

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Cserpák Zalán**
Törzskönyvi száma: **T009110/FI12904/N**
Neptun kódja: 

A dolgozat címe:

Nagy Nyelvi Model alapú gyógyszer javasló rendszer
Large Language Model based Medicine Recommender System

Intézményi konzulens: **Nemes Gyula Ádám**
Külső konzulens:

Beadási határidő: **2024. december 15.**

A záróvizsga tárgyai: **Számítógép architektúrák**
Mesterséges intelligencia specializáció

A feladat

A modern technológia és a mesterséges intelligencia alkalmazása az egészségügyben új dimenziókat nyit meg a betegellátásban és a gyógyszeres kezelések terén. Ennek a projektnek a célja egy olyan innovatív rendszer kifejlesztése, amely egy nagy nyelvi modell (LLM) segítségével képes gyógyszereket javasolni a betegek által megfogalmazott panaszok alapján. A rendszer a gyógyszerértárakban dolgozó szakemberek számára nyújt támogatást, lehetővé téve számukra, hogy gyorsan és hatékonyan találjanak megfelelő gyógyszeres kezeléseket a betegek számára.

A hallgató feladata egy olyan rendszer kifejlesztése, amely képes értelmezni a természetes nyelven megadott egészségügyi panaszokat, és azok alapján gyógyszereket javasolni. A rendszernek figyelembe kell vennie a beteg által megadott tüneteket, valamint az esetleges allergiákat és egyéb egészségügyi állapotokat.

A dolgozatnak tartalmaznia kell:

- a Nagy Nyelvi Modellek áttekintő elemzését,
- a lehetséges felhasználások és megoldások vizsgálatát,
- a rendszer tervét,
- a megvalósítás dokumentációját.



.....
Dr. Eigner György
intézetigazgató

A szakdolgozat elévülésének határideje: **2026. december 15.**
(ÓE HKR 54.§ (10) bekezdés szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2024. 12. 11.

hallgató aláírása

ABSZTRAKT

Az egészségügyi rendszerekben a megfelelő gyógyszerajánlás kritikus fontosságú a páciensek gyors és hatékony kezelése érdekében. A jelenlegi módszerek, mint például a manuális keresés vagy az egyszerű szűrők, gyakran nem képesek megfelelően figyelembe venni a tünetek és allergiák összetettségét, illetve ezek kombinációját. Ezért szükség van egy olyan intelligens rendszerre, amely hatékonyan szűkíti le a releváns gyógyszerek körét és minimalizálja a hibalehetőségeket. A rendszer célja, hogy a felhasználó által megadott tünetek alapján gyorsan és hatékonyan ajánlja a legrelevánsabb gyógyszereket.

A kifejlesztett rendszer kulcsszavakat generál a tünetek és allergiák alapján, FAISS-alapú keresési folyamattal azonosítja a potenciális gyógyszereket, majd egy OpenAI-alapú rangsorolási mechanizmus segítségével tovább szűkíti az eredményeket, figyelembe véve az allergén összetevőket is. A rendszer felhasználóbarát felületet biztosít, amely a Streamlit keretrendszeren keresztül érhető el.

Az eredmények azt mutatják, hogy a rendszer képes gyorsan és pontosan azonosítani az 5 legrelevánsabb gyógyszert a megadott tünetek és allergiák alapján. Ezáltal hatékonyabbá teszi a gyógyszerkeresési folyamatot, és hozzájárul a helyes kezelési döntések meghozatalához.

ABSTRACT

In healthcare systems, accurate medication recommendation is critical for the prompt and effective treatment of patients. Current methods, such as manual searches or basic filtering, often fail to adequately account for the complexity of symptoms and allergies or their combinations. Therefore, there is a need for an intelligent system that efficiently narrows down relevant medications while minimizing errors. The goal of the system is to quickly and effectively recommend the most relevant medications based on the symptoms provided by the user.

The developed system generates keywords based on symptoms and allergies, identifies potential medications using a FAISS-based search process, and then refines the results using an OpenAI-based ranking mechanism, ensuring allergens are considered. The system features a user-friendly interface accessible via the Streamlit framework.

