

**Розвідковий аналіз табличних даних засобами великих мовних моделей
(Exploratory analysis of tabled data by means of LLMs)**

Конкурсна робота

Шифр роботи: “Розвідкові моделі”

АНОТАЦІЯ

В роботі наведено огляд та аналіз джерел літератури за темою наукової роботи, визначені об'єкт і предмет наукового дослідження та вказано перелік завдань для дослідження.

Для розуміння тематичної області та завдань для дослідження, в роботі описано базові поняття та деякі найпоширеніші алгоритми аналізу даних. Серед описаних технологій виділено штучний інтелект, глибинне навчання та великі мовні моделі.

Основна увага в проведених дослідженнях зосереджена на можливість великих мовних моделей виконувати арифметичні чи статистичні операції у просторі “смислу” та розроблення засобів для проведення таких досліджень. Засоби дослідження представлені у вигляді чат-бот системи в середовищі Telegram. Для оцінки запропонованої концепції було розглянуті прості випадки аби краще зрозуміти ефективність великих мовних моделей як одного із інструментів розвідкового аналізу даних.

Практичні результати дослідження та деякі екранні форми тестування запропонованої концепції вказують на те, що концепція має деякі переваги над найближчими аналогами, а також виявлені наукові проблеми, які можуть бути вирішеними в наступних дослідженнях.

Ключові слова: розвідковий аналіз даних, інженерія підказок, великі мовні моделі, чат-бот системи, простір “смислу”

ABSTRACT

This article provides an overview and analysis of references on the topic of the scientific paper, defines the object and subject of scientific research, and specifies the list of research tasks.

In order to understand thematic area and research tasks, the paper describes the basic concepts and the most common data analysis algorithms. Among the mentioned technologies, artificial intelligence, deep learning, and large language models are highlighted.

The main focus of the research is centered around the ability of LLM to perform arithmetic and statistical operations in the meaning space, additionally, around the development of tools designated for conducting such research. Research tools are presented as chatbot system in the Telegram environment. To evaluate the proposed concept, simple cases were considered in order to better understand the effectiveness of LLM as one of the tools for exploratory data analysis.

The practical result of the research and some forms of its screen testing indicate that the concept has some advantages over the closest analogues, as well as identify several scientific problems that can be solved in subsequent studies.

Keywords: exploratory data analysis, prompt engineering, large language models, chatbot systems, “meaning” space

СПИСОК СКОРОЧЕНЬ

ВММ - великі мовні моделі

ГН - глибинне навчання

РАД - розвідковий аналіз даних

МН - машинне навчання

ОПМ - обробка природньої мови

ШІ - штучний інтелект

ШНМ - штучна нейронна мережа

ЗМІСТ

ВСТУП	6
1. ВИБРАНІ ТЕХНОЛОГІЇ ТА ЗАСОБИ ДЛЯ АНАЛІЗУ ДАНИХ	9
1.1. Вступ до аналізу даних	9
1.2. Наукові основи аналізу даних	10
1.3. Процес аналізу даних	10
1.4. Методи та технології для аналізу даних	12
1.4.1. Кластерний аналіз	12
1.4.2. Багатопросторове масштабування	13
1.4.3. Дерева рішень	13
1.4.4. Видобуток даних	13
1.4.5. Машинне навчання	14
1.4.6. Штучні нейронні мережі	15
1.4.7. Простір “смислу” та великі мовні моделі	16
1.5. Використання чат-бот систем	18
2. ЗАСОБИ АНАЛІЗУ ДАНИХ У ПРОСТОРІ “СМИСЛУ”	20
2.1. Дослідження ВММ в задачах аналізу даних	20
2.2. Виконання операцій у просторі “смислу”	25
2.3. Архітектура чат-бот системи	26
3. ЧАТ-БОТ СИСТЕМА ДЛЯ АНАЛІЗУ ТАБЛИЧНИХ ДАНИХ	28
3.1. Створення інтерфейсу в системі Telegram	28
3.2. Організація зв’язку з ВММ	30
3.3. Робота з базою даних	32
3.4. Візуалізація даних	35
ВИСНОВКИ	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39

ВСТУП

Великі дані та прогресивні технології в області аналізу даних приносять як нові можливості, так і нові виклики для інженерів й дослідників даних. [1] Провідні світові фірми накопичили величезні об'єми даних і цей процес продовжується, адже бізнес і наука бачать в цьому велику вигоду, особливо в аналізі даних. За оцінками, обсяг інформації, що накопичується в Інтернеті, є справді величезним. У 2024 році повинно існувати приблизно 149 зетабайт даних, і це число зростає в геометричній прогресії, як і населення світу, що здатне їх створити. [2] Зазвичай дані збираються з кількох джерел у різні моменти часу за допомогою різних технологій. Це створює проблеми неоднорідності, експериментальних варіацій та статистичних зміщень, і вимагає від нас розробки більш адаптивних і надійних способів. [1]

Розвідковий аналіз даних (РАД) [3], що забезпечує попередній аналіз з метою виявлення найзагальніших ознак інформації, відомий уже достатньо давно, ще з 1969 року. Наприклад, у роботі [4] надано рекомендації щодо того, як починати будь-який статистичний аналіз, а також дві цілі, що потрібно досягти: опис даних і формулювання моделі. [4] За останні десятиріччя змінилися об'єми, формати, засоби та безпосередньо виміри простору для аналізу даних. Відповідно до [5], двома основними цілями багатовимірного аналізу даних є розробка ефективних методів, які можуть точно передбачити майбутні спостереження, і в той же час отримати уявлення про взаємозв'язок між характеристиками та відповіддю для наукових цілей.

Цілі розвідкового аналізу даних можуть ставитися по-різному, і це залежить від багатьох факторів. У роботі [6], автори характеризують цілі дослідження як: профілювання (оцінка якості даних) і відкриття (отримання нових ідей). Серед багатьох аргументів для запропонованого підходу в [6] слід зазначити, що використання різних експертів предметної області для дослідження даних є перспективним рішенням, яке забезпечує всебічний РАД, тим не менш, існує багато проблем щодо розмежування РАД і статистичного

аналізу даних. У роботі [7], автори дійшли висновку щодо ефективності об'єднання статистичного з РАД та візуальними засобами аналізу.

Оскільки РАД вимагає глибинних аналітичних навичок, досвіду та знань у предметній області, за останнє десятиріччя було розроблено дуже багато пропозицій та відповідних систем для його автоматизації. Зокрема, прогрес у дослідженні машинного навчання (МН) створив потужні можливості не тільки для кращого розуміння результатів РАД, але й для повної автоматизації цього процесу. [8] Дійсно, в індустрії МН отримало широке застосування для РАД, наприклад: для виявлення шахрайства у банківській системі [9], при вивченні хронічних хворіб нирок у медицині [10] чи для аналогової мас-спектрометрії світу океану [11]. Загалом, для автоматизації РАД в індустрії уже запропоновано дуже багато методологій. Зокрема, у роботі [12] запропоновано концепцію розробки інтегральних методологій для управління даними в наукових проєктах на основі трьох фундаментальних методик: проєктної, командної, управління даними та інформацією. Запропонована концепція передбачає, що вона має постійно розвиватися, вдосконалюватися та адаптуватися до нових викликів у галузі даних.

РАД описує ознаки даних [3], тому можна припустити: якби алгоритми могли “говорити”, то вони б розказали нам про ознаки даних “словесно”, а не використовували різні графіки, діаграми тощо. Згідно з [13], природа РАД змінилася у зв'язку з появою нових методів і конвергенції між РАД та іншими методологіями, як-от аналізом даних і повторною вибіркою, тому звичайні концептуальні основи РАД можуть більше не впоратися з цією тенденцією.

Поява великих мовних моделей (ВММ) викликала особливий науковий інтерес до вивчення їх можливостей для застосування у галузі аналізу даних. Згідно з опрацьованими публікаціями [14-17] стає зрозумілим те, що ВММ є потужним інструментом у дослідницькій практиці, який дозволяє проводити якісний аналіз даних, пропонуючи засоби уникнення розбіжностей в аналізі. Очевидно, що в “просторі смислу” [18], де працюють ВММ, поняття, які

зображені на графіках чи діаграмах, можуть бути представленими у вигляді словесного опису і навпаки.

Враховуючи всі вище описані засоби для проведення РАД, тенденції в індустрії щодо побудови аналітичних систем [2] та особливості підготовки даних до РАД [19-21], можна стверджувати: ВММ здатні здійснювати деякі перетворення даних на основі відповідних підказок. Хоча зазвичай в індустрії використовують структуровані або табличні дані, які знаходяться в базах чи сховищах даних, для їх подальшого аналізу потрібно здійснити достатньо багато операцій з перетворення даних. [20, 22]

Враховуючи актуальність наукового дослідження можливостей ВММ для виконання РАД над табличними даними, а також їх практичного застосування при побудові систем аналізу даних, завдання для наукової роботи можна сформулювати наступним чином:

- дослідити можливості підтримки складних обчислювальних операцій над даними у задачах РАД;
- дослідити можливості перетворення даних за допомогою ВММ;
- дослідити можливості аналізу даних за допомогою ВММ;
- розробити концепцію керування аналізом даних за допомогою підказок;
- розробити архітектуру системи для виконання РАД над табличними даними;
- розробити чат-бот систему для апробації запропонованого рішення.

Об'єкт дослідження - виконання розвідникового аналізу даних у просторі "смислу" над даними, поданими в таблицях.

Предмет дослідження - можливості великих мовних моделей до здійснення перетворень та аналізу табличних даних у просторі "смислу".

1. ВИБРАНІ ТЕХНОЛОГІЇ ТА ЗАСОБИ ДЛЯ АНАЛІЗУ ДАНИХ

1.1. Вступ до аналізу даних

Незважаючи на колосальний обсяг інформації, яку ми продукуємо щодня, лише 0,5% фактично аналізується й використовується для виявлення та структурування даних. Це може видатись доволі невеликим числом, але якщо врахувати всю кількість інформації, що існує в цифровому форматі, значення все одно виходять велетенські. [1, 2] Коли мова йде про такі масштаби, одразу згадується тема бізнесу, і дійсно, аналіз даних відіграє в ньому дуже важливу роль. Згідно з [4, 6, 7] він допомагає:

- Приймати ефективні рішення. Як вже зазначалось, в сучасному світі інформація генерується з небувалою швидкістю, а фірми й корпорації мають до неї доступ. Аналіз даних дозволяє підприємствам приймати обґрунтовані рішення, визначати закономірності й тенденції. А це в свою чергу сприяє прийняттю кращих рішень, зменшенню ризиків та збільшенню прибутку.

- Накопичувати досвід роботи з клієнтами. Завдяки аналізу даних, підприємства можуть визначити сфери, де їм потрібно покращити рівень своїх послуг, або ж ліпше адаптувати їх до вимог людей. Таким чином, вони розуміють потреби та вподобання своїх клієнтів, і це допомагає надавати більш персоналізований досвід.

