необхідних для класифікації, потрібно лише невелику кількість навчальних даних. Оскільки припускаються незалежні перемінні, необхідно визначити тільки їх варіативність для кожного класу, а не всю коваріативну матрицю, це пов'язано з тим, що різниця лінійної залежності двох випадкових величин між незалежними змінними дорівнює 0. Крім того, баєсів класифікатор активно використовується в аналізі настроїв для виявлення позитивних і негативних емоцій.

1.2 Аналіз предметної галузі

Реабілітаційна психологія – це вивчення та застосування психологічних

принципів для осіб, які мають суттєві ментальні проблеми. Психологи-реабілітологи, часто в рамках команд, оцінюють і лікують когнітивні, емоційні та функціональні труднощі, а також допомагають людям подолати перешкоди для участі в житті. Психологи-реабілітологи беруть участь у практичних та дослідницьких роботах із широкою метою сприяння відновлення емоційного стану.

Поглиблені науково-теоретичні відомості:

- спеціальні знання про те, як оцінювати та лікувати емоційне подолання, психічний та психологічний статус та поведінку, яка сприяє позитивній адаптації;
- детальне знання конкретних станів, сімейної динаміки, командної динаміки та лідерства, соціальних впливів, а також законів і політики.

Вивчення емоційних станів у різних процесах життя в даний час набуває все більшого поширення. Це пов'язано насамперед з високою динамікою життєдіяльності людини, інтенсифікацією комунікаційних зв'язків та деяких інших особливостей сучасної епохи, які зумовили необхідність розвитку в практичні психологічні засоби підвищення потенціалу людини, вдосконалення її адаптаційних механізмів. Найбільш дослідженими предметами можна вважати функціональне значення емоцій, емоційних явищ з іншими психічними процесами.

Меншою мірою досліджено розвиток емоції в онтогенезі, еволюція та зміни емоційних станів, механізми регулювання. Показовими в цьому плані є праці вчених Р. Вудворца, Д. Ліндслі, П. Фресса, Г. Рейковського, К. Ізарда, а також: П. М. Якобсон, Б. І. Додонова, П.В. Симонов і В. Куликов. При висвітленні проблем психічного та емоційного контролю емоційних станів їх взаємозв'язку з поведінкою людини певне значення мають дослідження Л.М. Аболіної, А. Прохорова, А. Валлон Г.А., Вартанян Е. Петрова, Т.П. Гаврилова, А.Є. Олинаннікова, А.Й. Гозман, І. Рейковського та ін. Особливо актуальне дослідження станів і рівнів тривожності у зв'язку з навчанням і вихованням процесу, де відбувається розуміння психічних станів і управління ними в процесі діяльності пізнання і комунікація є однією з основних сторін. У розумних істот, зокрема у людей, важливим засобом вираження є мовлення, яке дозволяє обмінюватися

інформацією про власний стан та події навколо. Емоційна функція має важливий регуляційний вплив на процеси пізнання, що полягає в експресивному та комунікативному передаванні інформації про стан та події навколишнього середовища.

Емоції – це внутрішня мова як система сигналів, за допомогою яких суб'єкт навчається про необхідність значущості того, що відбувається. «Особливість емоцій полягає в тому, що вони негайно заперечують зв'язок між мотивацією і реалізацією відповідає цьому пояснення. Емоції в діяльності людини виконують функцію оцінки її ходу та результатів. Вони організовувати діяльність, стимулюючи і направляючи її» [5].

Список шести основних емоцій (рисунок 1.2):

- печаль;
- щастя;
- страх;
- гнів;
- здивування;
- огида.



Рисунок 1.2 – Емоційний спектр (джерело [5])

Подібним чином у 1980-х роках психолог Роберт Плутчик визначив вісім основних емоцій, які він згрупував у пари протилежностей, включаючи радість і смуток, гнів і страх, довіру і огиду, а також подив і очікування. Ця класифікація відома як колесо емоцій, і її можна порівняти з кольоровим колесом, оскільки певні емоції, змішані разом, можуть створювати нові складні емоції.

Теорії та гіпотези про емоції сягають багатьох століть. Справді, базові або первинні емоції згадуються в Книзі прав, китайській енциклопедії першого століття. Емоції набагато важче виміряти та правильно визначити, ніж багато інших людських реакцій. Значна частина досліджень, проведених в емоційній психології, присвячена основним емоціям, нашим психологічним і поведінковим реакціям і ролі емоційного інтелекту в нашому житті.

Інформаційні технології створені не лише для великих компаній, але і для звичайних людей, яким потрібні зручні робочі інструменти, в тому числі інформаційні технології. Наприклад, психолог з реабілітації змушений вести складні підрахунки для аналізу емоційного стану. Якщо за це буде відповідати програмна система, яку до речі можна розвернути будь-де та масштабувати із хмарного середовища, то це б значно спростило б спеціалісту життя. Ці завдання легко піддаються автоматизації. Тому очевидна необхідність створення інформаційної системи, яка дозволяє виконувати подібні задачі.

В контексті роботи допомогою спеціалісту з реабілітації є система, що визначає емоційний стан пацієнта. Вона складається з наступних частин:

- ведення прийому та запис сесії;
- транскрибування голосу в текст;
- AI-аналіз тексту на предмет емоційного забарвлення та виявлення найбільших скарг.

З цим допоможуть наступні хмарні технології від Azure.

Перша, Speech service (рисунок 1.3). Speech service надає можливості перетворення мовлення в текст і тексту в мовлення за допомогою ресурсу Azure Speech. Ви можете транскрибувати мовлення в текст із високою точністю,

відтворювати природні голосові перетворення тексту в мовлення, перекладати розмовний аудіо та використовувати розпізнавання мовця під час розмов [6].

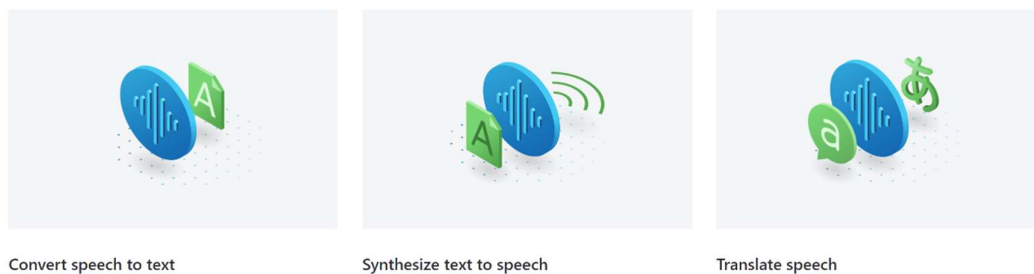


Рисунок 1.3 – Azure Speech алгоритм (джерело [6])

Загальні сценарії Speech service включають:

- субтитри: синхронізація субтитрів з вхідним звуком, застосування фільтрів нецензурної лексики, отримання часткових результатів, застосування налаштування та визначати розмовні мови для багатомовних сценаріїв;
- створення аудіоконтенту: можна використовувати нейронні голоси, щоб зробити взаємодію з чат-ботами та голосовими помічниками більш природною та привабливою, перетворювати цифрові тексти, такі як електронні книги, в аудіокниги та вдосконалювати автомобільні навігаційні системи;
- дзвінки на лінії підтримки: транскрибування дзвінків в режимі реального часу або оброблення групи дзвінків, редагування особистої інформації та витягати статистичних даних, наприклад настрої, щоб допомогти вам у використанні кол-центру;
- вивчення мови: відгуки про оцінку вимови тим, хто вивчає мову, транскрипція в режимі реального часу для дистанційного навчання та читання вголос навчальні матеріали за допомогою нейронних голосів;
- голосові помічники: створення природних, людських розмовних інтерфейсів для програм. Функція голосового помічника забезпечує швидку та надійну взаємодію між пристроєм і реалізацією помічника.

Speech-to-text є технологією, що перетворює вимовлені слова на текст за допомогою автоматичного процесу. Microsoft Azure пропонує сервіс, який використовує алгоритми машинного навчання для точного транскрибування аудіозаписів у текст. Цей сервіс можна використовувати в різних застосуваннях, таких як транскрипція промов, лекцій, інтерв'ю та дзвінків на лінії підтримки. Технологія також містить функції, такі як розпізнавання мовця та вставлення пунктуації, що поліпшують точність транскрипції.

Одна з цікавих особливостей сервісу Speech-to-text в Azure полягає в тому, що він може розпізнавати різні мовні акценти та діалекти, що дозволяє забезпечувати точні транскрипції навіть для людей з різними вимовними особливостями. Це, до прикладу, буде корисно у рамках даного дослідження коли ми будемо працювати з людьми з ПСТР та можливими травмами речового апарату.