Results demonstrate that the system can quickly and accurately identify the top 5 most relevant medications based on specified symptoms and allergies. This enhances the medication search process and supports better decision-making in treatment selection.

TARTALOMJEGYZÉK

1	BEVEZETÉS	4
2	Rendszerterv	4
3	Chatbot?	5
3.1	Mi a chatbot?	5
3.2	Fajtái	5
3.2.1	Generatív alapú chatbotok	5
3.2.2	Társasági chatbotok	5
3.2.3	Összehasonlítása	5
3.3	Chatbotok értéke	6
3.4	Chatbotok működése	6
3.4.1	Első próbálkozás	6
3.4.2	Modern Chatbotok	7
3.4.3	Chatbotok előnyei, korlátai	7
3.4.4	Következtetés a Chatbotokról	8
4	Mesterséges Inteligencia	8
4.1	Neurális háló	9
4.2	Mélygépi tanulás	10
4.3	Mélytanulás kontra klasszikus gépi tanulási módszerek	10
4.3.1	Következtetés	12
5	Nyelvi modellek	12
5.1	A nagy nyelvi modellek alapjai	13
5.2	LLM-ek alapfogalmai	14
5.3	LLM metrikák	15
5.3.1	FAISS	16
5.3.2	Fuizz	17

5.4	Pandas	17
5.4.1	Matplotlib	18
5.4.2	Streamlit.	18
5.4.3	Tokenizáció	19
5.4.4	Transzformerek	19
5.5	A Nagy Nyelvi Modellek áttekintő elemzése	20
5.5.1	ChatGPT	20
5.5.2	Bard	20
5.5.3	Összehasonlítás	20
5.6	A ChatGPT etikai kérdései az orvostudományban és az egészségügyben	23
6	Hasonló rendszerek bemutatása	23
7	Rendszerterv a Gyógyszerajánló Alkalmazásról.	25
7.1	A rendszer céljai	25
7.2	Technológiai eszközök	25
7.2.1	Egyszerű és gyors keresés	26
7.2.2	Allergiák kezelése	27
7.2.3	Rangsorolás	27
7.2.4	Könnyű hozzáférhetőség	28
7.3	Funkcionális követelmények	28
7.3.1	Adatbevitel és kulcsszógenerálás	28
7.3.2	Keresés és szűrés	29
7.3.3	Rangsorolás és adatok megjelenítése.	30
7.4	Nem-funkcionális követelmények	32
7.5	Architektúra	34
7.5.1	Felhasználói interfész.	34
7.5.2	Backend logika	36
7.5.3	Adatbázis	36

7.6	Folyamatok részletes leírása	37
7.7	Kockázatok.	38
7.8	Nehézségek amikbe ütköztem	39
8	az Iradalomkutatásrólösszegzés	41
8.1	Magyar összefoglaló	41
8.2	Angol összefoglaló az iradalomkutatásról.	42
9	IRODALOMJEGYZÉK	43
10	ÁBRAJEGYZÉK.	45
11	TÁBLAJEGYZÉK	46
12	RÖVIDÍTÉSEK	47

1 BEVEZETÉS

Az Óbudai Egyetemen eltöltött éveim során gyakran gondolkodtam azon, hogy miért nem létezik egy olyan webes felület, amely a mesterséges intelligencia használatával segíti a gyógyszerértékesítőben dolgozó gyógyszerészek és szakasszisztensek munkáját. A program koncepcióját nővéremtől kaptam. Aki elmesélte, hogy kikerülve az egyetemről a különböző megbetegedések gyógyítására használatos hatóanyagokat ismerik ugyan, de manapság a gyógyszerpiac is telített, egy hatóanyag rengeteg márkanév alatt futhat. Ez a felület megkönnyíthatná a munkájukat, főleg a betanulási időszakban, amikor a különböző hatóanyag tartalmú készítményeket még nem tudják a megfelelő márkanévhez kötni. Kezdetben a weboldal bekérné a gyógyszerértékesítőbe érkező betegek tüneteit, majd pontosítás után felsorolná az öt leghatékonyabb vény nélküli készítményt és leggyakrabban eladott vény nélküli készítményt. A weboldallal rengeteg időt meg lehetne spórolni, továbbá csökkenteni lehetne a sorban állás idejét a vásárlók számára.