- Отримати перевагу над конкурентами. Вивчення ринкових тенденцій та поведінки користувачів дозволяє оптимізувати свою діяльність, підвищити ефективність, зменшити кількість помилок й таким чином випередити конкурентів.

- Підтримувати ефективність прийняття рішення за рахунок інноваційних технологій. Існує багато напрямків, що мають високий потенціал з точки зору бізнесу, науки та новітніх технологій. Якщо їх досліджувати, можна створювати інноваційні продукти чи послуги швидше за інших, або ж банально вчасно до них адаптуватись і рухатись разом із прогресом.

Із розростом об'ємів даних щоденно, аналіз даних все частіше відіграє важливу роль у процесах прийняття рішень, а його значення стає більш глобальним, особливо для науковців по даних.

1.2. Наукові основи аналізу даних

Узагальнюючи [1-7, 10-12, 19], можна сказати, що аналіз даних - це процес збору, моделювання та вивчення даних за допомогою різних математичних рівнянь, статистичних моделей, засобів моделювання чи моделей машинного навчання і т.п.

Переходячи до більш складних концепцій, давайте згадаємо про деякі типи інформації що часто виступають інструментами в області науки про дані:

- Великі дані [1, 21] - масивні набори даних, що аналізуються з допомогою передового програмного забезпечення для виявлення закономірностей та тенденцій. Вважаються одним із найкращих аналітичних засобів, оскільки забезпечують ширший обсяг інформації з більшою швидкістю.

- Метадані - це дані, що характеризують або пояснюють інші дані. Використовуються, як правило, для підвищення якості пошуку.

- Дані реального часу - це такий тип інформації, що відображається одразу після її отримання. Це найцінніші дані, адже вони можуть допомогти з прийняттям важливих рішень на основі актуальних подій.

- Машинні дані - це цифрова інформація, яка автоматично створюється в результаті роботи мережевих пристроїв: комп'ютерів, мобільних телефонів, вбудованих систем тощо. Цей тип інформації, що містить сервісні дані, також є корисним для аналізу даних, тому що він дозволяє ідентифікувати різні події в часі, або локалізувати джерела інформації та їх походження.

1.3. Процес аналізу даних

Процес аналізу даних є достатньо стандартизованим та формалізованим міжгалузевим стандартним процесом інтелектуального аналізу даних, відомим як CRISP-DM (англ. Cross-Industry Standard Process for Data Mining) - є одним із

найпоширеніших серед загально прийнятих підходів. [19, 23] Згідно CRISP-DM, процес аналізу даних складається з 6 основних етапів:

- Ідентифікація бізнес вимог. На самому першому етапі ми заглиблюємося у бізнес проблему, де визначаємо, для чого саме потрібні дані. Це етап, на якому ви ставите ключові запитання, на які самі ж повинні відповісти. Наприклад, як клієнти сприймуть наш бренд? Або який тип упаковки більше їх зацікавить? Тільки після цього можна перейти до наступного кроку. З наукової точки зору, цей процес можна охарактеризувати як визначення умов та цілей для проведення наукового експерименту чи визначення умов та критеріїв вирішення наукової проблеми.

- Розуміння даних. На цьому етапі ми маємо проаналізувати дані та визначити, які дані ми маємо і як ми можемо їх оцінити. Саме на цьому етапі із застосовує розвідниковий аналіз даних. У нашому випадку, розвідниковий аналіз табличних даних є предметом нашого дослідження, але у просторі “смыслу”.

- Збір та підготовка даних. Даних етап характеризується інженерними рішеннями та є серед головних обов’язків інженерів по даних, аналітиків даних, інженерів МН та науковців по даних. Саме тут виконуються задачі щодо перетворення даних та підготовка їх до використання визначеними алгоритмами. Саме до цього етапу мають відношення згадані вище кількісний та якісний методи дослідження. Серед методів підготовки даних також ми їх і очищаємо тому, що не все, що ми збираємо, є потенційно корисним. Під час обробки великого обсягу інформації в різних форматах, дуже ймовірно, що трапляться дублікати або погано відформатовані дані. Щоб уникнути цього, перед початком роботи за допомогою коду можна стерти пробіли, копії та погане форматування, для уникнення зайвих помилок.

- Моделювання. За допомогою моделювання ми пробуємо відшукати найбільш ефективний спосіб вирішення заданої бізнес-задачі. У випадку наукових досліджень - це найкраща модель, спосіб чи метод розв’язання наукової проблеми.

- Оцінювання якості вирішення проблеми та аналіз помилок. На цьому етапі здійснюється пошук тенденцій, кореляцій, варіацій та закономірностей, що допомагають дати відповіді на запитання, чи вирішена проблема чи ні, і які ще дані нам потрібно визначити на етапі ідентифікації бізнес-проблеми. Дослідник розробляє план дій на основі отриманих результатів, часто може виявити нюанси чи обмеження, і звичайно ж працює над тим, як знайти практичне застосування усій зібраній інформації.

- Розгортання. Цей етап скоріше технічний ніж науковий, на якому розгортають апробовані моделі чи алгоритми для вирішення проблем у реальних умовах.

Наведений міжгалузевий стандарт CRISP-DM враховує ітераційність процесу аналізу даних та забезпечує успішний ітераційний пошук найкращого рішення на основі РАД та моделювання.

1.4. Методи та технології для аналізу даних

Існує багато методів та алгоритмів для аналізу даних, і вибір конкретного з них залежить від характеру даних та поставлених завдань. Загалом, методи для кількісного та якісного способів дослідження мають спільні риси в межах своєї категорії, тому ми зосередимося на найбільш поширеніших.

1.4.1. Кластерний аналіз

Групування елементів в багато різних наборів за їх спільними ознаками. Такі набори називаються кластери, звідси й походження назви. [24, 25] З тієї причини, що під час кластеризації немає цільової змінної, метод часто використовується для пошуку прихованих патернів даних, а також для надання додаткового контексту тенденціям.

1.4.2. Багатопросторове масштабування

Багатопросторове масштабування (англ. Multidimensional Scaling, MDS) [26] - це метод, який використовується для спостереження подібності або відмінності між об'єктами, які можуть бути кольорами, брендами, людьми, географічними координатами тощо. Об'єкти зображені за допомогою «MDS-карти», яка позиціонує схожі об'єкти разом і розрізнені далеко один від одного. Спільні риси об'єктів представлені у вигляді одного або декількох вимірів, які можна спостерігати за допомогою чисельного масштабу. Наприклад, якщо ви хочете знати, як люди ставляться до вакцини проти COVID-19, ви можете використовувати 1 для «взагалі не вірять у вакцину» та 10 для «твердо вірять у вакцину» та шкалу від 2 до 9 для між відповідями. При аналізі MDS карти єдине, що має значення відстань між об'єктами, орієнтація розмірів довільна і не має сенсу взагалі.

1.4.3. Дерева рішень

Дерева рішень, як інструмент підтримки для прийняття стратегічних рішень, мають широке застосування на практиці. [9] Візуально відображаючи потенційні результати, наслідки та витрати в деревоподібній моделі, дослідники можуть легко оцінити всі сценарії, задіяні фактори та вибрати найкраще рішення. Дерева рішень допомагають аналізувати кількісні дані та дозволяють поліпшити процес прийняття рішень, допомагаючи нам визначити можливості поліпшення, та зменшення витрат. Цей метод працює як блок-схема, яка починається з основного рішення, яке потрібно прийняти, а розгалужується на основі різних результатів і наслідків кожного рішення. Кожен результат окреслює свої наслідки, витрати та прибутки, і наприкінці аналізу є можливість кожен з них порівняти.

1.4.4. Видобуток даних

Добування даних передбачає вивчення та аналіз великих блоків інформації для збору значущих моделей та тенденцій. Він використовується в

управлінні кредитними ризиками, виявлення шахрайства та фільтрації спаму. Це також інструмент дослідження ринку, який допомагає виявити настрої або думки певної групи людей. [19, 23]

Зазвичай, видобуток даних застосовується в інтелектуальному аналізі даних для різних тематичних областей [19] на етапі підготовки даних для моделювання. В основному, такі задачі здійснюються для видобування необхідних атрибутів із доступних ресурсів, особливо з текстових ресурсів.

1.4.5. Машинне навчання

Машинне навчання є підмножиною штучного інтелекту (ШІ), що фокусується на розробці комп'ютерних алгоритмів, які покращуються автоматично за допомогою досвіду та використання даних. Іншими словами, машинне навчання дозволяє комп'ютерам вчитися на даних і приймати рішення або прогнози без явного програмування. За своєю суттю, воно є створенням та впровадженням алгоритмів, що здатні полегшити рішення та прогнози. [9-11]

Здатність вчитися на даних і вдосконалюватися з часом робить МН універсальним потужним засобом для аналізу даних, тому, за останні десятиліття, МШ показало, що це рушійна сила від голосових помічників та рекомендаційних систем до безпілотних автомобілів та прогнозової аналітики.

МШ часто плутають зі ШІ або глибинним навчанням (ГН). Різниця між цими поняттями полягає в наступному [27]:

- ШІ відноситься до розробки програм, які ведуть себе “розумно” та імітують людський інтелект за допомогою набору алгоритмів. Все фокусується на трьох навичках: навчання, міркування та самокорекція для отримання максимальної ефективності. ШІ може стосуватися як програм на основі МН, так і навіть явно запрограмованих комп'ютерних програм.

- МН - це область ШІ, яка використовує алгоритми, що навчаються з даних, для того що робити прогнози. Ці передбачення можуть бути згенеровані шляхом керованого навчання, де алгоритми вивчають закономірності з існуючих даних або навчання без нагляду, де вони виявляють загальні закономірності в

даних. Моделі ML можуть передбачати числові значення на основі історичних даних, класифікувати події як істинні або хибні, а також кластерні точки даних на основі обраних мір схожості.

- ГН - це є область МН, що займається алгоритмами, які побудовані на основі глибинних штучних нейронних мереж.

- На відміну від звичайних алгоритмів МН, алгоритми ГН є менш лінійними, складнішими та ієрархічними, здатними вчитися з величезної кількості даних та давати високоточні результати.

1.4.6. Штучні нейронні мережі

Останнім часом в галузі ІТ все частіше починають застосовуватися штучна нейронна мережа (ШНМ) як інструмент для аналізу даних. Найкращі приклади можна навести для виявлення аномалій, пошуку близьких за значенням прикладів, побудова рекомендаційних систем та ін. [28-30]

ШНМ імітують людський мозок, у якому біологічні нейрони передають один одному сигнали. На відміну від мозку, ШНМ складаються з вузлових шарів, що містять вхідний, один або кілька прихованих і вихідний шари. Кожен вузол, або штучний нейрон, з'єднується з іншим та має відповідну вагу й активаційну функцію. Коли нейрони ставлять запит або проблему для вирішення, вони проводять математичні розрахунки, щоб з'ясувати, чи достатньо інформації для передачі інформації наступному нейрону.