Крім того, сервіс включає в себе можливість розпізнавання імен та технічних термінів, що допомагає досягнути ще більшої точності в транскрипції спеціалізованого контенту. Також слід зазначити, що Speech-to-text в Azure може бути легко інтегрований з іншими сервісами хмарного рішення Azure, що робить його чудовим вибором для розробників та бізнес-клієнтів, які шукають зручний та ефективний спосіб отримання письмового тексту з аудіозаписів.

Друга – Cognitive Service for Language. Це хмарний сервіс, що надає функції обробки природної мови (NLP) для розуміння й аналізу тексту. Ця служба застосовується, щоб створювати програми за допомогою веб-мов, REST API та клієнтських бібліотек. Цей мовний сервіс об'єднує Text Analytics, QnA Maker і LUIS, а також надає кілька нових функцій [7].

Ці функції такі:

- попередньо налаштовані, що означає, що моделі AI, які використовує функція, не можна налаштувати. Ви просто надсилаєте свої дані та використовуєте результати функції у своїх програмах;
- можливість налаштування, що означає, що ви навчите модель штучного інтелекту за допомогою наших інструментів відповідно до ваших даних.

В контексті роботи буде використовуватися Text Analytics. Azure Text Analytics – це сервіс обробки природньої мови, який дозволяє аналізувати та витягувати корисну інформацію з текстових даних. Сервіс пропонує інструменти для автоматичної обробки тексту, такі як аналіз тональності, розпізнавання мови, виявлення ключових слів та фраз, а також навіть розпізнавання іменованих сутностей.

Однією з основних можливостей Azure Text Analytics є аналіз емоційного забарвлення тексту, що дозволяє визначати настрій або тон повідомлення. Це дуже корисна функція для моніторингу бренду в соціальних медіа або відстеження реакції на новини або події в різних медіа. Azure Text Analytics також може бути використаний для автоматизованої категоризації тексту за темами, що дозволяє швидко визначати головні теми та ідеї, які містяться у текстових даних. Інший важливий аспект сервісу – його можливості до МН. Text Analytics може бути використано для розробки моделей штучного інтелекту, що дозволяють автоматично визначати категорії або емоційний забарвлення для нових текстових даних. Усі ці функції роблять Text Analytics потужним інструментом для обробки та аналізу текстових даних, що має багато потенційних застосувань у бізнесі, соціальних науках та інших галузях.

1.3 Аналіз аналогів

Застосунок «Rehabilife» призначений для того, щоб підвищити продуктивність спеціалістів з реабілітації у веденні своїх справ, тобто користувач за допомогою програмного продукту матиме можливість вести облік пацієнтів, записувати їх дані, та вести голосові сеанси, які обробляються сервісами розпізнавання голосу та аналізу емоційного забарвлення тексту.

Результати аналізу ринку дозволили виявити конкурентоспроможність. Це пов'язано, з ідеєю підтримки діяльності реабілітолога, прямих аналогів у світі немає, а конкурентом може бути застосунок, який має можливість вести базу даних пацієнтів та аналізувати голос та текст. Також, багато подібних програм медичного

направлення використовуються в межах однієї держави, а «Rehabilife» поширюватиметься в усьому світі (таблиця 1.1).

Таблиця 1.1 – Характеристика інновацій

Класифікаційна ознака	Характеристика інновації за ознакою класифікації
1. За предметом інноваційної діяльності	1.2 процесні інновації
2. Масштаб новизни інновації	2.1 нові для галузі в світі;
3. За місцем в системі (на підприємстві, у фірмі)	3.3 інновації системної структури підприємства (управлінської, виробничої, технологічної).
4. За сферою застосування	4.1 організаційні; 4.6 соціальні;
5. За глибиною внесених змін	5.1 радикальні (базові);
6. За задоволенням потреб	6.1 задовольняють існуючу потребу;
7. За напрямом реалізації	7.2 призначені для продажу (трансферу);
8. За приналежністю до розроблення	8.1 розроблені власним силами;
10. За джерелами фінансування	9.3 змішане фінансування;

Суть ідеї. Застосунок «Rehabilife» дозволяє користувачам:

- вести інформацію про кожного клієнта;
- записувати та завантажувати голосове відображення сеансу;
- з допомогою AI аналізувати емоційний стан пацієнта.

Розглянемо найбільш популярні на 2023 рік аналоги.

Lexalytics Intelligence Platform – це платформа аналітики даних із функціями обробки природної мови (NLP) і аналітики тексту. Рішення застосовне в різних

галузях промисловості. Воно може обробляти великі обсяги текстових даних і розгортатися локально, працювати за брандмауером або в приватній хмарі. Програмне забезпечення розбиває речення та фрази за допомогою NLP для оцінки семантики, синтаксису та контексту для виконання аналізу настроїв, категоризації, розпізнавання іменованих об'єктів, виділення теми, виявлення наміру та резюмування.

MeaningCloud – це інструмент для виявлення та аналітики тексту. Це просте у використанні та доступне рішення для отримання інформації з різного роду неструктурованого вмісту, такого як соціальні розмови, статті та документи. Продукти включають інтеграції та API. Наприклад, він має надбудови для Excel, Google Sheets, RapidMinder і Zapier, щоб у цих програмах можна було використовувати функції аналізу тексту. Він також має хмарні API, які розробники можуть використовувати для додавання можливостей семантичного аналізу до своїх програм через веб-сервіси, SDK і плагіни.

За результатами аналізу аналогів можна зробити висновок, що найбільш подібним є Lexalytics, оскільки є масштабованим та обробляти великі обсяги текстових даних. Також, Azure Text Analytics це, перш за все, NLP рішення.

Крім даних програм, слід згадати інші, окрім Azure, платформи та інструменти, які пропонують розробникам своє API для створення застосунків [8].

IBM Watson Discovery – це система пошуку та текстової аналітики на базі штучного інтелекту, яка використовує NLP для пошуку відповідей у вмісті та розкриття бізнес-анкетації з документів, веб-сторінок і великих даних. Watson Discovery використовує семантичний пошук для додавання контексту до відповідей шляхом перевірки вмісту підключених джерел даних, визначення найбільш релевантного фрагмента та надання вихідного документа або веб-сторінки. Він може знаходити приховані шаблони, тенденції та зв'язки між різними фрагментами вмісту. Програмне забезпечення також може допомогти створити розумніших чат-ботів.

Amazon Comprehend пропонує рішення для аналізу тексту за допомогою NLP і ML, щоб допомогти користувачам виявити ідеї та зв'язки в тексті. Він

використовує ML для перегляду електронних листів клієнтів, запитів у службу підтримки, оглядів продуктів, соціальних мереж і реклами для аналізу настроїв клієнтів. Сервіс може виділяти ключові фрази, місця, людей, бренди та події, аналізувати, позитивно чи негативно це використано, а також організовувати колекцію текстів за темами. Це може бути рішення для аналітики колл-центру, для індексування та пошуку відгуків про продукти, для персоналізації вмісту на веб-сайті або для обробки запитів у службу підтримки клієнтів.

Google Cloud Natural Language Natural Language допомагає користувачам отримувати інформацію з неструктурованого тексту за допомогою ML від Google. Це дозволяє користувачам отримувати інформацію про людей, місця та події. ML розкриває структуру та значення текстових даних, які допомагають зрозуміти настрої соціальних мереж і розмови клієнтів. Він включає спеціальні моделі ML для класифікації, вилучення та виявлення настроїв, API, які можна інтегрувати в програми, і аналіз у реальному часі неструктурованого медичного тексту.

Сильні й слабкі сторони конкурентів (таблиця 1.2):

- Lexalytics;
- MeaningCloud.

Таблиця 1.2 – Порівняння з аналогами

№	Властивість	Lexalytics	MeaningCloud
1	Ведення бази	-	+
2	API для NLP	-	+
3	Аналіз голосу	-	-
4	Аналіз тексту	+	+
5	Підтримка	-	+

2 РОЗРОБКА ПРОГРАМИ ДЛЯ АНАЛІЗУ ДАНИХ

Програмний застосунок «Rehabilife» дозволить користувачам реєструвати нових пацієнтів та записувати їх сеанси, аналізуючи емоційний підтекст. Застосунок дозволяє додавати в базу особисті дані пацієнтів, дату сеансу та додавати головний файл розмови для подальшого аналізу. Додаток захищений за допомогою системи авторизації [9].

Застосунок «Rehabilife» дозволяє користувачам:

- вести інформацію про кожного клієнта;
- записувати та завантажувати голосове відображення сеансу;
- з допомогою AI аналізувати емоційний стан пацієнта;
- з допомогою наївного баєсового класифікатора емоцій аналізувати текст та передбачати емоційний стан пацієнта.

Повний код проекту представлено у додатку А. Інструкція користувача, відповідно – в додатку Б.

2.1 Критерії системи

Вище згадувалося, чому важливо розбиратися в людських емоціях та вміти їх вивчати. Людям часто необхідно звертатися до кваліфікованого спеціаліста з ПТСР. Сьогодні в нашій країні це відчувається особливо гостро. Відповідно, попит на ці послуги зростає, пацієнтів стає все більше і більше. Отже, новітні технології інформаційних систем стають в пригоді. Вони допоможуть не тільки зберігати всю інформацію в зручному вигляді, а також обробляти емоції за допомогою AI, щоб підтвердити або спростувати.