2 RENDSZERTERV

A rendszer mélygépi tanuláson fog alapulni, amit egy LLM alapú rendszerrel fogok megvalósítani. Kezdetben megadom a rendszernek, hogy egy gyógyszerértékesítő asszisztensként funkcionáljon. Ezt követően, amikor a beteg elmondja a panaszait, a rendszer legenerálja a szöveget, majd megkapjuk az asszisztens választ. Ezután a program további információkat kér, amelyekre a felhasználó ismét válaszol, és a megadott információkkal újra futtatjuk a modellt. A modelleddig csak a kontextust és a saját súlyait ismeri. Ekkor fogjuk megadni neki az adatbázist, amelyből válaszolni fog. Az adatbázist egy Python könyvtárba töltöm be, amit a kontextusrészbe illesztünk. Innentől a modell ismerni fogja a gyógyszereket és azok összes tulajdonságát. Az adatbázist finomítani is kell. Két megoldást fontoltam meg: az első, ha két gyógyszer összetevői és tulajdonságai ugyanazok, akkor ár szerint lehetne rendezni őket, ezzel csökkentve az adatbázis méretét, a második megoldás lényege, hogy ha egy felhasználó egy bizonyos panasszal keres meg minket, például, hogy lázas, akkor arra a tartalomra keresek. Végül ezt a megoldást találtam hatékonyabbnak, szóval emellett döntöttem, hogy az eredményhez fog vezetni, mint az első.

Ehhez a programhoz egy gyógyszerértékesítő adatbázisát fogom használni, amit a későbbiekben frissíteni fogok, hogy naprakészen tartsam a programot.

3 CHATBOT?

3.1 Mi a chatbot?

A chatbot egy számítógépes alkalmazás, amely emberi párbeszédet imitál a felhasználóval. Nem minden ilyen program rendelkezik mesterséges intelligencia képességekkel, azonban a korszerű chatbotok egyre inkább alkalmazzák a beszélgetési AI technológiákat, mint például a természetes nyelvfeldolgozást (NLP), annak érdekében, hogy értelmezzék a felhasználók kérdéseit és automatizálják a válaszadást.

3.2 Fajtái

3.2.1 Generatív alapú chatbotok

A következő generációs, generatív mesterséges intelligenciával felszerelt chatbotok még fejlettebb funkcionalitást nyújtanak majd. Ezek a rendszerek kiválóan értik majd a köznyelvet és a bonyolult kérdéseket, képesek lesznek alkalmazkodni a felhasználók beszélgetési stílusához, valamint empátiával válaszolni a felhasználói kérdésekre. Egy vállalati szintű mesterséges intelligencia rendszer lehetővé teszi a cégek számára az önkiszolgáló folyamatok automatizálását és hozzájárul a kivételes felhasználói élmények gyorsabb kialakításához.

3.2.2 Társasági chatbotok

A beszélgetési mesterséges intelligencia olyan technológiákat foglal magában, amelyek képesek a beszéd és szöveges bevitel felismerésére, valamint az arra való válaszadásra. Az ügyfélszolgálat területén ezeket a technológiákat arra használják, hogy vásárlókkal emberhez hasonló módon kommunikáljanak. Az interakciók megtörténhetnek egy online chatboton keresztül vagy egy telefonos hangasszisztens segítségével. A beszélgetési AI a nagy mennyiségű képzési adatokat használja fel a mélytanulási algoritmusokkal annak érdekében, hogy pontosabban azonosítsa a felhasználói szándékokat és finomítsa a gépek emberi nyelv értését.

3.2.3 Összehasonlítása

Míg a beszélgetési AI chatbotok képesek feldolgozni a felhasználók kérdéseit és megjegyzéseit, majd emberszerű válaszokat adni, addig a generatív AI chatbotok tovább lépnek ezen a területen azzal, hogy új tartalmat hoznak létre válaszként. Ez az új tartalom

magas minőségű szövegekből, képekből és hangokból állhat, amelyeket a nagyméretű nyelvi modellek (LLM-ek) tanításával állítanak elő. A generatív AI-vel felszerelt chatbot felületek képesek felismerésre, összegzésre, fordításra, előrejelzésre és új tartalom létrehozására a felhasználók kérdéseire válaszolva, emberi beavatkozás nélkül.