ГН, на відміну від традиційного машинного навчання та алгоритмів інтелектуального аналізу даних, може створити представлення даних надзвичайно високого рівня з величезної кількості необроблених даних. [29]

Алгоритми ГН витягують складні абстракції високого рівня як представлення даних за допомогою ієрархічного процесу навчання, здійснюють вилучення складних шаблонів із величезних обсягів даних, семантичне індексування, тегування даних, швидкий пошук інформації та спрощення дискримінаційних завдань. Ключовою перевагою глибокого навчання є аналіз і вивчення величезних обсягів неконтрольованих даних, що робить його цінним

інструментом для аналітики великих даних, де необроблені дані здебільшого не позначені та не категоризовані. [30]

Не дивлячись на те, скільки переваг мають нейронні мережі, в них є й суттєвий недолік: загалом, їх важко перевірити. Деякі нейромережеві процеси можуть вести себе як «чорний ящик», куди подаються вхідні дані, де виконуються складні процеси, звідки повідомляється лише про вихід.

1.4.7. Простір “смислу” та великі мовні моделі

Велика мовна модель - це тип моделі ГН, яка розуміє та генерує текст людською мовою. Ці моделі навчаються на величезних обсягах текстових даних (книги, статті, сайти та інші джерела) і містять велику кількість параметрів. Параметри — це змінні, які є в моделі, вони змінюються в процесі навчання. Вважається, що мовна модель є великою, якщо містить більше одного мільярда параметрів. Саме завдяки численній кількості параметрів ВММ і здатні розпізнавати, перекладати, прогнозувати чи генерувати текст чи інший контент. Зазвичай на вхід до таких моделей подається одне або кілька речень, спираючись на які модель намагається зрозуміти, що від неї хочуть, і генерує відповідь. [14-17]

ВММ, що побудовані на архітектурі трансформер, здатні навчатися без спостереження, хоча точніше сказати, що вони здійснюють самонавчання. Саме завдяки цьому процесу вони починають розуміти базову граматику та розрізняти мови, а також засвоювати знання. На відміну від попередніх архітектур - рекурентних нейронних мереж, які послідовно обробляють вхідні дані, трансформери обробляють цілі послідовності паралельно.

Ключовим чинником у роботі ВММ і те, як вони представляють слова. [14] Більш ранні форми машинного навчання використовували числову таблицю подання кожного слова [18]. Але ця форма представлення не могла розпізнати взаємозв'язки між словами, наприклад, що мають схожі значення. Це обмеження було подолано з допомогою використання багатовимірних векторів, так званих

“ембеддингів” слів, щоб слова зі схожими контекстними значеннями чи іншими взаємозв'язками перебували близько друг до друга у векторному просторі.

У роботі [18] можна відшукати пояснення векторного простору для відображення слів. Оскільки, ми можемо здійснювати операції на словами у векторному просторі, то ми можемо стверджувати, що цей простір можна вважати простором “смислу”. У цьому просторі працюють текстологічні методи пошуку інформації.

Використовуючи ембеддинг слів, трансформери можуть попередньо обробляти текст у вигляді числових представлень через кодувальник, розуміти контекст слів і фраз зі схожими значеннями, а також інші взаємозв'язки між словами, наприклад частинами мови. Потім ВММ використовує ці знання мови з використанням дешифратора для отримання вихідних даних. [14, 17]

Загалом процес навчання ВММ дуже подібний до того, як тренуються ШНМ. Найпоширеніші три моделі навчання ВММ:

- Навчання без прикладів. Базові ВММ можуть реагувати широкий спектр запитів без спеціального навчання, часто з допомогою підказок, хоча точність відповідей в такому випадку не стабільна.

- Навчання у кілька прийомів. Навівши якесь число відповідних прикладів - параметрів, можна значно підвищити продуктивність базової моделі у конкретній галузі.

- Точне налаштування - це розширення навчання у кілька прийомів, під час якого фахівці з аналізу даних навчають базову модель коригувати свої параметри з допомогою додаткових даних для конкретного застосування.

Спеціалізовані інструменти ШІ також існують для самих науковців, що займаються аналізом даних. Ці застосунки зазвичай підключаються до баз даних і надають користувачам набір функцій для роботи з файлами.

DataLang здійснює аналіз даних на основі підказок, та не вимагає технічних знань. Він представляє результат у доступній для людини формі за допомогою технологій GPT-3. Інструмент може обробляти широкий спектр

запитань і надавати інформацію про такі показники, як кількість користувачів протягом певного періоду часу або загальні значення в певному полі.

Owlbot забезпечує багатоджерельну інтеграцію інформації: документи, електронні таблиці, бази даних. Hal9 дає можливість спілкуватися з хмарними даними компанії. ChatCSV працює з файлами CSV та з деякими засобами візуалізації діаграм для відповідей.

1.5. Використання чат-бот систем

Чат-боти широкого призначення, як-от ChatGPT, Microsoft Bing Chat, розроблені для загального користування, оскільки вони здатні вести розмову, схожу до людської, на різні теми. Бізнес-орієнтовані боти, наприклад Claude, теж вважаються загальними, якщо їх функціональні можливості досить широкі.

Дослідницькі чат-боти мають вищий рівень точності, ніж попередні, і найкраще підходять для вивчення конкретного предмету. Їх часто використовують як інструменти запитань і відповідей, хоча вони мають й інші сильні сторони. Perplexity відомий своєю генерацією ідей та аналізом тексту, Khanmingo є надійним інструментом навчання для тих, хто працює в галузі освіти, Poe використовує інші нейронні мережі, щоб бути ефективнішим.

У роботі [31] показано, що застосування природної мови для покращення взаємодії студентів з інформаційними системами є корисним. Зокрема, досягнення в когнітивних обчисленнях дозволяють новий спосіб взаємодії, який прискорює розуміння з існуючих джерел інформації, тим самим сприяючи процесу навчання. Ця система була реалізована як персональний агент, щоб допомогти студентам у вивченні методів науки про дані та МН. Його реалізація включає навчання моделей машинного навчання та алгоритмів розуміння природної мови в людиноподібному інтерфейсі. Ефективність системи підтверджено експериментально. [31]

Із ростом популярності ВММ, дослідження чат-ботів стрімко зростають з 2021 року та проводяться на основі теорії впровадження технологій. В роботі [32] наведено результати багатьох дослідження чат-бот систем, які

підтверджують вплив людяності чат-бота. Однак досліджень щодо окремих факторів і стратегій, які впливають на впровадження та поширення чат-ботів, ще є недостатньо, тому в роботі надано вказівки щодо напрямків майбутніх досліджень шляхом визначення тенденцій дослідження, пов'язаних із чат-ботами, аналізу великих даних та ін.

В роботі [33] наведено рішення стосовно чат-ботів, що є удосконаленими бізнес-інтелектом на основі інтеграції з великими даними. Це є найближчий аналог до запропонованої нами чат-бот системи.

Інтеграція великих даних як бази знань у чат-боти може генерувати динамічні відповіді на запити користувачів і покращити аналітичні можливості чат-ботів із даними з розподіленого середовища. Ця передова технологія безпосередньо відкриває світ великих даних для чат-ботів, дозволяючи також використовувати чат-ботів як інструмент аналізу бізнес-розвідки. [33]

Приклад професійної системи для бібліометричного аналізу публікацій показано в роботі [34]. Автори розробили чат-бот на основі ChatGPT, який дозволяє відшукувати найбільш релевантні публікації. Це рішення також можна вважати найближчим аналогом до нашої системи.

2. ЗАСОБИ АНАЛІЗУ ДАНИХ У ПРОСТОРИ “СМИСЛУ”

2.1. Дослідження ВММ в задачах аналізу даних

Дослідження можливостей ВММ в задачах аналізу були проведені із використанням ChatGPT 3.5. На першому етапі були згенеровані табличні дані, а далі, використовуючи різні підказки досліджувався аналіз даних. Деякі приклади наведено нижче.

Підказка 1:

"Згенеруй дані для таблиці з такими колонками:

1. Назва компанії (цей атрибут не повинен повторюватись)
2. Галузь діяльності
3. Країна базування
4. Розмір компанії (кількість працівників)
5. Річні доходи (в доларах США)
6. Чистий прибуток за останній фінансовий рік
7. Число філій/відділень
8. Середній рівень зарплати працівників
9. Рейтинг задоволення клієнтів
10. Витрати на рекламу та PR

Результат 1:

Назва компанії	Галузь діяльності	Країна базування	Розмір компанії	Річні доходи (\$)	Чистий прибуток (\$)	Число філій/відділень	Середній рівень зарплати	Рейтинг задоволення клієнтів	Витрати на рекламу та PR (\$)
AlphaTech	Інформаційні технології	США	1000	50000000	10000000	50	60000	4.5	2000000
BrightSolutions	Консалтинг	Велика Британія	500	30000000	6000000	25	55000	4.7	1500000
GlobalGoods	Виробництво	Китай	2000	70000000	15000000	75	48000	4.3	2500000
QuantumCorp	Фінанси	США	750	40000000	8000000	40	65000	4.6	1800000
TechSavvy	Інформаційні технології	Індія	1500	60000000	12000000	60	50000	4.4	2200000
PowerHouse	Енергетика	Німеччина	1000	55000000	11000000	55	70000	4.8	1700000
InnovateX	Наукові дослідження	Японія	800	45000000	9000000	45	62000	4.5	1900000
EcoTech	Екологія	Іспанія	300	20000000	4000000	20	58000	4.2	1300000
AlphaOmega	Логістика	Франція	1200	65000000	13000000	65	54000	4.6	2100000
SmartTech	Техніка	США	1800	80000000	16000000	80	70000	4.7	2300000
MegaCorp	Фінанси	Китай	2500	90000000	18000000	90	60000	4.8	2800000
InfoBiz	Інформаційні послуги	Німеччина	600	35000000	7000000	35	58000	4.4	1600000
PrimeIndustries	Виробництво	Росія	1400	60000000	12000000	60	52000	4.5	2000000
GreenEnergy	Енергетика	Китай	800	40000000	8000000	40	60000	4.3	1700000
DataMasters	Аналітика	США	500	30000000	6000000	30	58000	4.5	1900000
EuroTech	Технології	Німеччина	1000	45000000	9000000	45	62000	4.4	1800000
SuperSolutions	Консалтинг	Франція	1500	75000000	15000000	75	64000	4.7	2200000
TechGenius	Інформаційні технології	Велика Британія	1800	80000000	16000000	80	68000	4.8	2500000
AlphaSystems	Системна інтеграція	Канада	1000	50000000	10000000	50	60000	4.6	2000000
DataTech	Дані та аналітика	США	1200	65000000	13000000	65	63000	4.7	2300000
BrightFuture	Інновації	Індія	600	30000000	6000000	30	55000	4.4	1800000

Ці дані є випадковими і не відповідають реальності. Якщо ми хочемо більш реальних даних, то ми можемо вказати додаткові умови у підказках. У цьому випадку, мережа візьме їх до уваги та згенерує дані у заданому діапазоні. Єдиний недолік у такому підході є те, що мережа не може згенерувати велику кількість табличних даних. Для формування статистичних відомостей про дані, ми використовуємо підказки, які наведено нижче.