Система повинна надавати користувачам усю необхідну інформацію щодо пацієнтів, аналізу їх емоційного стану та даним про сеанси.

Мета бізнесу – програмна система для аналізу психологічного та емоційного стану пацієнта. Користувач за допомогою застосунку зможе вести облік клієнтів, записувати їх сеанси (таблиця 2.1).

Мета створення системи (вимоги) – є створення системи, ціна обслуговування якої має бути дешевше за конкурентів та відповідати наступним критеріям [10].

Таблиця 2.1 – Критерії системи

Критерії	Узгодження
Можливість реєстрації пацієнтів	✓
Можливість корегування пацієнтів	✓
Можливість видалення пацієнтів	✓
Додати сторінку додавання пацієнта	✓
Додати сторінку перегляду пацієнта	✓
Додати можливість реєстрації за допомогою засобів Azure	✓
Додати голосових записів сеансів для аналізу	✓
Реалізувати наївний баєсів класифікатор емоцій	✓
Додати можливість збереження словника класифікатора	✓
Додати можливість збереження результатів класифікації	✓

Архітектура має бути модульною. Інтерфейс системи повинен бути дружелюбним до користувача, виконаним в мінімалістичному стилі та розроблений з урахуванням сучасних технологій та практик.

Життєвий цикл:

- розробка системи;
- тестування системи;
- випуск системи;

- просування;
- підтримка та подальший розвиток продукту.

Для медичних компаній важливим моментом є безпека даних. Оскільки підприємства домену Healthcare дуже піклуються про безпеку та доступ до даних, то вони повинні зберігатися в зашифрованому вигляді. Комунікації повинні бути захищеними та зашифрованими. Система повинна підтримувати авторизацію через Azure Active Directory. Потенційна аудиторія – це робітники великих медичних корпорацій, приватних клінік, центрів реабілітації, а також приватні психологи-реабілітологи, яким необхідно аналізувати емоційний стан [11].

2.2 Технології

Технології системи представлені на рисунку 2.1 нижче:

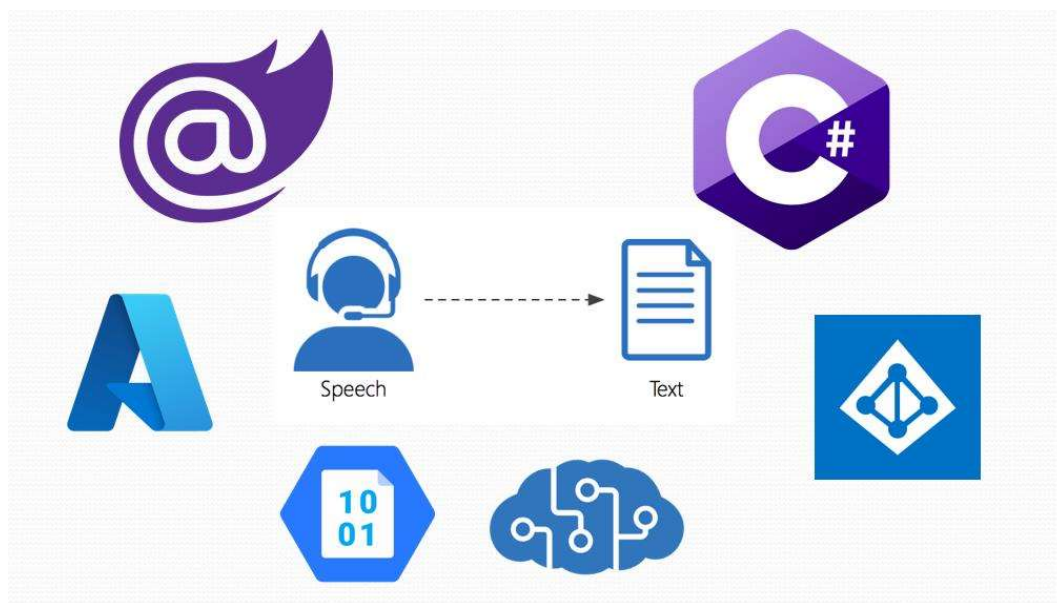


Рисунок 2.1 – Технології та інструменти (Рисунок виконаний самостійно)

Авторизація, автентифікація – Azure Active Directory.

Azure Active Directory – це хмарний сервіс керування доступом та ідентифікації користувачів, який дозволяє забезпечувати безпеку та захист веб-додатків, сервісів та ресурсів в обласному середовищі Microsoft Azure. Він дозволяє

керувати доступом до ресурсів, аутентифікувати користувачів за допомогою різних методів ідентифікації (наприклад, пароль, мультифакторна автентифікація, сертифікати), а також здійснювати централізоване управління правами доступу користувачів до різних сервісів та ресурсів.

Frontend:

- Blazor WASM;
- C#;
- JS;
- CSS;
- HTML.

Blazor WASM (WebAssembly) – це нова технологія розробки веб-додатків, яка дозволяє писати клієнтський код на C# та запускати його в браузері, використовуючи веб-стандарт WebAssembly. Blazor WASM дозволяє розробникам використовувати відомий мову програмування C# та бібліотеки .NET для розробки веб-додатків, забезпечуючи при цьому високу продуктивність та безпеку, а також багатий функціонал, який надається .NET. Крім того, Blazor WASM дозволяє зменшити кількість коду, що необхідний для розробки клієнтської частини веб-додатка, тому що можна використовувати існуючі бібліотеки .NET та C# код безпосередньо на клієнтській стороні.

Backend:

- .NET Core API;
- C#;
- Azure Speech;
- Azure Language Service;
- Azure Functions.

Azure Speech – це сервіс хмарного розпізнавання мови, який дозволяє перетворювати мовлення в текст, розпізнавати мову та голосові команди, а також синтезувати мовлення з тексту.

Azure Language Service – це сервіс хмарної обробки природньої мови, який дозволяє аналізувати та витягувати корисну інформацію з текстових даних, а також

здійснювати переклад тексту на різні мови.

Azure Functions – це хмарний сервіс обчислень, який дозволяє створювати та запускати функції відповідно до визначених подій чи розкладу, без необхідності розгортання та керування власним інфраструктурним середовищем. Він дозволяє написати код функції на одній з підтримуваних мов програмування, таких як C#, Java, Python, а потім запустити її в хмарі без необхідності керувати інфраструктурними питаннями, такими як масштабування, резервне копіювання та моніторинг. Azure Functions дозволяє реагувати на різноманітні події, наприклад, на запити веб-сервера, введення користувача, зміни в базі даних тощо.

Storage – Azure Blobs Storage.

2.3 Ціннісна пропозиція

Відносини зі споживачами, також відомі як відносини з клієнтами, – це процес, у якому команда створює та підтримує позитивні стосунки зі своїми споживачами. У двох словах, стосунки зі споживачами зосереджуються на спілкуванні з поточними та потенційними клієнтами, щоб отримати краще розуміння того, як організація може покращитися в багатьох сферах.

Взаємовідносини з клієнтами призначені для подальшого аналізу того, як організації можуть краще встановлювати та підтримувати позитивні стосунки з клієнтами. Команда організовує інформацію, щоб допомогти розвивати ці відносини. Споживачами системи є:

- великі медичні корпорації, які планують розширити свій перелік хмарних застосунків та надають послуги з реабілітації (наприклад, Unitedhealth Group, Express Scripts Holding Co., Bristol-Myers Squibb Co.);
- приватні медичні клініки;
- приватні спеціалісти;
- центри реабілітації.

Таким чином, споживачам представлено систему, щоб вирішити ці бізнес потреби.

Value Proposition Canvas – це структура, яка може пояснити, як продукт або послуга позиціонуються навколо того, що цінує та потребує клієнт [12]. З проблемою пошуку свого місця на ринку стикаються як стартапи, так і давно існуючі компанії. Щоб знайти його, вони повинні розбиратися, які завдання намагається вирішити клієнт за допомогою їхнього продукту чи послуги, які при цьому виникають складнощі, і що принесе покупцю найбільшу радість від придбання.

Value Proposition Canvas можна використовувати, коли потрібно вдосконалити наявну пропозицію продукту чи послуги або коли новий продукт розробляється з нуля. Щоб почати роботу з Value Proposition Canvas, завжди треба починати з клієнта. Звичайно, може бути багато різних сегментів клієнтів, яких компанія обслуговує (або хоче обслуговувати). Отже, перше завдання – обговорити, хто насправді клієнти на високому рівні, після чого можна прийняти деякі рішення щодо того, для кого застосунок розробляється.