Az öntanuló, vállalati szintű generatív AI chatbotok, amelyek beszélgetési AI platformokon alapulnak, folyamatosan fejlődnek és automatikusan tanulnak. Ezek az algoritmusok képesek a múltbeli interakciók alapján automatikusan megtanulni, hogyan reagálnak a leghatékonyabban a felmerülő kérdésekre és hogyan irányítsák a beszélgetési folyamatokat.

3.3 Chatbotok értéke

Chatbotok jelentős előnyöket kínálnak az információkeresés terén, hiszen azonnali választ biztosítanak a felhasználók kérdéseire, szöveges vagy hangalapú formában, manuális beavatkozás nélkül. A technológia széles körben elterjedt, az otthoni intelligens eszközöktől kezdve a munkahelyi kommunikációs platformokig, mint például a Slack.

A fejlett AI chatbotok, vagy más néven intelligens virtuális asszisztensek, mint például Siri, Alexa, és ChatGPT, nem csak a beszélgetéseket értik meg, hanem képesek releváns feladatokat is automatizálni. Ezen eszközök vállalati környezetben is egyre gyakoribbak, ahol növelik az ügyfél- és alkalmazotti elégedettséget.

A chatbotok integrálása a meglévő rendszerekkel, mint a Microsoft Teams, lehetővé teszi a munkafolyamatok optimalizálását és egy központi, együttműködésre kész platform kialakítását. Ezek a rendszerek képesek a CRM rendszerekkel szorosan integrálva kezelni a műveleteket, mint például a jelszómódosításokat, és elemezni a kommunikációs adatokat, hogy mélyebb betekintést nyújtsanak az ügyfél-interakciókba.

A mesterséges intelligencia chatbotjai kulcsszerepet játszanak a modern ügyfélszolgálat és marketingstratégiákban, hiszen folyamatos elérhetőségükkel és a felhasználói adatokból nyert betekintésekkel segítik a vállalkozásokat abban, hogy hatékonyabb, személyre szabott kommunikációt folytassanak az ügyfeleikkel.

3.4 Chatbotok működése

3.4.1 Első próbálkozás

1966-ban Joseph Weizenbaum, az MIT professzora létrehozta az első chatbotot, amely egy pszichoterapeuta szerepét töltötte be. A felhasználók egy elektromos írógépen keresztül üzenetet küldtek egy nagyszámítógéphez, és rövid időn belül a „pszichoterapeuta”

chatbot válaszolt nekik. Ami így nézett ki.

- User: Men are all alike.
- Chatbot: IN WHAT WAY
- User: They're always bugging us about something or other.
- Chatbot: CAN YOU THINK OF A SPECIFIC EXAMPLE
- User: Well, my boyfriend made me come here.
- Chatbot: YOUR BOYFRIEND MADE YOU COME HERE
- User: He says I'm depressed much of the time.
- Chatbot: I AM SORRY TO HEAR YOU ARE DEPRESSED

A szoftver viszonylag egyszerű működési elven alapult. Elemzésbe vette a felhasználótól érkező bemenetet, és egy előre definiált szabályrendszer segítségével generált megfelelő válaszokat.