Підказка 2:

Сформулюй статистику для річних доходів (гістограма): від компанії, що заробляє найбільше - до тієї, що заробляє найменше.

Результат 2:

Для створення гістограми річних доходів потрібно спочатку отримати список річних доходів усіх компаній з таблиці. Потім ми можемо підрахувати кількість компаній, які мають річний дохід в певному діапазоні і побудувати гістограму. Ось код на Python для цього:

```
```python
import matplotlib.pyplot as plt

Річні доходи усіх компаній
річні_доходи = [
 50000000, 30000000, 70000000, 40000000, 60000000, 55000000, 45000000,
 20000000, 65000000, 80000000, 90000000, 35000000, 60000000, 70000000,
 85000000, 40000000, 30000000, 45000000, 25000000, 35000000, 75000000,
 80000000, 50000000, 65000000, 30000000
]

Побудова гістограми
plt.figure(figsize=(10, 6))
plt.hist(річні_доходи, bins=10, color='skyblue', edgecolor='black')
plt.xlabel('Річний дохід ($)')
plt.ylabel('Кількість компаній')
plt.title('Гістограма річних доходів компаній')
plt.grid(True)
plt.show()
```
```

Мережа, у відповідь на нашу підказку, щодо обрахунків генерує код програми на мові Python. Цей код побудує гістограму річних доходів компаній з використанням бібліотеки matplotlib у Python. Таким чином, ми можемо зробити висновок, що мережа надає перевагу обчисленням за допомогою коду, а не безпосередньо проводити їх у просторі “смыслу”.

Підказка 3:

Проведи статистичний аналіз таблиці: обчисли середнє значення розміру компаній.

Результат 3:

Щоб обчислити середнє значення розміру компаній, нам потрібно взяти розмір кожної компанії з таблиці і знайти їх середнє значення. Ось код на Python для цього:

```
```python
Розміри компаній
```

```
розміри_компаній = [
 1000, 500, 2000, 750, 1500, 1000, 800,
 300, 1200, 1800, 2500, 600, 1400, 1200,
 2000, 800, 500, 1000, 400, 700, 1500,
 1800, 1000, 1200, 600
]

Обчислення середнього значення розміру компаній
середнє_значення = sum(розміри_компаній) / len(розміри_компаній)
print("Середнє значення розміру компаній: ", середнє_значення)
'''
```

Результатом буде середнє значення розміру компаній.

#### Підказка 4:

Знайди медіану чистого доходу.

#### Результат 4:

Щоб знайти медіану чистого прибутку, нам потрібно впорядкувати значення чистого прибутку за зростанням і знайти середнє значення двох середніх значень, якщо кількість значень парна, або просто взяти середнє значення, якщо кількість значень непарна. Ось код на Python для цього:

```
```python
# Чистий прибуток за останній фінансовий рік
чистий_прибуток = [
    10000000, 6000000, 15000000, 8000000, 12000000, 11000000, 9000000,
    4000000, 13000000, 16000000, 18000000, 7000000, 12000000, 14000000,
    17000000, 8000000, 6000000, 9000000, 5000000, 7000000, 15000000,
    16000000, 10000000, 13000000, 6000000
]

# Впорядковані значення чистого прибутку
впорядкований_прибуток = sorted(чистий_прибуток)

# Кількість значень чистого прибутку
кількість_значень = len(впорядкований_прибуток)

# Обчислення медіани
if кількість_значень % 2 == 0:
    медіана = (впорядкований_прибуток[кількість_значень // 2 - 1] +
впорядкований_прибуток[кількість_значень // 2]) / 2
else:
    медіана = впорядкований_прибуток[кількість_значень // 2]

print("Медіана чистого прибутку: ", медіана)
'''
```

Результатом буде медіана чистого прибутку.

Підказка 5:

Обчисли, скільки людей (розмір компанії) працює в кожній з галузей діяльності.

Результат 5:

Щоб обчислити кількість людей (розмір компанії), які працюють в кожній галузі діяльності, нам потрібно зібрати дані про розміри компаній і галузі діяльності, а потім підрахувати загальну кількість працівників для кожної галузі. Ось код на Python для цього:

```
```python
Галузі діяльності та розміри компаній (кількість працівників)
галузі_розміри_компаній = {
 'Інформаційні технології': [1000, 1500, 1800, 1200, 2000],
 'Консалтинг': [500, 1500],
 'Виробництво': [2000, 1400, 800],
 'Фінанси': [750, 2500, 1200],
 'Енергетика': [1000, 800, 400],
 'Наукові дослідження': [800],
 'Екологія': [300],
 'Логістика': [1200],
 'Техніка': [1800],
 'Інформаційні послуги': [600],
 'Розробка програмного забезпечення': [700],
 'Телекомунікації': [1200],
 'Відновлювана енергетика': [400],
 'Аналітика': [500],
 'Технології': [1000],
 'Системна інтеграція': [1000],
 'Дані та аналітика': [1200],
 'Інновації': [600]
}

Обчислення загальної кількості працівників для кожної галузі діяльності
результат = {}
for галузь, розміри in галузі_розміри_компаній.items():
 результат[галузь] = sum(розміри)

Виведення результатів
for галузь, кількість in результат.items():
 print(f"У галузі {галузь} працює {кількість} людей.")
...
```
```

Цей код обчислить загальну кількість працівників для кожної галузі діяльності і виведе результати.

Підказка 6:

Створи приклад табличних даних із 100 прикладів у вигляді таких атрибутів даних: назва товару, рік виготовлення, кількість на складі, ціна, опис. Використовуючи згенеровану таблицю зроби її статистичний аналіз та обчисли наступні параметри: Обчислимо середнє значення цін на товари.

Знайдемо медіану кількості товарів на складі.

Порахуємо загальну кількість товарів за кожний рік виготовлення.

Визначимо найдорожчий та найдешевший товари у таблиці.

Побудуємо гістограму розподілу цін на товари.

Результат 6 наведено на рис. 1.

Тепер здійснимо статистичний аналіз цих даних:

1. Обчислимо середнє значення цін на товари.
2. Знайдемо медіану кількості товарів на складі.
3. Порахуємо загальну кількість товарів за кожний рік виготовлення.
4. Визначимо найдорожчий та найдешевший товари у таблиці.
5. Побудуємо гістограму розподілу цін на товари.

Щоб здійснити обчислення та побудувати гістограму, використаємо Python та бібліотеки для аналізу даних. Даваймо реалізуємо ці кроки.

Нижче наведено Python-код для статистичного аналізу та побудови гістограми на основі наведених даних:

```
python Copy code  
  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
  
# Створення DataFrame з наведеними даними  
data = {  
    'Назва товару': ['Чайник', 'Телевізор', 'Кавоварка', 'Фен', 'Смартфон'], # Додайте решту  
    'Рік виготовлення': [2021, 2022, 2020, 2023, 2021], # Додайте решту років  
    'Кількість на складі': [10, 5, 8, 12, 15], # Додайте решту кількостей  
    'Ціна': [250, 1500, 300, 80, 900] # Додайте решту цін  
}  
  
df = pd.DataFrame(data)
```

Message ChatGPT...



Рис. 1. Екранна форма фрагменту відповіді ВММ на підказку 6

Аналізуючи наведені вище та всі інші приклади роботи з ВММ можна зробити наступні висновки, а саме:

- ВММ розуміють структури даних, можуть читати, генерувати та проводити різні операції по структуруванню, наприклад сортування, пошук найбільших та найменших значень, тощо;
- на задане питання, ВММ генерує адекватний код для обчислення заданих статистичних показників, що може означати, що у просторі “смислу” ВММ правильно встановлює відповідність між завданням та згенерованим текстом програми на Python;
- ВММ пояснює код, що дозволяє перевірити чи правильно вона зрозуміла завдання;
- ВММ, дійсно, старається сформулювати відповідь, у відповідності до заданої підказки, що вказує на те, що ми можемо керувати порядком виконання операцій виконуючи РАД.

2.2. Виконання операцій у просторі “смислу”

Для опису операцій у просторі “смислу” візьмемо до уваги приклад відповіді мережі, що зображено на рис. 1:

Тепер здійснимо статистичний аналіз цих даних:

- 1. Обчислимо середнє значення цін на товари.*
- 2. Знайдемо медіану кількості товарів на складі.*
- 3. Порахуємо загальну кількість товарів за кожний рік виготовлення.*
- 4. Визначимо найдорожчий та найдешевший товари у таблиці.*
- 5. Побудуємо гістограму розподілу цін на товари.*

Як видно з цього прикладу, нам не потрібно операції представляти у вигляді формул чи математичного опису, ВММ прекрасно розуміє усі поняття у просторі “смислу”, принаймі прості перетворення. У цьому випадку, ми можемо говорити про технологію підказок, на основі якої ВММ буде здійснювати РАД.

Опираючись на отриманий досвід проведення РАД у [4, 6-7, 9-13] можна сказати, що РАД являє собою послідовність операцій над даними. Задані

операції залежать від цілей та завдань РАД. Навіть коли ми маємо різні цілі, то, деякі частини чи операції РАД можуть бути використані для них. Звідси слідує, що РАД, згідно рис. 1, математично у вигляді підказки можна зобразити наступним чином:

“““““

Оригінальні дані: “...”

$$\{ \text{Інструкція 1} \} \cup \{ \text{Оригінальні дані} \} \cup \{ \text{Інструкція 2} \} \cup \{ \text{Операція 1} \} \cup \{ \text{Операція 2} \} \cup \{ \text{Операція 3} \} \cup \{ \text{Операція 4} \} \cup \{ \text{Операція 5} \}, \quad (1)$$

”””””

де: Інструкція 1 = “*Виконай статистичний аналіз над оригінальними даними:*”, Інструкція 2 = “*у порядку виконання наступних операцій:* ”, Операція 1 = “*Обчислимо середнє значення цін на товари;*”, Операція 2 = “*Знайдемо медіану кількості товарів на складі;*”, Операція 3 = “*Порахуємо загальну кількість товарів за кожний рік виготовлення;*”, Операція 4 = “*Визначимо найдорожчий та найдешевший товари у таблиці;*”, Операція 5 = “*Побудуємо гістограму розподілу цін на товари.*”.

Як видно з наведено прикладу, ми можемо легко конструювати порядок виконання операцій у РАД у відповідності до його цілей та завдання.

2.3. Архітектура чат-бот системи

Для ефективного виконання розвідкового аналізу табличних даних та дослідження можливостей ВММ розроблено відповідну чат-бот систему. Основна увага чат-бот системи зосереджено на комунікацію з Microsoft DialoGPT та базою даних (рис. 2).