Customer segment:

- gains – переваги, які клієнт потребує, або те, що принесе йому задоволення і те, що збільшить шанс прийняття ціннісної пропозиції;
- pains – негативний досвід, емоції та ризики, які зазнає клієнт у процесі виконання роботи;
- customer jobs – функціональні, соціальні та емоційні завдання, які клієнти намагаються виконати, проблеми, які вони намагаються вирішити, і потреби, які вони хочуть задовольнити.

Customer segment (профіль клієнта) має бути створений для кожного сегмента клієнтів, оскільки кожен сегмент має певну «роботу, яку потрібно виконати», труднощі та переваги.

Value proposition:

- gain creators – як продукт або послуга створює переваги для клієнтів і як вони пропонують додаткову цінність для клієнта;
- pain relievers – опис того, як саме продукт або послуга полегшує біль клієнта;

– products and services – продукти та послуги, які створюють вигоду та полегшують біль, і які лежать в основі створення цінності для клієнта.

Ціннісна пропозиція системи представлена на рисунку 2.2:

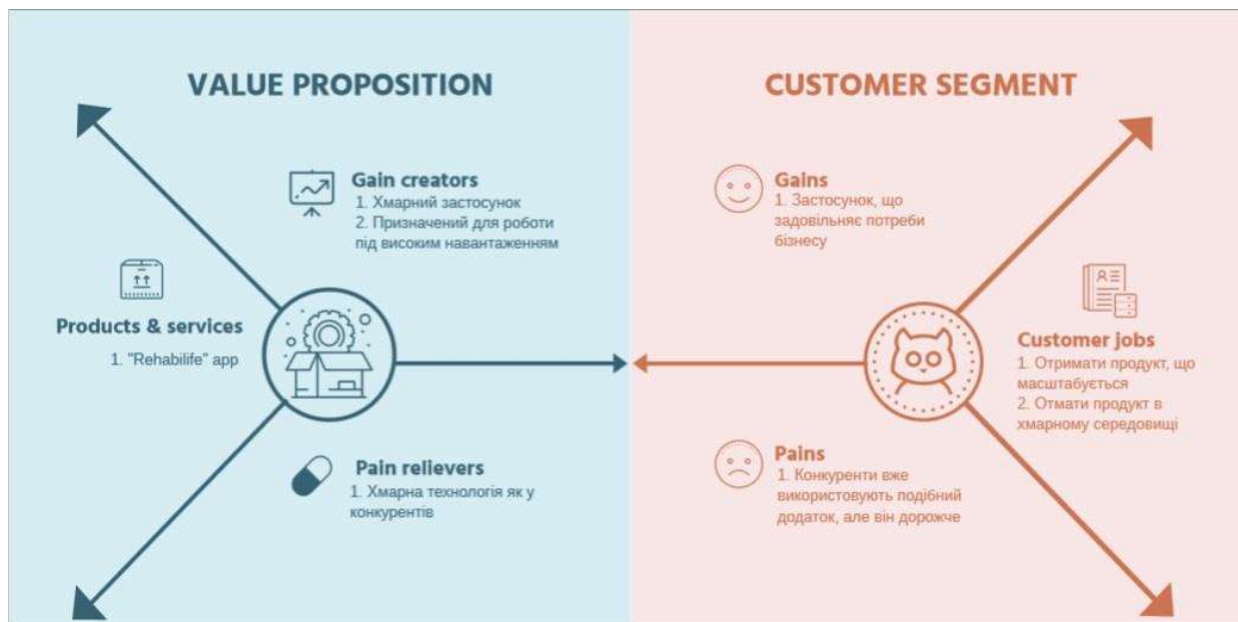


Рисунок 2.2 – Value Proposition Canvas (Рисунок виконаний самостійно)

Вище ми уже розбиралися, чому важливо розбиратися в людських емоціях та вміти їх вивчати. Людям часто необхідно звертатися до кваліфікованого спеціаліста. Відповідно, попит на цю професію зростає, пацієнтів стає все більше і більше.

Сучасні технології допомагають не тільки зберігати всю інформацію в зручному вигляді, а також обробляти дані за допомогою AI та моделей прогнозування.

Таким чином, споживачам представлено систему, щоб вирішити ці бізнес потреби.

2.4 Канвас бізнес-моделі

Canvas – це загальний алгоритм створення різних бізнес-моделей. Загальна

структура Canvas – це таблиця з дев'ятьма блоками [7]:

- сегменти споживачів;
- ключові цінності;
- канали збуту;
- відносини з клієнтами;
- потоки доходів;
- ключові ресурси;
- ключова діяльність;
- ключові партнери;
- структура витрат.

Для побудови цієї бізнес-моделі потрібно визначити майбутніх споживачів програмного продукту та виділити основні цінності, через які споживачі виберуть саме цей додаток. Також, потрібно визначити, за яким каналах збуту споживачі отримують цінності і як вибудувати взаємини з ними. Далі, з'ясувати основні потоки доходів, і які ресурси необхідні для здійснення проекту, визначити ключові дії, які необхідно зробити для реалізації даного проекту, а також виділити основних партнерів та постачальників.

Останній етап побудови бізнес-моделі Canvas – виявлення структури витрат, тобто найбільш важливих витрат при реалізації проекту «Rehabilife». На рисунку 2.3 показано бізнес-модель до програмного додатку «Rehabilife».

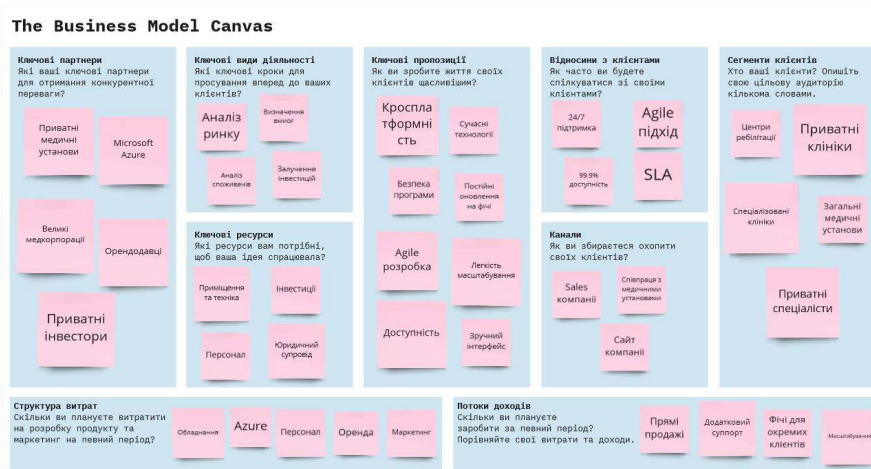


Рисунок 2.3 – Бізнес модель Canvas (Рисунок виконаний самостійно)

При побудові бізнес-моделі Canvas було враховано вищезазначені вимоги. Проведені розрахунки та дана модель дозволяють зробити висновки за інноваційною ідеєю.

2.5 Вихідні форми

Фронтенд частина застосунку буде створено на .NET 6 з використанням наступних NuGet-пакетів:

- Microsoft.NET.Sdk.BlazorWebAssembly;
- Microsoft.Extensions.Configuration.Json;
- Microsoft.Extensions.DependencyInjection;
- Microsoft.Extensions.Http.

Вимоги до платформи в цілому і необхідність розгортання оточення наведені нижче в таблиці 2.2:

Таблиця 2.2 – Вимоги до платформи і оточення

Назва	Критерії
Хмарне середовище	Microsoft Azure
ОС	Windows, Linux

Можливості нового користувача:

- реєстрація;
- автоматична авторизація;
- перегляд стартової сторінки.

Можливості клієнта:

- авторизація;
- перегляд стартової сторінки;
- додавання пацієнтів;
- видалення пацієнтів;
- редагування пацієнтів;

- перегляд інформації про пацієнтів;
- додавання голосових записів сеансів для аналізу.

За темою роботи побудовані основні діаграми визначення, представлення, проектування та моделювання програмних систем за допомогою мови UML.

Use Case – це методологія, яка використовується в системному аналізі для визначення, уточнення та організації системних вимог. Use Case описує сценарій взаємодії учасників (як правило, користувача та системи). Учасників може бути двоє та більше. Користувачем може бути як людина, і інша система.

Метою Use Case діаграми в UML є демонстрація різних способів взаємодії користувача з системою.