3.4.2 Modern Chatbotok

Modern AI chatbotok a természetes nyelv feldolgozás technológiáját (NLP) alkalmazzák a felhasználók által szabadon megfogalmazott bevitel elemzésére, akár gépelési hibák, akár fordítási problémák kezelésére. Ezek az okos eszközök az elemzést követően azonosítják a felhasználó szándékait, és a beszélgetési AI segítségével formálják meg a helyes válaszokat. Ezek a technológiák egyaránt használják a gépi és a mélytanulási módszereket, melyek bár különbségeket mutatnak, mégis együttműködnek a felhasználói interakciókból épülő részletes tudásbázis létrehozásában. Ez a technológiai előrehaladás, amely a legfrissebb nagy nyelvi modellekkel (LLM) dolgozik, javította a fogyasztói megelégedettséget és támogatta a multifunkcionális chatbot rendszerek fejlesztését. Az AI chatbotok kifejlesztéséhez szükséges időtartam változó lehet, attól függően, hogy milyen technológiai eszközkészletet és fejlesztői platformokat alkalmaznak, milyen bonyolultságú a chatbot, milyen funkciókat kell teljesítenie, mennyire hozzáférhetőek az adatok, valamint hogy szükséges-e más rendszerekkel, adatbázisokkal vagy platformokkal történő integráció. Egy felhasználóbarát, kód nélküli vagy alacsony kódigényű platform segítségével az AI chatbotokat még gyorsabban lehet összeállítani.

3.4.3 Chatbotok előnyei, korlátai

Egy chatbot előnyei hátrányossá válhatnak, ha nem megfelelő platformot, programozást vagy adatforrásokat használnak. Hagyományos AI chatbotok képesek gyors ügyfélszolgálatot nyújtani, de ezeknek korlátai vannak. Sok esetben ezek a rendszerek szabályokon alapulnak, amelyek automatizálják a műveleteket és standard válaszokat adnak a felhasználók kérdéseire.

A korszerűbb, generatív mesterséges intelligencia alapú chatbotok biztonsági kihívásokkal járhatnak, beleértve az adatvesztés veszélyét, a bizalmas információk védelmének és felelősségvállalásának hiányát, a szellemi tulajdonjoggal kapcsolatos összetettségeket, a forrásadatok nem megfelelő engedélyeztetését, valamint a magánélet és a nemzetközi jogszabályoknak való megfelelés bizonytalanságát. Ha nincsenek megfelelő bemeneti adatok, előfordulhatnak „hallucinációk”, azaz pontatlan vagy irreleváns válaszok, ami arra késztetheti az ügyfelet, hogy a párbeszédet más kommunikációs csatornára irányítsa.

A biztonsági kockázatokat növeli, ha érzékeny adatokat, legyen az harmadik féltől származó vagy vállalati belső információ, beépítenek egy generatív AI chatbotba. Ez a chatbot adatmodelljének részévé válhat, és mások által feltett releváns kérdésekre adott válaszokban megosztható, ami adatszivárgáshoz vezethet és megsérti a vállalati biztonsági előírásokat.