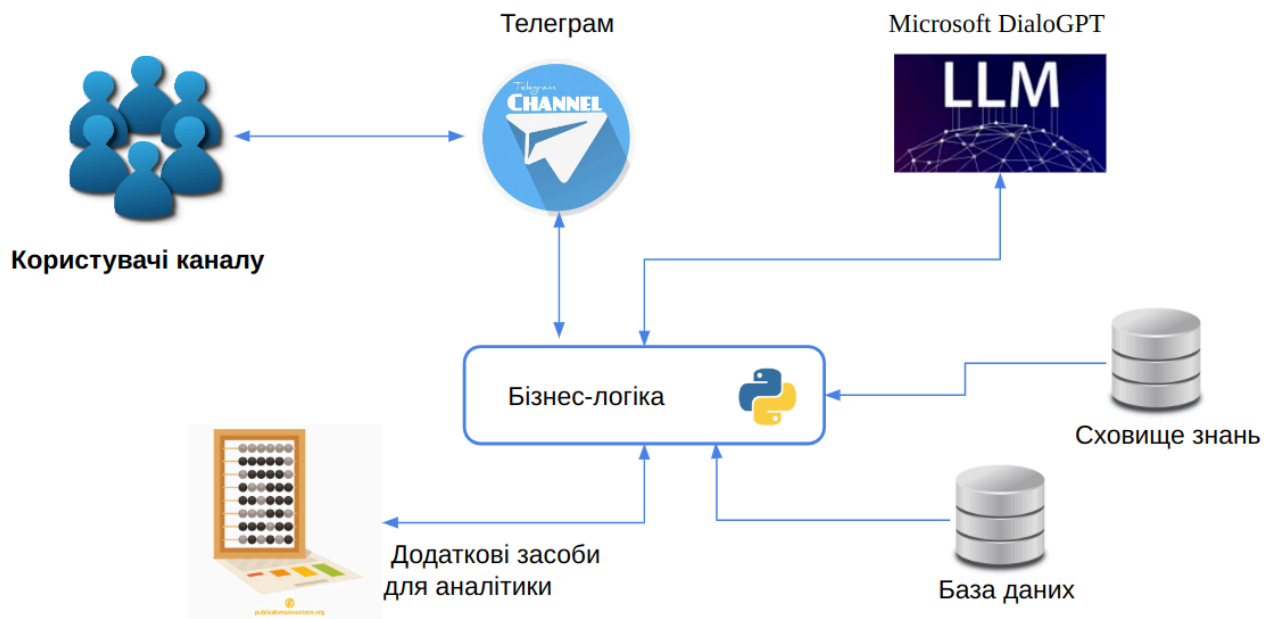


Рис. 2. Архітектура чат-бот системи для проведення досліджень

В основі чат-боту лежить бізнес-логіка, яка керує роботою всієї чат-бот системи. Через телеграм каналу, користувачі можуть спілкуватися з ВММ. У випадку, коли потрібно використати якусь команду, то ми можемо набрати її прямо у каналі. Це дуже зручно, особливо для тих, хто уже давно користується і не потребує друкування великих і довгих повідомлень ВММ.

Бізнес-логіка чат-боку має зв'язок з базою даних та сховищем знань. У базі даних містяться табличні дані, а у базі знань у нас мітяться шаблони підказок. У майбутньому планується розширення можливостей чат-боту за рахунок контексту та інших знань у сховищі знань. Також, бізнес-логіка може використовувати додаткові засоби аналітики, що виходять за рамки можливостей ВММ, наприклад, розрахунок логарифмічних функцій, тощо.

3. ЧАТ-БОТ СИСТЕМА ДЛЯ АНАЛІЗУ ТАБЛИЧНИХ ДАНИХ

3.1. Створення інтерфейсу в системі Telegram

Чому саме Telegram? Перш за все, це популярна месенджерська платформа, яка дозволяє користувачам обмінюватися повідомленнями, файлами, фотографіями та іншими мультимедійними даними. Широко використовується у світі як інструмент для спілкування, обміну інформацією чи організації спільнот. Також, Telegram має достатньо зручних засобів для розробки ботів, а саме:

- Прикладний програмний інтерфейс (API) для розробників. Telegram надає потужний API, що дозволяє створювати різноманітних ботів. Він має ряд функцій, таких як обробка повідомлень, робота з клавішами, відправка зображень і файлів тощо.

- Можливості ботів. Боти в Telegram здатні виконувати різноманітні завдання, від надсилань повідомлень до автоматизації завдань, що є досить цікаво для аналітичних додатків.

- Inline-режим. Telegram дозволяє створювати inline-ботів, які можуть відповідати на запитання і виконувати завдання прямо в текстових полях чату. Це спрощує і прискорює взаємодію з користувачами.

- Багатомовна підтримка. Telegram підтримує багато мов, що дозволяє розробникам створювати ботів, які можуть взаємодіяти з користувачами із різних країн.

- Гнучкість та безпека. Telegram був розроблений з акцентом на безпеку. Боти також можуть використовувати шифрування для захисту конфіденційної інформації.

- Безкоштовне використання. Мабуть, це одна із основних причин - практично все доступно для розробників безкоштовно.

Для того, щоб почати працювати з телеграмом можна використати вказівки бота-помічника BotFather (рис. 3).

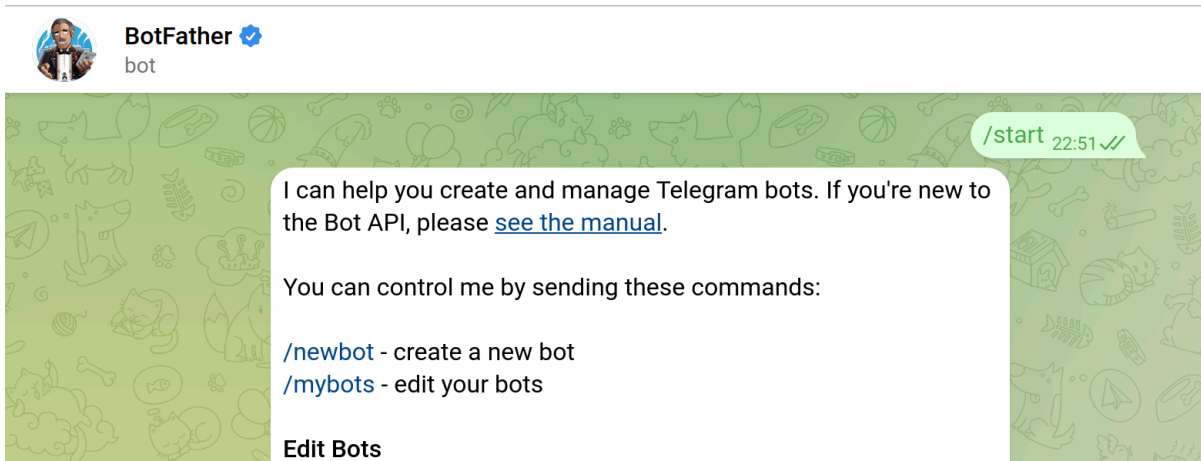


Рис. 3. BotFather - зручний інструмент для керування інтерфейсу

В процесі спілкування, бот-помічник BotFather допомагає створити нового чат-бота, що зображено на рис. 4.

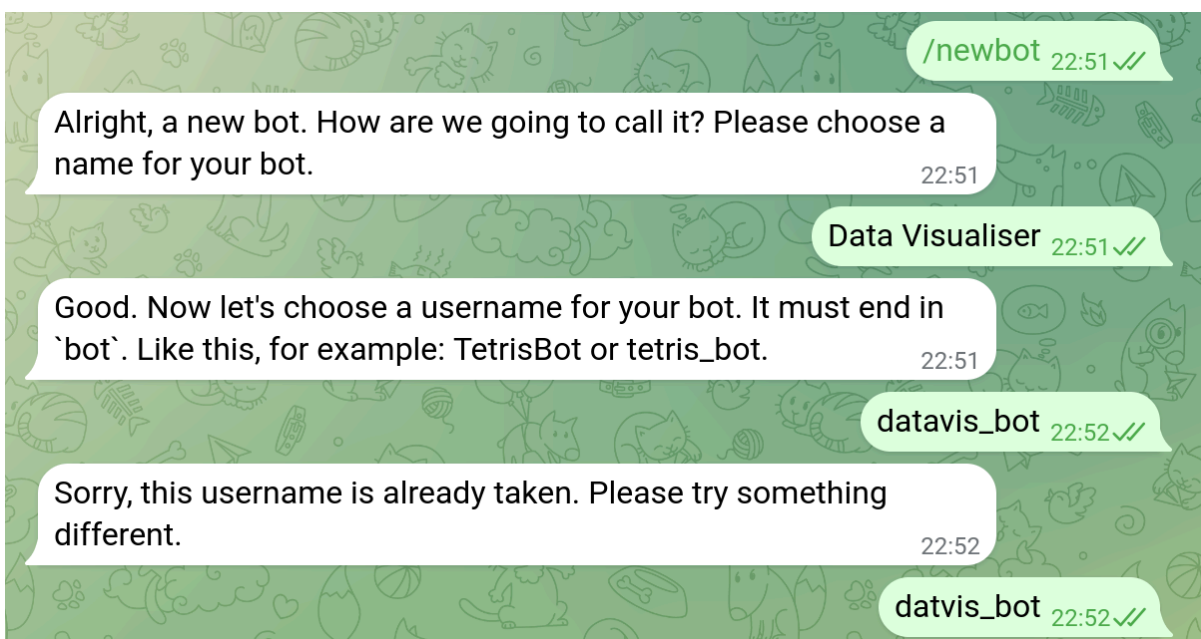


Рис. 4. Частина процесу налаштування базових характеристик бота

Після введення кількох простих команд, інтерфейс бота готовий (рис. 5), але поки він ще не вмє нічого робити.

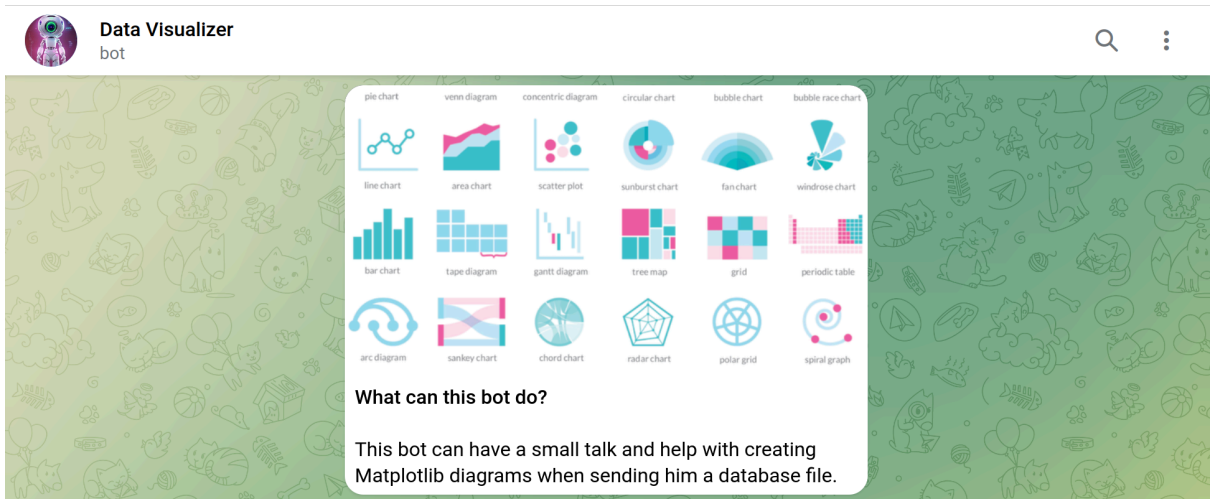


Рис. 5. “Сирий” візуалізатор даних

На даному етапі, розроблений чат-бот готовий до інтеграції із бек-енд частиною, в якій організовано бізнес-логіку чат-бот системи для аналізу даних.