На рисунку 2.4 представлено Use Case діаграму для застосунку «Rehabilife»:

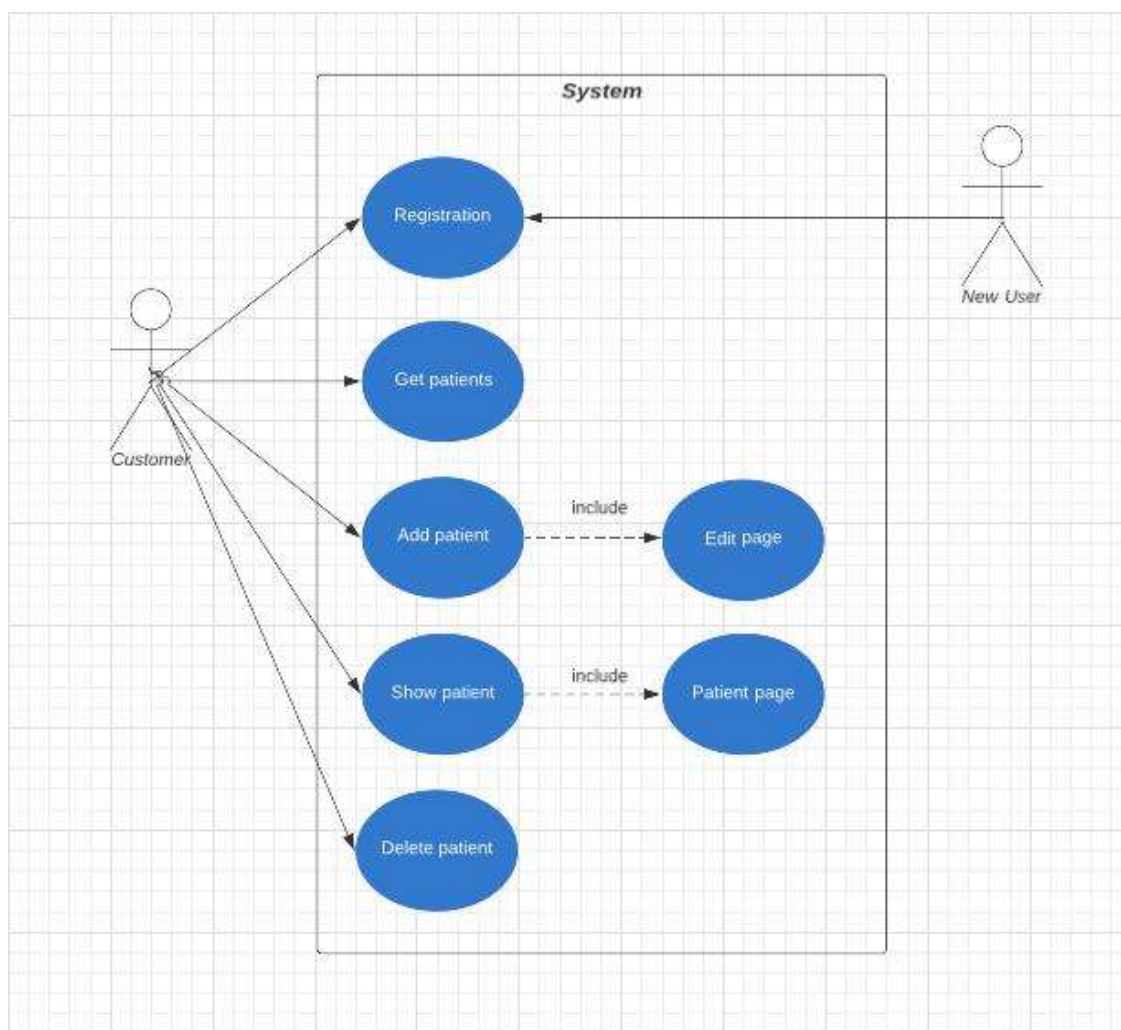


Рисунок 2.4 – Use Case діаграма (Рисунок виконаний самостійно)

Sequence діаграма – це різновид діаграми взаємодії, оскільки вона описує, як і в якому порядку група об'єктів працює разом. Ці діаграми використовуються розробниками програмного забезпечення та бізнес-професіоналами, щоб зрозуміти вимоги до нової системи або задокументувати існуючий процес. Простіше кажучи, діаграма послідовності показує, як різні частини системи працюють у послідовності, щоб щось зробити.

Sequence діаграма структурована таким чином, що вона представляє часову шкалу, яка починається вгорі та поступово спускається вниз, щоб позначити послідовність взаємодій. Кожен об'єкт має стовпець, а повідомлення, якими вони обмінюються, позначаються стрілками.

На рисунку 2.5 представлено Sequence діаграму для застосунку, що розробляється:

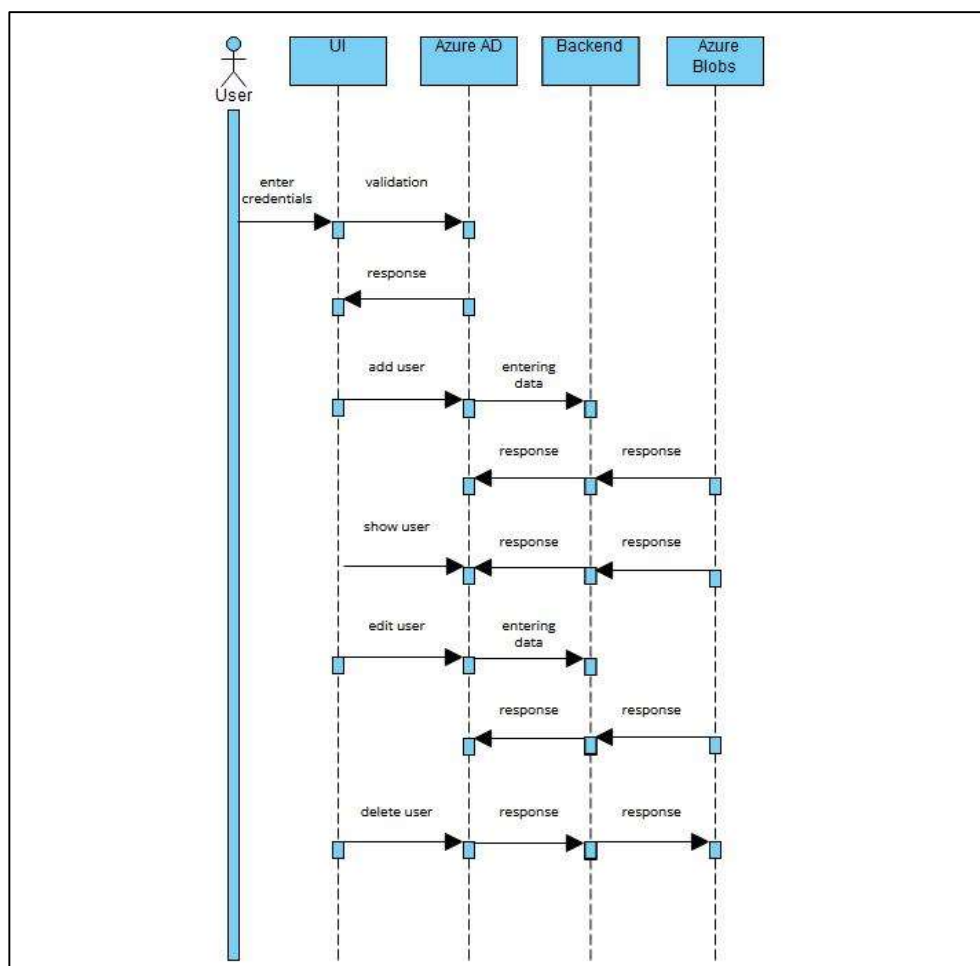


Рисунок 2.5 – Sequence діаграма (Рисунок виконаний самостійно)

Як бачимо, дана діаграма дозволяє стисло та якісно відобразити динамічну поведінку застосунку та виокремити наступні можливості «Rehabilife»:

- додавання користувачів;
- редагування користувачів;
- видалення користувачів;
- а також відобразити, які події відбуваються під час сеансу з пацієнтом, що додається, наприклад, до Azure Blobs Storage.

2.6 Робочий процес застосунку

Застосунок буде також записувати сеанси з пацієнтами, та зберігати їх в Azure Blobs Storage. Після цього записи мають бути зконвертовані у текст з допомогою Azure Speech To Text, а потім проаналізовані з допомогою Azure Text Analytics (рисунок 2.6).



Рисунок 2.6 – Workflow «Rehabilife» (Рисунок виконаний самостійно)

Застосування має включати в себе:

- розпізнавання мови;
- синтаксичний розбір;
- виділення основних змістових конструкцій;
- емоційний аналіз тексту.

Додаємо в застосунок серверний проект на .NET 6 з використанням наступних NuGet-пакетів:

- Azure.AI.TextAnalytics;
- Azure.Storage.Blobs;
- Microsoft.CognitiveServices.Speech;
- Newtonsoft.Json;
- Microsoft.Extensions.DependencyInjection;
- Microsoft.Extensions.Hosting.

Спочатку створимо subscription [13], потім ресурс Speech (рисунок 2.7):

Create Speech Services ...

Basics Network Identity Tags Review + create

Transcribe audible speech into readable, searchable text. Add real-time speech translations to your apps and services. Convert text to audio nearly in real time. Quickly build speech-enabled apps and services using the programming languages you already work with. Customize speech systems to optimize quality for specific scenarios.