3.2. Організація зв'язку з ВММ

Для розроблення бізнес-логіки ми використали мову програмування Python, яка відома своєю читабельністю та універсальністю, доступна для початківців. Підтримує кілька парадигм програмування, має простий синтаксис та широкий спектр бібліотек, таких як python-telegram-bot для взаємодії з API Telegram, бібліотеки для МН, такі як transformers. Це робить Python хорошим вибором для ефективної розробки чат-бота.

Серед великої кількості ВММ було обрано авторегресійну мовна модель Microsoft DialoGPT. Це означає, що вона може передбачити наступне майбутнє слово з послідовності слів, які задано на вхід. Дана мережа була навчена на текстових даних за допомогою GPT-3. Однією з її особливостей є вибір тону відповідей чат-бота: креативний, збалансований та точний. Щоб почати роботу, користувачам потрібно написати тему тексту, а потім визначити різні його характеристики відповідно до своїх потреб: тон, формат і довжина (рис. 6).

```

3 import torch
4 import telebot
5 import pandas as pd
6 import matplotlib.pyplot as plt
7 from transformers import AutoModelForCausalLM, AutoTokenizer
8
9 # Get data
10 df = pd.read_csv('data/data.csv')
11
12 # Settings for LLM
13 tokenizer = AutoTokenizer.from_pretrained("microsoft/DialoGPT-large", padding_side='left')
14 model = AutoModelForCausalLM.from_pretrained("microsoft/DialoGPT-large")
15 chat_history_ids = torch.tensor([[0]])
16 step = 0
17
18 # Settings for Telegram Bot
19 BOT_TOKEN = os.environ.get('BOT_TOKEN')
20 bot = telebot.TeleBot(BOT_TOKEN)
21
22 ~

```

Рис. 6. Виклик бібліотек, ініціалізація токенів

Приклад функції для оброблення подій в чат-боті наведено на рис. 7.

```

@bot.message_handler(commands=['start', 'hello'])
def send_welcome(message):
    bot.reply_to(message, "Nice to talk with you!")

@bot.message_handler(func=lambda msg: True)
def echo_all(message):
    chat_id = message.chat.id
    global chat_history_ids, step

    # encode the new user input, add the eos_token and return a tensor in Pytorch
    new_user_input_ids = tokenizer.encode(message.text + tokenizer.eos_token, return_tensors='pt')

    # append the new user input tokens to the chat history
    bot_input_ids = torch.cat([chat_history_ids, new_user_input_ids], dim=-1) if step > 0 else new_user_input_ids

    # generated a response while limiting the total chat history to 1000 tokens,
    chat_history_ids = model.generate(bot_input_ids, max_length=1000, pad_token_id=tokenizer.eos_token_id)

    # pretty print last output tokens from bot
    output_text = tokenizer.decode(chat_history_ids[:, bot_input_ids.shape[-1]:][0], skip_special_tokens=True)

```

Рис. 7. Функція для обробки подій в чат-боті з LLM

Деколи виникають ситуації, коли не потрібно викликати для пояснення ВММ, тому, у чат-боті передбачено кілька команд, приклад декільки з них наведено на рис. 8.

```

try:
    if 'plot for' in message.text:
        mes_text = message.text.split()
        columns = [i for i in mes_text if i in list(df.head())]
        df[columns].plot()
        plt.savefig('imgs/plot.png')
        bot.reply_to(message, "Look at the plot for chosen columns")
        bot.send_photo(chat_id, photo=open('imgs/plot.png', 'rb'))

    elif 'plot' in message.text:
        df.plot()
        plt.savefig('imgs/plot.png')
        bot.reply_to(message, "Please, see at a plot of the database")
        bot.send_photo(chat_id, photo=open('imgs/plot.png', 'rb'))

    elif 'bar for' in message.text:
        mes_text = message.text.split()
        columns = [i for i in mes_text if i in list(df.head())]
        df[columns].plot.bar()
        plt.savefig('imgs/bar.png')
        bot.reply_to(message, "Look at the bar plot for chosen columns")
        bot.send_photo(chat_id, photo=open('imgs/bar.png', 'rb'))

```

Рис. 8. Приклад зарезервованих команд для чат-бота

Також, зарезервовані команди використовуються для візуалізації даних. Таке рішення прийнято тому, що ВММ працює з текстовою інформацією, яку має на вході і на виході. Якщо ми хочемо візуалізувати результати за допомогою ВММ, то тоді нам потрібно було б використовувати одну або дві інші ВММ.

3.3. Робота з базою даних

Для прикладу бази даних ми використали ресурс [kaggle.com](https://www.kaggle.com/) - місце, де у вільному доступі є датасети для користування. Саме звідти був взятий файл `Cardiovascular_Disease_Dataset.csv`, про який варто сказати кілька слів (рис. 9).

Це доступний для дослідницьких цілей датасет про хвороби серця, зібраний в одній з багатопрофільних лікарень Індії. Він складається із 1000

суб'єктів з 14 особливостями. Серед них є ID, вік, стать, рівні багатьох медичних показників пацієнтів.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|----|-----------|-----|--------|-----------|-----------|------------------|-------------------|----------------|--------------|---------------|---------|-------|------------------|--------|
| 1 | patientid | age | gender | chestpain | restingBP | serumcholesterol | fastingbloodsugar | restingelectro | maxheartrate | exerciseangia | oldpeak | slope | noofmajorvessels | target |
| 2 | 103368 | 53 | 1 | 1 | 171 | 0 | 0 | 1 | 147 | 0 | 5,3 | 3 | 3 | 1 |
| 3 | 119250 | 40 | 1 | 0 | 94 | 229 | 0 | 1 | 115 | 0 | 3,7 | 1 | 1 | 0 |
| 4 | 119372 | 49 | 1 | 2 | 133 | 142 | 0 | 0 | 202 | 1 | 5 | 1 | 0 | 0 |
| 5 | 132514 | 43 | 1 | 0 | 138 | 295 | 1 | 1 | 153 | 0 | 3,2 | 2 | 2 | 1 |
| 6 | 146211 | 31 | 1 | 1 | 199 | 0 | 0 | 2 | 136 | 0 | 5,3 | 3 | 2 | 1 |
| 7 | 148462 | 24 | 1 | 1 | 173 | 0 | 0 | 0 | 161 | 0 | 4,7 | 3 | 2 | 1 |
| 8 | 168686 | 79 | 1 | 2 | 130 | 240 | 0 | 2 | 157 | 0 | 2,5 | 2 | 1 | 1 |
| 9 | 170498 | 52 | 1 | 0 | 127 | 345 | 0 | 0 | 192 | 1 | 4,9 | 1 | 0 | 0 |
| 10 | 188225 | 62 | 1 | 0 | 121 | 357 | 0 | 1 | 138 | 0 | 2,8 | 0 | 0 | 0 |
| 11 | 192523 | 61 | 0 | 0 | 190 | 181 | 0 | 1 | 150 | 0 | 2,9 | 2 | 0 | 1 |
| 12 | 201030 | 59 | 0 | 1 | 190 | 529 | 1 | 1 | 151 | 1 | 3,2 | 2 | 2 | 1 |
| 13 | 208877 | 58 | 1 | 2 | 192 | 409 | 1 | 0 | 138 | 0 | 2,3 | 3 | 1 | 1 |
| 14 | 223205 | 27 | 1 | 0 | 128 | 135 | 0 | 1 | 102 | 1 | 1 | 0 | 0 | 0 |

Рис. 9. Таблиця з показниками пацієнтів

Перед тим, як запуснути бота, ми створили файл бази даних MySQL. Це легко робиться мовою Python через Pandas. Вона забезпечує прості у використанні структури даних, такі як дата фрейми, і функції, призначені для забезпечення безперебійної роботи зі структурованими даними (рис. 10). Приклад читання і запису даних у базу даних наведено на рис. 11.

```

14 # Establish a connection to the MySQL server
15 connection = mysql.connector.connect(
16     host=hostname,
17     user=username,
18     password=password,
19     database=database
20 )
21
22
23 def read_db():
24     # Define your SQL query to fetch data from the database
25     sql_query = 'SELECT * FROM new_db.info'
26
27     # Read data from the database into a Pandas DataFrame
28     df = pd.read_sql(sql_query, connection)
29
30     # Close the connection
31     connection.close()
32
33     # Display the DataFrame
34     print(df)
35
36     return df
.. ..

```

Рис. 10. Функції для читання даних з бази

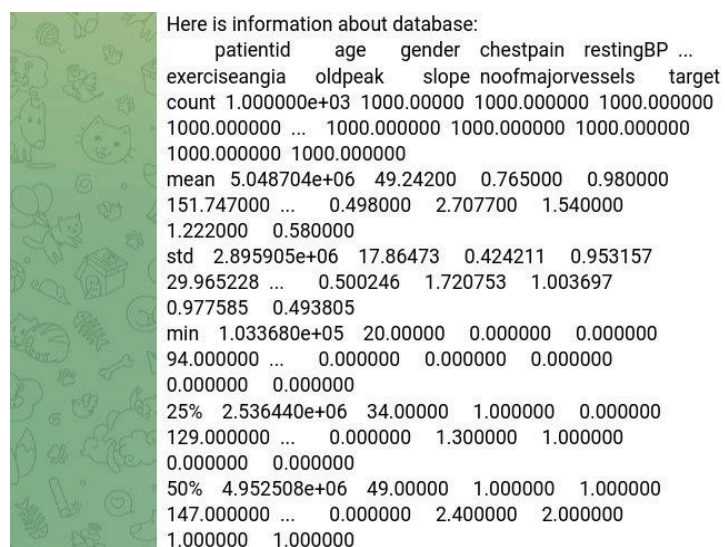
```

39 def create_db():
40     df = pd.read_csv('data.csv')
41     df1 = df[["patientid", "age", "gender", "chestpain", "restingBP"]]
42
43     try:
44         mydb = mysql.connector.connect(
45             host='127.0.0.1',
46             port='3306',
47             user="admin",
48             password="admin",
49             database="new_db"
50         )
51         print("Connection established")
52         cursor = mydb.cursor()
53         cursor.execute("create database if not exists new_db")
54         mydb.commit()
55         print("Database created successfully")
56         cursor.execute("use new_db")
57
58     except mysql.connector.Error as err:
59         print("An error occurred:", err)
60
61     engine = create_engine("mysql+pymysql://{user}:{pw}@{host}:{port}/{db}".format(user=username,
62