[Learn more](#)

Project Details

Subscription * ⓘ Подписка Visual Studio Professional

Resource group * ⓘ DefaultResourceGroup-WEU
[Create new](#)

Instance Details

Region ⓘ West Europe

Name * ⓘ speech-app

Pricing tier * ⓘ Free F0
[View full pricing details](#)

Warning: Cognitive services resource creation requires subscription registration, we detected that your selected subscription did not register cognitive services resource type before, we will help you to register cognitive services resource type when you select a subscription in subscription dropdown. Click to learn more how to check registration state for your selected subscription.

[Review + create](#) < Previous Next: Network >

Рисунок 2.7 – Speech Resource (Рисунок виконаний самостійно)

Обираємо free pricing tier (F0) та деплой в Western Europe. Після того, як Azure створить необхідну ресурс групу, переходимо до неї на порталі. Копіюємо ключі та регіон, ці дані знадобляться для створення застосунку.

Код програми буде виглядати наступним чином:

- IConfigurationRoot? configuration – конфігурація з ключами;
- ISpeechToTextConverter – сервіс для перетворення записів у текст;
- ISpeechRecognitionService – сервіс для роботи з розпізнаним текстом;
- ITextAnalyticsService – сервіс для аналізу емоцій в розпізаному тексті;
- Решта сервісів та конфігів – зі Speech SDK.

```

1.         builder.Services.AddAzureClients(fb =>
2.             {
3.                 fb.AddBlobServiceClient(builder.Configuration[BlobSettings
4. PatientsStorageAccount]).WithName(BlobSettings.PatientsStorageName)
5.                 .ConfigureOptions(options =>
6.                     {
7.                         options.Retry.Mode =
8. Azure.Core.RetryMode.Exponential;
9.                         options.Retry.MaxRetries =
10. BlobSettings.MaxRetries;
11.                         options.Retry.MaxDelay =
12. BlobSettings.MaxDelayBetweenRetries;
13.                     });
14.             });
15.         builder.Services
16.             .AddSingleton(_ =>
17.             {
18.                 SpeechConfig? speechConfig =
19. SpeechConfig.FromSubscription(builder.Configuration[SpeechSettings.Subscrip
20. tionKey], builder.Configuration[SpeechSettings.ServiceRegion]);
21.                 speechConfig.SpeechRecognitionLanguage = "en-US";
22.                 return speechConfig;
23.             });
24.         builder.Services
25.             .AddSingleton(_ => new TextAnalyticsClient(new
26. Uri(builder.Configuration[TextAnalyticsSettings.ApiUri]), new
27. AzureKeyCredential(builder.Configuration[TextAnalyticsSettings.ApiKey]));
28.         builder.Services
29.             .AddTransient<IPatientsBlobService,
30. PatientsBlobService>()
31.             .AddTransient<ISpeechToTextConverter,
32. SpeechToTextConverter>()

```

```

27.         .AddTransient<ISpeechRecognitionService,
SpeechRecognitionService>()
28.         .AddTransient<ITextAnalyticsService,
TextAnalyticsService>();

```

2.7 Модель прогнозування

Серед задач наївного байєсового класифікатора є наступні:

- розпізнавання спаму;
- аналіз емоційного забарвлення текстів;
- виявлення расизму у текстовій виборці;
- інші.

Робота даного класифікатору базується на припущенні: «Кожен параметр даних, що класифікуються, розглядається незалежно від інших параметрів класу». У випадках, коли належність спостереження до певного класу має різну ймовірність, класифікатор може повернути вектор, що містить компоненти-ймовірності приналежності до кожного з класів.

Математичне представлення Теорема Байєса (2.1):

$$P(A | B) = \frac{P(B | A) P(A)}{P(B)} \quad (2.1)$$

де A і B є подіями;

$P(A)$ є апіорною ймовірністю гіпотези A ;

$P(B)$ є повною вірогідністю настання події B ;

$P(A | B)$ є ймовірністю гіпотези A за умови настання події B ;

$P(B | A)$ є ймовірністю настання події B за умови істинності гіпотези A .

Для того, щоб провести класифікацію даних, необхідно мати словник слів та відповідні ймовірності їх належності до певних класів. Для кожного класу потрібно обчислити загальну ймовірність належності повідомлення до цього класу. Тут необхідна контекстна модель, з якої створюється спочатку ненормований, а потім нормований словники.

Завантажуємо емулятор Cosmos DB з сайту Microsoft. Дана база була обрана для роботи, оскільки є простою у використанні, та добре підходить для зберігання моделей датасетів з даної роботи.

Додаємо в застосунок ще один серверний проект на .NET 6 з використанням наступних NuGet-пакетів:

- AutoMapper;
- GemBox.Spreadsheet;
- Microsoft.Azure.Cosmos;
- Microsoft.Extensions.Configuration.Json;
- Microsoft.Extensions.DependencyInjection;
- Microsoft.Extensions.Hosting;
- Microsoft.Extensions.Options.ConfigurationExtensions;
- HtmlAgilityPack.

Запускаємо емулятор бази, копіюємо Connection String (рисунок 2.8):

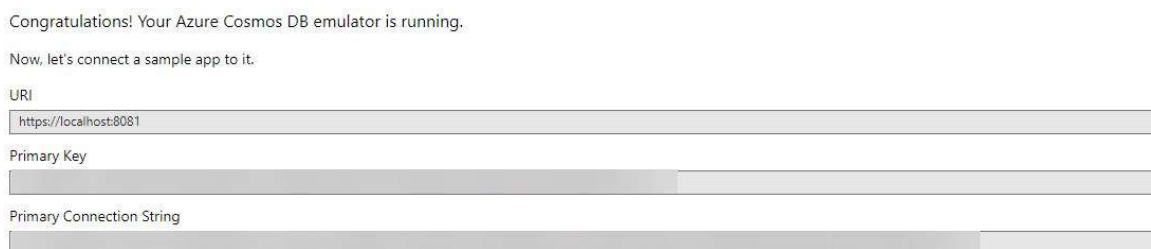


Рисунок 2.8 – Підключаємося до бази даних (Рисунок виконаний самостійно)

Для зберігання результату використаємо GemBox.Spreadsheet, який дозволяє генерувати xlsx-файли розміром до 150 кліток безоплатно навіть в комерційних проектах (згідно умов ліцензії). Цього вистачить для цілей дослідження. Для ініціалізації використаємо ключ «FREE-LIMITED-KEY».

Спочатку створюємо з'єднання з базою, реєструючи Cosmos client та репозиторій для роботи з базою в DI контейнер. Потім ініціалізуємо сторонюю бібліотеку для роботи з Excel GemBox.Spreadsheet та інжектимо сервіси в DI-контейнер:

```

1.     private static IHostBuilder CreateHostBuilder()
2.     {
3.         return Host.CreateDefaultBuilder()
4.             .UseConsoleLifetime()
5.             .ConfigureHostConfiguration(builder =>
6.                 {
7.                     const string configRelativePath =
"Config/appsettings.json";
8.
9.                     string configPath =
Path.Combine(AppContext.BaseDirectory, configRelativePath);
10.                    builder
11.                        .AddJsonFile(configPath);
12.                })
13.                .ConfigureServices((context, services) =>
14.                    {
15.                        SpreadsheetInfo.SetLicense(context.Configuration[ConfigurationConstants.Gem
BoxSpreadsheetKey] ?? string.Empty);
16.
17.                        services.AddAutoMapper(me =>
18.                            me.CreateMap<MessageInfoContext,
MessageDbModel>()
19.                                .ReverseMap());
20.
21.                        services.AddHostedService<HostedService>();
22.
23.                        services.AddSingleton(_ =>
24.                            {
25.                                CosmosClient cosmosClient =
new(context.Configuration[ConfigurationConstants.DbConnectionString]);
26.                                return cosmosClient;
27.                            });
28.
29.                        services
30.                            .AddSingleton<IMessagesRepository,
MessagesRepository>()
31.                            .AddScoped<IDataSetInfoProvider,
DataSetInfoProvider>()
32.                            .AddScoped<IMessageInfoContextParser,
MessageInfoContextParser>()
33.                            .AddScoped<IMessagesUploader,
MessagesUploader>()
34.                            .AddTransient<IStandardizedDictionaryBuilder,
StandardizedDictionaryBuilder>()
35.                            .AddTransient<INaiveBayesClassifier,
NaiveBayesClassifier>();
36.                    })
37.                    .ConfigureLogging( (_, logging) =>
38.                        {
39.                            logging.ClearProviders();
40.                            logging.AddSimpleConsole(options =>
options.IncludeScopes = true);
41.                        });
42.                }
43.

```

```

        IServiceCollection services = new ServiceCollection();
        IConfigurationBuilder builder = new ConfigurationBuilder();

        string configPath = Path.Combine(AppContext.BaseDirectory,
ConfigRelativePath);
        IConfigurationRoot configuration = builder
            .AddJsonFile(configPath)
            .Build();

        services
            .AddSingleton(_ =>
            {
                return new(configuration[DbConnectionString]);
            });

```

Тепер створюємо репозиторій:

```

1.     internal abstract class BaseCosmosDbRepository<TModel> :
IBaseCosmosDbRepository<TModel>
2.         where TModel : BaseDbModel
3.     {
4.         protected Container Container { get; }
5.
6.         protected ItemRequestOptions WriteItemRequestOptions { get; }
7.
8.         protected BaseCosmosDbRepository(
9.             CosmosClient cosmosClient,
10.            string databaseName,
11.            string containerName)
12.        {
13.            Container = cosmosClient.GetContainer(databaseName,
containerName);
14.
15.            WriteItemRequestOptions = new ItemRequestOptions
16.            {
17.                EnableContentResponseOnWrite = false
18.            };
19.        }
20.
21.        public async Task<ItemResponse<TModel>> AddItemAsync(TModel
item, string partition)
22.        {
23.            return await Container.CreateItemAsync(item,
24.                partitionKey: new PartitionKey(partition),
25.                requestOptions: WriteItemRequestOptions);
26.        }
27.
28.        public async Task<ItemResponse<TModel>> DeleteItemAsync(string
id, string partition)
29.        {
30.            return await Container.DeleteItemAsync<TModel>(id, new
PartitionKey(partition), requestOptions: WriteItemRequestOptions);
31.        }
32.
33.        public FeedIterator<TModel> GetItemsAsync(

```

```

34.         Expression<Func<TModel, bool>> predicate = null,
35.         string partition = null)
36.     {
37.         QueryRequestOptions queryRequestOptions = new()
38.         {
39.             PartitionKey = new PartitionKey(partition,
40.         };
41.
42.         IQueryable<TModel> queryable =
Container.GetItemLinqQueryable<TModel>(requestOptions:
queryRequestOptions);
43.
44.         if (predicate != null)
45.         {
46.             queryable = queryable.Where(predicate);
47.         }
48.
49.         return queryable.ToFeedIterator();
50.     }
51. }

```

Тепер необхідно додати в проект ініціалізацію бази даних:

```

1.     private async Task InitializeDbAsync()
2.     {
3.         DatabaseResponse databaseResponse = await
_cosmosClient.CreateDatabaseIfNotExistsAsync(DatabaseNames.NaiveBayesLab);
4.         await
databaseResponse.Database.CreateContainerIfNotExistsAsync(ContainerNames.Me
ssages, PartitionKeyPath);
5.     }

```

Перевіряємо, що база та контейнер створені в емуляторі (рисунок 2.9):

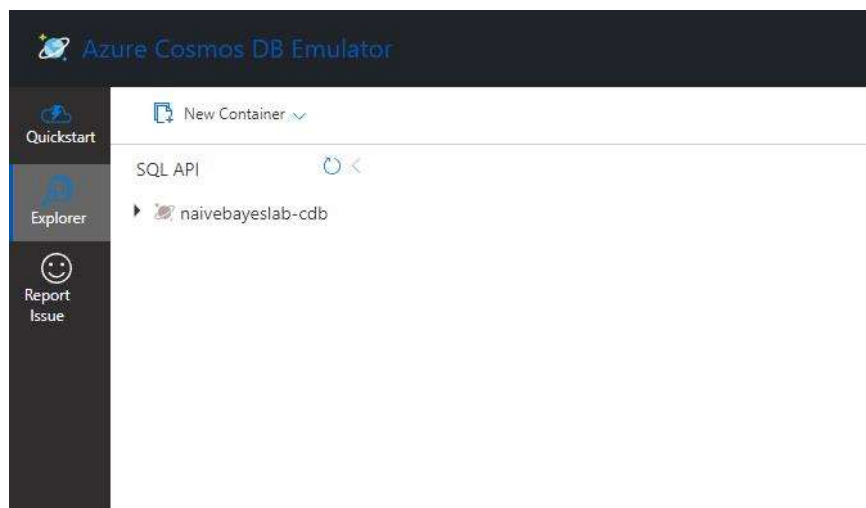


Рисунок 2.9 – База в емуляторі (Рисунок виконаний самостійно)

Далі необхідно спарсити навчальний датасет. Для цього імпортуємо навчальний датасет з Excel в HTML та парсимо з допомогою HtmlAgilityPack, а потім зберігаємо до бази даних. Цим буде займатися окремий сервіс MessageInfoContextParser:

```

1.     public class MessageInfoContextParser : IMessageInfoContextParser
2.     {
3.         public ICollection<MessageInfoContext> Parse(string
dataSetRelativePath)
4.         {
5.             string dataSet = File.ReadAllText(dataSetRelativePath);
6.
7.             HtmlDocument doc = new();
8.             doc.LoadHtml(dataSet);
9.
10.            IEnumerable<Dictionary<string, string>> rows =
11.                from table in doc.DocumentNode.SelectNodes("//table")
12.                from tableBody in table.SelectNodes("tbody")
13.                from row in tableBody.SelectNodes("tr")
14.                select row
15.                    .SelectNodes("td")
16.                    .Select(n => new
17.                    {
18.                        Id = n.Attributes["class"].Value,
19.                        Text = n.InnerText
20.                    })
21.                    .ToDictionary(n => n.Id, n => n.Text);
22.
23.            return rows
24.                .Select(d => new MessageInfoContext(
25.                    d.ElementAt(0).Value,
26.                    d.ElementAt(1).Value,
27.                    d.ElementAt(2).Value
28.                ))
29.                .ToArray();
30.        }
31.    }

```

Проводимо пробний сеанс у програмі та в результаті отримуємо контекстну модель.

Ймовірності нормуються за наступною формулою 2.2:

$$P_{\text{слова_нормована}} = \frac{n_{\text{повторень_слова}} \cdot P_{\text{слова_ненормована}} + P_{\text{класу}}}{n_{\text{повторень_слова}} + 1}, \quad (2.2)$$

Використовуючи набуту інформацію, створюємо у програмі сервіс наступного вигляду:

```

1.     internal class StandardizedDictionaryBuilder :
IStandardizedDictionaryBuilder
2.     {
3.         public StandardizedDictionaryContext Build(
4.             ICollection<MessageInfoContext> contexts)
5.         {
6.             string[] labels = contexts
7.                 .Select(i => i.Label)
8.                 .Distinct()
9.                 .ToArray();
10.
11.             double labelProbability =
CalculateLabelProbability(labels.Length);
12.
13.             Dictionary<string, Dictionary<string, (int count, double
probability)>> nonStandardizedDictionary =
14.                 BuildNonStandardizedDictionary(contexts, labels);
15.
16.             Dictionary<string, IReadOnlyDictionary<string, double>>
standardizedDictionary = nonStandardizedDictionary
17.                 .Select(d =>
18.                 {
19.                     Dictionary<string, double>
standardizedCategoryDictionary = new();
20.
21.                     foreach (KeyValuePair<string, (int count, double
probability)> kvp in d.Value)
22.                     {
23.                         (int count, double probability) = kvp.Value;
24.
25.                         standardizedCategoryDictionary.Add(kvp.Key,
CalculateStandardizedProbability(count, probability, labelProbability));
26.                     }
27.
28.                     return new
29.                     {
30.                         Label = d.Key,
31.                         StandardizedDictionary =
(IReadOnlyDictionary<string, double>)new ReadOnlyDictionary<string,
double>(standardizedCategoryDictionary)
32.                     };
33.                 })
34.                 .ToDictionary(d => d.Label, d =>
d.StandardizedDictionary);
35.
36.             ReadOnlyDictionary<string, IReadOnlyDictionary<string,
double>> result = new(standardizedDictionary);
37.
38.             return new StandardizedDictionaryContext(result,
labelProbability);
39.         }

```

```

40.
41.         private static Dictionary<string, Dictionary<string, (int
count, double probability)>> BuildNonStandardizedDictionary(
42.             ICollection<MessageInfoContext> contexts,
43.             IEnumerable<string> labels)
44.         {
45.             IEnumerable<string> allMessages = contexts.Select(i =>
i.Text);
46.             IEnumerable<string[]> allTokens = allMessages.GetTokens();
47.
48.             string[] categories = contexts
49.                 .Select(i => i.Category)
50.                 .Distinct()
51.                 .ToArray();
52.
53.             ICollection<(string category, int totalCount)>
categoryContexts = categories
54.                 .Select(c =>
55.                 {
56.                     int totalCount = allTokens
57.                         .Count(t => t.Contains(c));
58.
59.                     return (c, totalCount);
60.                 })
61.                 .ToArray();
62.
63.             Dictionary<string, Dictionary<string, (int count, double
probability)>> nonStandardizedDictionary = new();
64.             foreach (string label in labels)
65.             {
66.                 IEnumerable<string> labelMessages = contexts
67.                     .Where(c => c.Label == label)
68.                     .Select(i => i.Text);
69.                 IEnumerable<string[]> labelTokens =
labelMessages.GetTokens();
70.
71.                 Dictionary<string, (int count, double probability)>
nonStandardizedCategoryDictionary = categoryContexts
72.                     .Select(ctx =>
73.                     {
74.                         int totalCount = ctx.totalCount;
75.                         int labelCount = labelTokens
76.                             .Count(t => t.Contains(ctx.category));
77.
78.                         double categoryProbability =
(double)labelCount / totalCount;
79.
80.                         return new
81.                         {
82.                             Category = ctx.category,
83.                             Count = totalCount,
84.                             Probability =
Math.Round(categoryProbability, 2)
85.                         };
86.                     })
87.                 .ToDictionary(c => c.Category, c => (c.Count,
c.Probability));
88.