```

Рис. 11. Створення бази і запис даних

За умови, що дані завантажені у базу даних, чат-бот, коли стартує, завантажує їх у пам'ять і продовжує там працювати з ними. За бажанням, користувач може повторно зчитати дані, якщо там могли б бути зміни. Приклад читання бази даних наведено на рис. 12.



```

Here is information about database:
  patientid  age  gender  chestpain  restingBP ...
count 1.000000e+03 1000.000000 1000.000000 1000.000000
1000.000000 ... 1000.000000 1000.000000 1000.000000
1000.000000 1000.000000
mean 5.048704e+06 49.24200 0.765000 0.980000
151.747000 ... 0.498000 2.707700 1.540000
1.222000 0.580000
std 2.895905e+06 17.86473 0.424211 0.953157
29.965228 ... 0.500246 1.720753 1.003697
0.977585 0.493805
min 1.033680e+05 20.00000 0.000000 0.000000
94.000000 ... 0.000000 0.000000 0.000000
0.000000 0.000000
25% 2.536440e+06 34.00000 1.000000 0.000000
129.000000 ... 0.000000 1.300000 1.000000
0.000000 0.000000
50% 4.952508e+06 49.00000 1.000000 1.000000
147.000000 ... 0.000000 2.400000 2.000000
1.000000 1.000000

```

Рис. 12. Відповідь на команду “read db” - вивід першої частини таблиці з БД

3.4. Візуалізація даних

Через не дуже хорошу тренованість, ВММ відповідає не ідеально, але тим не менш, повністю розрізняє те, що від неї вимагається. Якщо мережа не знає як відповісти, то повертається те саме повідомлення (рис. 13). Приклад опису атрибутів даних показано на рис. 14.

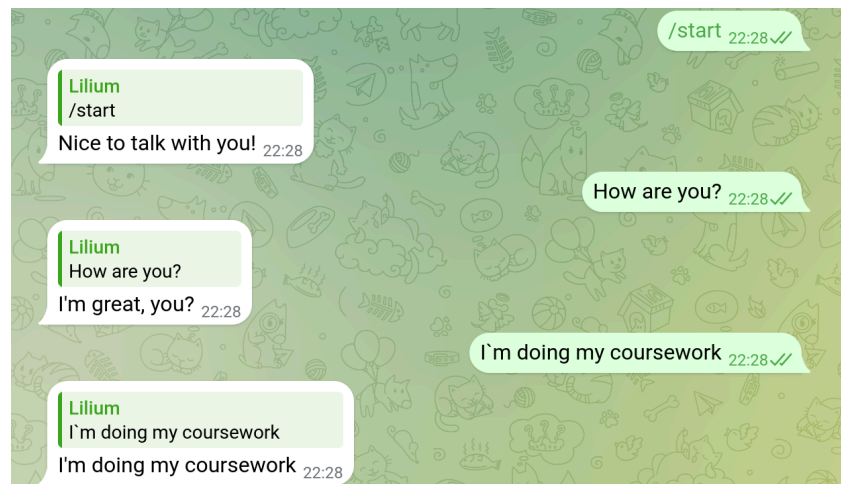


Рис. 13. Приклад повернення повідомлення користувачу

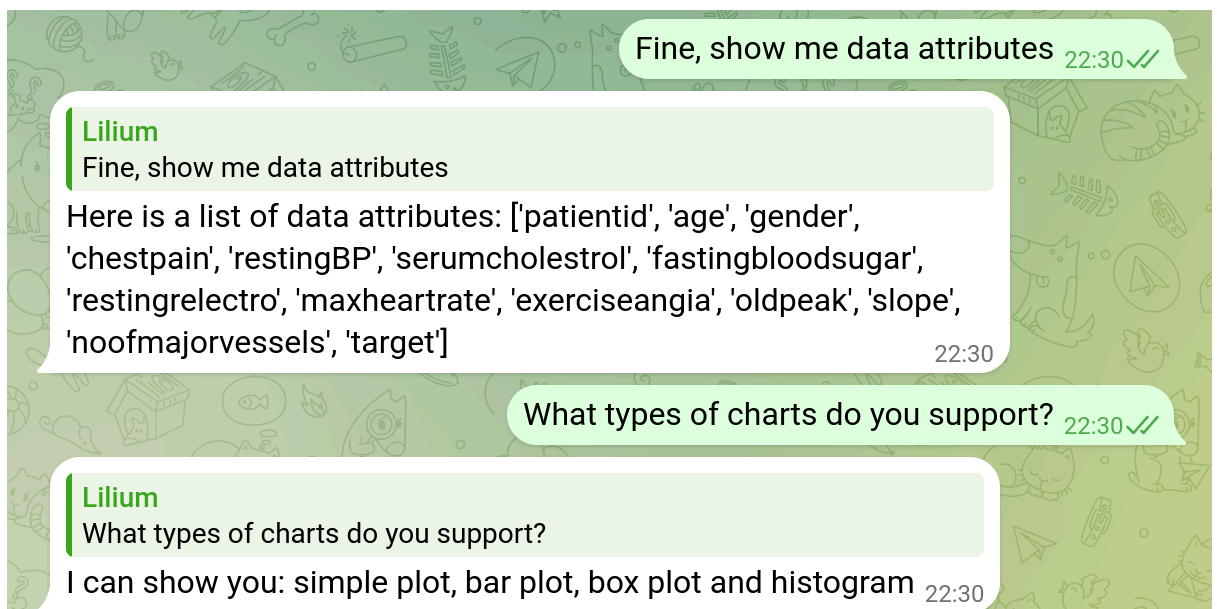


Рис. 14. Приклад показу атрибутів даних

Чат-бот дозволяє візуалізувати всі атрибути одночасно або вибрані, що є дуже зручно для перегляду. На рис. 15 зображено приклад виводу гістограм для всіх атрибутів, та обраних на рис. 16.

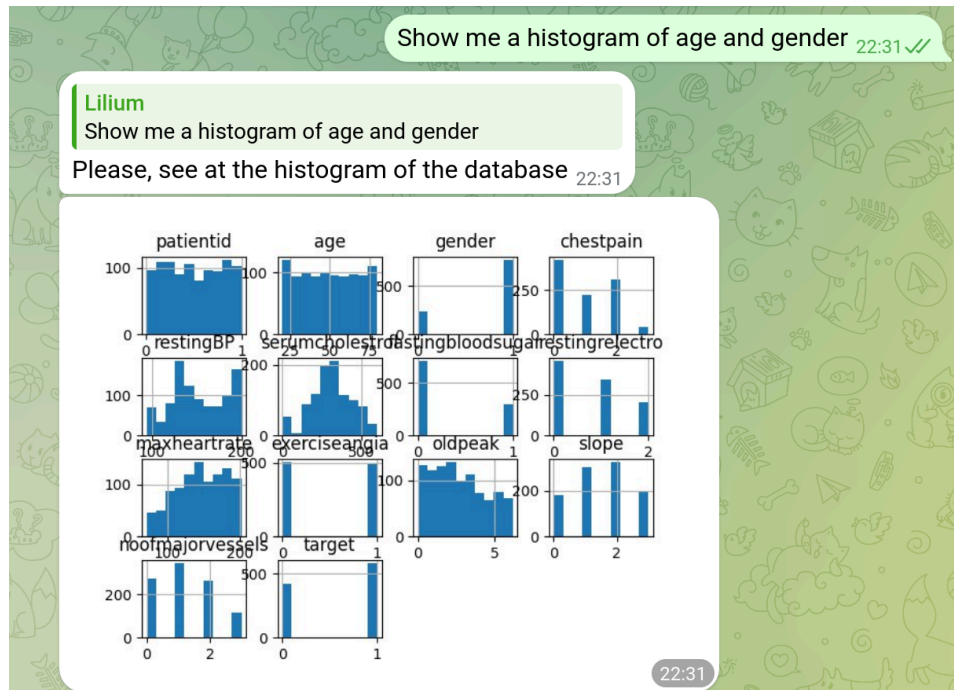


Рис. 15. Бот візуалізує гістограми для всіх атрибутів

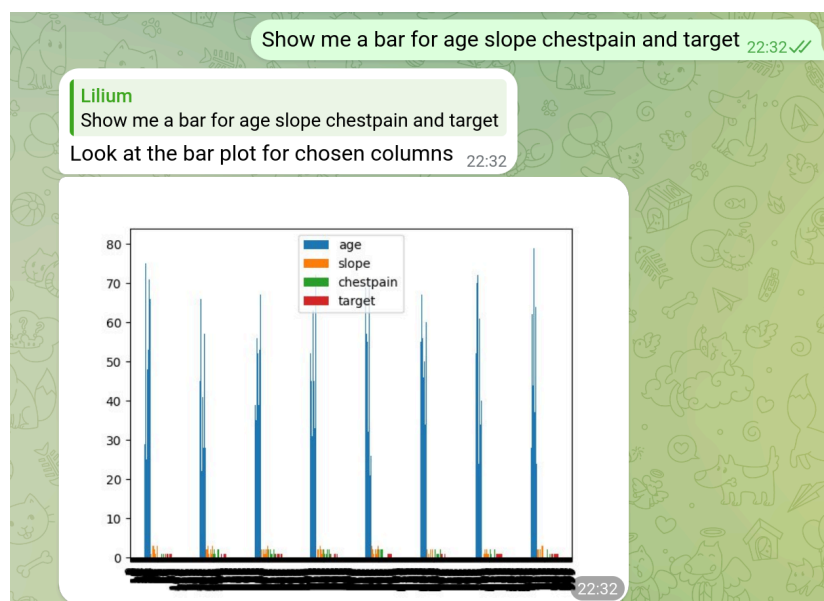


Рис. 16. Бот показує стовпчасті діаграми для вибраних атрибутів

ВИСНОВКИ

Підбиваючи підсумки, технології ШІ відіграють важливу роль у багатьох галузях. Різні процеси автоматизуються або оптимізуються завдяки розумному використанню чат-ботів, і це скоріше є відправною точкою для подальших досліджень. За допомогою нових підходів і алгоритмів можна створювати ще більш складні аналітичні системи, здатні виконувати завдання, які поки що можуть виконувати лише самі люди. Тому вчені, оптимізувавши простіші процеси, зможуть зосередитись на проектах більш високого рівня.

Проведений аналіз джерел літератури вказує на те, що обраний напрям наукових досліджень є актуальним, а тема виконання операцій аналізу даних у просторі “смислу” має великий науковий та практичний інтерес. Серед отриманих результатів маємо:

- досліджено можливість підтримки складних обчислювальних операцій над даними у задачах РАД та зазначено, що складні обчислювальні операції потрібно підтримувати засобами аналітики або кодом мовою Python, а інші складні типи операцій потрібно декомпонувати;
- досліджено можливості перетворення даних за допомогою ВММ та визначено, що ВММ гарно працюють з форматами даних, легко ідентифікують сутності, також їм доступна тектова аналітика. У разі складних операцій з перетворення даних, їх треба розділити на більш прості за можливості. У такий спосіб мережа краще виконує завдання;
- досліджено можливості ВММ для аналізу даних та визначено, що мережа у просторі “смислу” повністю розуміє завдання РАД, а також може запропонувати код мовою Python для його вирішення. Якщо уточнити підказку, то мережа не пропонує код, а намагається пояснити результати обчислень. При роботі з дійсними числами краще використовувати інші засоби для аналізу даних або ж згенеровані скрипти на Python;
- розроблено концепцію керування аналізом даних за допомогою послідовності операцій та інструкцій у підказках;

- спроектовано архітектуру та розроблено чат-бот систему для виконання РАД над табличними даними для апробації запропонованого рішення.

Основний функціонал бота полягає у його вмінні підтримувати розмову завдяки використаній ВММ, реакції на ключові слова, аналізі бази даних, наданої користувачем, зв'язку із бібліотеками Matplotlib та візуалізації даних у вигляді діаграм. Такий бот може бути потенційно корисним для тих, кому потрібно швидко створити графік “одним кліком”. Отже, він автоматизує цілу задачу, що часто виконується багатьма працівниками сфери аналізу даних, адже справжні аналітичні системи програмуються та не можуть динамічно змінювати порядок відображення тих чи інших графіків і діаграм.

Розроблений чат-бот на платформі Telegram дозволяє проводити подальші дослідження ВММ та їх можливостей в аналізі даних, використовуючи й досліджуючи різні шаблони підказок.

Враховуючи сказане вище, наступним завданням для подальших досліджень розглядаємо:

- декомпозицію складних операцій перетворення даних;
- підтримку РАД за допомогою підказок, що побудовані на основі онтологій чи баз знань;
- підтримку арифметичних операцій над дійсними числами мовою Python;
- розроблення типових шаблонів РАД для різних його цілей та завдань;
- удосконалення аналітичних чат-бот систем реального часу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jianqing Fan, Fang Han, Han Liu. Challenges of Big Data analysis: National Science Review, June 2014. - V. 1. - I. 2. - pp. 293–314. - DOI: 10.1093/nsr/nwt032
2. Pranay Ahlawat, Justin Borgman, Samuel Eden, Steven Huels, Jess Iandiorio, Amit Kumar, and Philip Zakahi. A New Architecture to Manage Data Costs and Complexity: BCG, February 7, 2023. - URL: <https://www.bcg.com/publications/2023/new-data-architectures-can-help-manage-data-costs-and-complexity?linkId=200819392>
3. Розвідковий аналіз даних: Вікіпедія. - URL: https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%B2%D1%96%D0%B4%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85
4. Chris Chatfield. Exploratory data analysis / European Journal of Operational Research, 1986. - V. 23, I. 1, pp. 5-13. - DOI: 10.1016/0377-2217(86)90209-2
5. Peter Bickel. Discussion on the paper “Sure independence screening for ultrahigh dimensional feature space” by Fan and Lv. J. Roy. Statist. Soc. Ser. B, 70(5):883–884, 2008
6. Kanit Wongsuphasawat, Yang Liu, Jeffrey Heer. Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study, 1 Nov 2019. - 10 p. - arXiv: <https://arxiv.org/pdf/1911.00568.pdf>
7. Jessica Hullman, Andrew Gelman. Challenges in Incorporating Exploratory Data Analysis Into Statistical Workflow: Harvard Data Science Review, 2021. - V.3. - I. 3. - 11 p. - DOI: 10.1162/99608f92.9d108ee6
8. Tova Milo, Amit Somech. SIGMOD '20: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, June 2020. - pp.2617-2622. - DOI: 10.1145/3318464.3383126

9. Miguel Ângelo Lellis Moreira, Claudio de Souza Rocha Junior, Diogo Ferreira de Lima Silva and others. Exploratory analysis and implementation of machine learning techniques for predictive assessment of fraud in banking systems: *Procedia Computer Science*, 2022. - V. 214. - pp. 117-124. - DOI: 10.1016/j.procs.2022.11.156.
10. Mehta V, Batra N, Poonam, Goyal S, Kaur A, Dudekula KV, Victor GJ. Machine Learning based Exploratory Data Analysis (EDA) and Diagnosis of Chronic Kidney Disease (CKD): *EAI Endorsed Transactions on Pervasive Health and Technology*, 2024. - 8 p. - DOI: 10.4108/eetpht.10.5512
11. Da Poian V, Theiling B, Clough L, McKinney B, Major J, Chen J and Hörst S. Exploratory data analysis (EDA) machine learning approaches for ocean world analog mass spectrometry, 2023. - 17 p. - DOI: 10.3389/fspas.2023.1134141
12. Inigo Martinez, Elisabeth Vilesb, Igor G Olaizolaa. *Data Science Methodologies: Current Challenges and Future Approaches*, Jan 2022. - 22 p. - arXiv: <https://arxiv.org/pdf/2106.07287.pdf>
13. Chon Ho, Yu. (2010). Exploratory data analysis in the context of data mining and resampling. *International Journal of Psychological Research*, 3(1), 9-22. - URL: <https://www.redalyc.org/pdf/2990/299023509014.pdf>
14. Parishad Behnam Ghader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, Siva Reddy. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders, 9 Apr 2024. - arXiv: <https://arxiv.org/pdf/2404.05961v1.pdf>
15. Mohamed Nejjar, Luca Zacharias, Fabian Stiehle, Ingo Weber. LLMs for Science: Usage for Code Generation and Data Analysis, 7 Dec 2023. - arXiv: <https://arxiv.org/pdf/2311.16733.pdf>
16. Tai, R. H., Bentley, L. R., Xia, X., Sitt, J. M., Fankhauser, S. C., Chicas-Mosier, A. M., & Monteith, B. G. (2024). An Examination of the Use of Large Language Models to Aid Analysis of Textual Data. *International Journal of Qualitative Methods*, 23. - DOI: 10.1177/16094069241231168

17. Jacqueline A Jansen, Artür Manukyan, Nour Al Khoury, Altuna Akalin. Leveraging large language models for data analysis automation, 2024. - 18 p. - DOI: 10.1101/2023.12.11.571140
18. Локазюк В. М. Розроблення програмного забезпечення для опрацювання текстів з діагностичною інформацією [Електронний ресурс] / В. М. Локазюк, В. Я. Ляшкевич, О. Я. Олар // Радіоелектронні і комп'ютерні системи. - 2007. - № 6. - С. 123–129. - Режим доступу: http://nbuv.gov.ua/UJRN/recs_2007_6_25
19. Christoph Schocka, Jonas Dumlerb, Prof. Dr.-Ing. Frank Doeppera. Data Acquisition and Preparation – Enabling Data Analytics Projects within Production // 54th CIRP Conference on Manufacturing Systems: Procedia CIRP, 2021. - V. 104, pp. 636-640. - DOI: 10.1016/j.procir.2021.11.107
20. Fernandes, A.A.A., Koehler, M., Konstantinou, N. et al. Data Preparation: A Technological Perspective and Review. SN COMPUT. SCI. 4, 425 (2023). - DOI: 10.1007/s42979-023-01828-8
21. Preprint of Atzmueller, M., Schmidt, A., Hollender, M. (2016) Data Preparation for Big Data Analytics: Methods & Experiences. In: Enterprise Big Data Engineering, Analytics, and Management, IGI Global. - URL: <https://www.kde.cs.uni-kassel.de/wp-content/uploads/atzmueller/paper/Atzmueller-Preprint-Big-Data-Preparation-Methods.pdf>
22. Costello, Tim & Blackshear, Lori. Prepare Your Data for Tableau: A Practical Guide to the Tableau Data Prep Tool, 2020. - DOI: 10.1007/978-1-4842-5497-4
23. F. Martínez-Plumed et al., "CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories," in IEEE Transactions on Knowledge and Data Engineering, 1 Aug 2021. - V. 33. - N 8. - pp.3048-3061, - DOI: 10.1109/TKDE.2019.2962680
24. N. Bratchell. Cluster analysis: Chemometrics and Intelligent Laboratory Systems, 1989. - V. 6. - I. 2. - pp. 105-125. - DOI: 10.1016/0169-7439(87)80054-0

25. Brady Lund, Jinxuan Ma. A review of cluster analysis techniques and their uses in library and information science research: k-means and k-medoids clustering. *Performance Measurement and Metrics*, 2021. - N 22. - pp.161-173. - DOI: 10.1108/PMM-05-2021-0026
26. Mugavin, Marie. (2008). Multidimensional Scaling: A Brief Overview. *Nursing research*. 57. 64-8. - DOI: 10.1097/01.NNR.0000280659.88760.7c
27. Tabassum, Lubna. (2020). Fundamentals of artificial intelligence and deep learning techniques. 2020
28. Hend A. Selmy, Hoda K. Mohamed, Walaa Medhat. Big data analytics deep learning techniques and applications: A survey: *Information Systems*, 2024. - V.120. - DOI: 10.1016/j.is.2023.102318
29. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN COMPUT. SCI.* 2, 420 (2021). - DOI: 10.1007/s42979-021-00815-1
30. Najafabadi, Maryam & Villanustre, Flavio & Khoshgoftaar, Taghi & Seliya, Naeem & Wald, Randall & Muharemagic, Edin. (2016). Deep Learning Techniques in Big Data Analytics. - DOI: 10.1007/978-3-319-44550-2_5
31. Carlander-Reuterfelt, Daniel & Carrera, Álvaro & Iglesias, Carlos & Araque, Oscar & Sánchez-Rada, J. Fernando & Muñoz, Sergio. (2020). JAICOB: A Data Science Chatbot. *IEEE Access*. 8. 1-1. - DOI: 10.1109/ACCESS.2020.3024795
32. Sae Eun Lee, Naan Ju, Kyu-Hye Lee. Service chatbot: Co-citation and big data analysis toward a review and research agenda: *Technological Forecasting and Social Change*, 2023. - V. 194. - DOI: 10.1016/j.techfore.2023.122722
33. Sankar, Reshmi. (2018). EMPOWERING CHATBOTS WITH BUSINESS INTELLIGENCE BY BIG DATA INTEGRATION. *International Journal of Advanced Research in Computer Science*. 9. 627-631. - DOI: 10.26483/ijarcs.v9i1.5398
34. Hamed Khosravi, Mohammad Reza Shafie, Morteza Hajiabadi, Ahmed Shoyeb Raihan, Imtiaz Ahmed. Chatbots and ChatGPT: A Bibliometric Analysis and

Systematic Review of Publications in Web of Science and Scopus Databases:
arXiv, 2023. - 30 p. - arXiv: <https://arxiv.org/pdf/2304.05436.pdf>