```

```

89.             nonStandardizedDictionary.Add(label,
nonStandardizedCategoryDictionary);
90.             }
91.
92.             return nonStandardizedDictionary;
93.         }
94.
95.     private static double CalculateLabelProbability(int
labelCount)
96.     {
97.         double probability = (double)1 / labelCount;
98.
99.         return Math.Round(probability, 2);
100.    }
101.
102.    private static double CalculateStandardizedProbability(
103.        int count,
104.        double nonStandardizedProbability,
105.        double labelProbability)
106.    {
107.        double probability = (count * nonStandardizedProbability)
+ labelProbability;
109.
110.        return Math.Round(probability / (count + 1), 2);
111.    }
112. }

```

Результатом стане нормований словник ймовірностей класів. Таку модель будемо використовувати для ранів класифікатора.

Отже, маючи навчальну вибірку даних та словник з нормованими класами, можемо виконати наступний алгоритм. За баєсів класифікаційний аналіз в програмі буде відповідати наступний сервіс:

```

1.     internal class NaiveBayesClassifier : INaiveBayesClassifier
2.     {
3.         private readonly ILogger<NaiveBayesClassifier> _logger;
4.
5.         public NaiveBayesClassifier(ILogger<NaiveBayesClassifier>
logger)
6.         {
7.             _logger = logger;
8.         }
9.
10.        public ClassifiedContext Classify(
11.            ICollection<MessageInfoContext> contexts,
12.            StandardizedDictionaryContext standardizedDictionary)
13.        {
14.            List<string> loggingList = new();
15.            List<bool> resultList = new();
16.            int contextsCount = contexts.Count;
17.

```

```

18.         for (int i = 0; i < contextsCount; i++)
19.         {
20.             MessageInfoContext context = contexts.ElementAt(i);
21.
22.             ICollection<string> tokens = context.Text
23.                 .GetTokens()
24.                 .ToArray();
25.
26.             List<(string label, double probability)>
labelProbabilities = new();
27.             foreach (string label in
standardizedDictionary.StandardizedDictionary.Keys)
28.                 {
29.
standardizedDictionary.StandardizedDictionary.TryGetValue(label, out
IReadOnlyDictionary<string, double> standardizedCategories);
30.                 List<double> probabilityValues = new();
31.
32.                 if (standardizedCategories == null)
33.                 {
34.                     continue;
35.                 }
36.
37.                 foreach (KeyValuePair<string, double> category in
standardizedCategories)
38.                 {
39.                     if (tokens.Contains(category.Key))
40.                     {
41.                         probabilityValues.Add(category.Value);
42.                     }
43.                 }
44.
45.                 double labelProbability = 0.01;
46.                 if (probabilityValues.Any())
47.                 {
48.                     double multipliedProbability =
probabilityValues
49.                         .Aggregate((x, y) => x * y);
50.                     labelProbability =
standardizedDictionary.LabelProbability * multipliedProbability;
51.                 }
52.
53.                 labelProbabilities
54.                     .Add((label, labelProbability));
55.             }
56.
57.             (string result, double _) = labelProbabilities.MaxBy(l
=> l.probability);
58.
59.             loggingList.Add($"{i + 1}. Message: '{context.Text}'
classified as {result}.\n-> It is {result == context.Label}");
60.
61.             resultList.Add(result == context.Label);
62.         }
63.
64.         _logger.LogDebug(string.Join("\n", loggingList.ToArray()));
65.
66.         int correctResults = resultList.Count(r => r);

```

```
67.         double correctPercentage = (double)correctResults /
contextsCount;
68.
69.         return new ClassifiedContext(
70.             contextsCount,
71.             Math.Round(correctPercentage * 100, 2));
72.     }
73. }
```

Сервіс працює з навчальними даними. Як бачимо, це лише контекст, необхідний для словника, нічого зайвого (витягуємо з бази за допомогою AutoMapper). Результатом є нормований словник ймовірностей класів. Таку модель будемо використовувати для запусків класифікатора уже з готовим набором даних.

ВИСНОВКИ

У ході магістерського дослідження було вирішено задачу аналізу психологічного стану та надання спеціалістам з емоційної реабілітації додаткової підтримки в роботі з пацієнтами в галузі охорони здоров'я. Для цього було розроблено хмарний кросплатформний додаток «Rehabilife», який використовує модель прогнозування на основі наївного баєсівого класифікатору.

У додатку також використано сервіси штучного інтелекту Azure Speech та Azure Language Service, які дозволяють виконувати текстовий аналіз та розпізнавання мови. Також для зменшення необхідності в розгортанні та керуванні власною інфраструктурою використано хмарні технології, зокрема Azure Functions. У процесі дослідження було проведено аналіз та змодельовано аналіз конкретного датасету наївного баєсівого класифікатору, створено на його основі програму класифікації та випробувано її на навчальній вибірці. Результатом виконання магістерської роботи став розроблений програмний додаток «Rehabilife», написаний на мові програмування C#.

Програмний продукт має потенціал для подальшого вдосконалення, зокрема використання нейронних мереж для задач класифікації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bryant RA, McFarlane AC, Silove D, O'Donnell ML, Forbes D, Creamer M. The lingering impact of resolved PTSD on subsequent functioning. [Електронний ресурс] / URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6681635> (дата звернення: 30.04.2023).
2. Betancourt TS, Thomson DL, Brennan RT, Antonaccio CM, Gilman SE, VanderWeele TJ. Social acceptance helps mental health after war trauma [Електронний ресурс] / URL: <https://www.nih.gov/news-events/nih-research-matters/social-acceptance-helps-mental-health-after-war-trauma> (дата звернення: 30.04.2023).
3. Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. [Електронний ресурс] / URL: <https://www.sciencedirect.com/science/article/pii/S2001037014000464?via%3Dihub> (дата звернення: 30.04.2023).
4. Webb, G. I.; Boughton, J.; Wang, Z. Not So Naive Bayes: Aggregating One-Dependence Estimators. Machine Learning. 58 [Електронний ресурс] / 24.05.2005. URL: <https://doi.org/10.1007%2Fs10994-005-4258-6> (дата звернення: 30.04.2023).
5. M.A. Yerkinbekova, R.S. Kasumova, O.Fedorovich. A Study of Emotional States Students. International Journal of Humanities Social Sciences and Education (IJHSSE) [Текст] / ISSN 2349-0373 (Print) & ISSN 2349-0381 (Online). Volume 1, Issue 2 February 2014, PP 41-47.
6. Docs / Azure / Cognitive Services / Speech. What is the Speech service? [Електронний ресурс] / 12.05.2022. URL: <https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview> (дата звернення: 30.04.2023).
7. Docs / Azure / Cognitive Services / Cognitive Service for Language. What is Azure Cognitive Service for Language? [Електронний ресурс] / 22.07.2022. URL: <https://docs.microsoft.com/en-us/azure/cognitive-services/language-service/overview> (дата звернення: 30.04.2023).

8. JOSE MARIA DELOS SANTOS. Best Text Analysis Software [Электронный ресурс] / URL: <https://project-management.com/best-text-analysis-software/> (дата звернення: 30.04.2023).

12. Dr Timothy Mansfield. Value Proposition Canvas explained: how to match your services to customer needs [Электронный ресурс] / URL: <https://interaction.net.au/articles/value-proposition-canvas-explained> (дата звернення: 30.04.2023).

13. Build in the cloud with an Azure free account. Create, deploy, and manage applications across multiple clouds, on-premises, and at the edge [Электронный ресурс] / URL: <https://azure.microsoft.com/en-us/free> (дата звернення: 30.04.2023).

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ
КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

9. Alkilani, M., Kobziev, V. Enhancing E-government Services by Using Cloud Computing / CEUR Workshop Proceedings, 2019, 2683, pp. 66-69.

10. Khovrat A., Kobziev V., Nazarov A., Yakovlev S. Parallelization of the VAR Algorithm Family to Increase the Efficiency of Forecasting Market Indicators During Social Disaster. CEUR Workshop Proceedings, 2022, 3347, pp. 222–233.

11. Gordiievych, A. , Shubin, I. Forecasting of Airfare Prices Using Time Series // Information Technologies in Innovation Business Conference, ITIB 2015 - Proceedings, 2015, с. 68-71, 7355055.