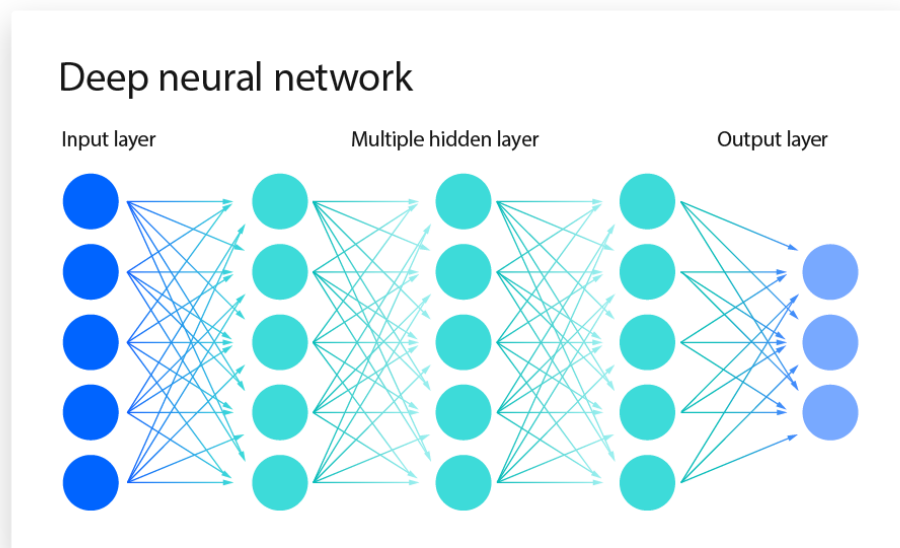
3.4.4 Következtetés a Chatbotokról

Chatbotok fontos szereplőivé váltak az interaktív technológiának, széles körben alkalmazhatók. A chatbotokat széleskörűen lehet alkalmazni számos platformon és különböző szektorokban, mint például az e-kereskedelem, az egészségügy vagy az oktatás, ahol elengedhetetlen a gyors és hatékony kommunikáció. Lehetővé teszik a rutin feladatok automatizálását, felszabadítva ezzel a vállalatok erőforrásait más, értékteremtő tevékenységek számára. Összességében a chatbotok jelentős lehetőségeket kínálnak, de fontos, hogy folyamatosan figyelemmel kísérjük és kezeljük a velük kapcsolatos kockázatokat és kihívásokat.

4 MESTERSÉGES INTELIGENCIA

Az AI definíciója olyan technológia, amely lehetővé teszi a számítógépek és gépek számára, hogy szimulálják az emberi intelligenciát és a problémamegoldó képességeket.. Alan Turing a "Számítógépek és intelligencia" című munkájában ajánlotta a közismert Turing-tesztet, amely szerint egy gépet akkor tekinthetünk intelligensnek, ha egy semleges szemlélő nem tudja megkülönböztetni a beszélgetés során egy emberi partnerétől. A modern értelemben vett mesterséges általános intelligencia azt a képességet jelenti, hogy egy gép az emberhez hasonlóan kommunikáljon, gondolkodjon és önállóan cselekedjen ismerős és új helyzetekben egyaránt. Ez a definíció jóval túlmutat a jelenleg alkalmazott módszereken, és nem ezekre gondolnak, amikor általában az „AI” kifejezést használják. Napjainkban a legtöbb mesterséges intelligencia említése gyakran felcserélhető a „gépi tanulás” vagy a „mély tanulás” kifejezésekkel, ahol az utóbbi a gépi tanulás egy speciális változata. A gépi tanulás olyan algoritmusokra és statisztikai modellekre utal, amelyek címkézett tanulási adatokból képesek mintázatokat azonosítani és következtetéseket le-

vonni.[1] [2]



4.1. ábra: Gépi tanulás edzése címkézett adatokon, teljesítményértékeléssel.
[2]

funkcionalitás szerint csoportosítás:

Reaktív gép	Az AI legelső és legegyszerűbb típusai.
Tanulni képes gép	Memóriájuk van, így felhasználják a múltat a döntéshez.
Prediktív gép	Fejlesztés alatt áll, korlátozott, alulértékeli az emberit.
Öntudatos mesterséges intelligencia	Hipotetikus végső AI, érzelmekkel, potenciálisan veszélyes.

4.1. táblázat: Mesterséges intelligencia típusai

4.1 Neurális háló

Neurális hálózatokat (NN-eket) sok más néven is ismerik, mint például kapcsolatokon alapuló modellek, neuromorf rendszerek, és párhuzamosan elosztott rendszerek, valamint mesterséges neurális hálózatok, amelyek megkülönböztetik őket a biológiai hálózatoktól. Ezek tartalmazznak számos sűrűn összekapcsolt elemet, amelyeket neuronoknak vagy csomópontoknak neveznek, és ezek csupán nemlineáris számítási elemek. Egyetlen csomópont működik mint az őt érő súlyozott bemenetek integrátora. Amint az eredmény elkészül, továbbítódik más csomópontokhoz a szinapszisoknak nevezett kapcsolatokon keresztül. Minden csomópontot jellemez egy paraméter, amit küszöbértéknek vagy eltolásnak neveznek, valamint a nemlinearitás típusa, amelyen keresztül az összes bemenet

összege áthalad. Tipikus nemlinearitások közé tartozik a kemény korlátozó (hardlimiter), a lejtő (küszöb logikai elem), és a széles körben használt szigmoid.

A legkisebb NN egy egyszerű rétegű perceptron, amely egy egyszerű hálózat, amely eldöntheti, hogy a bemenet egyik vagy másik lehetséges osztályba tartozik-e.[3] [4]

4.2 Mélygépi tanulás

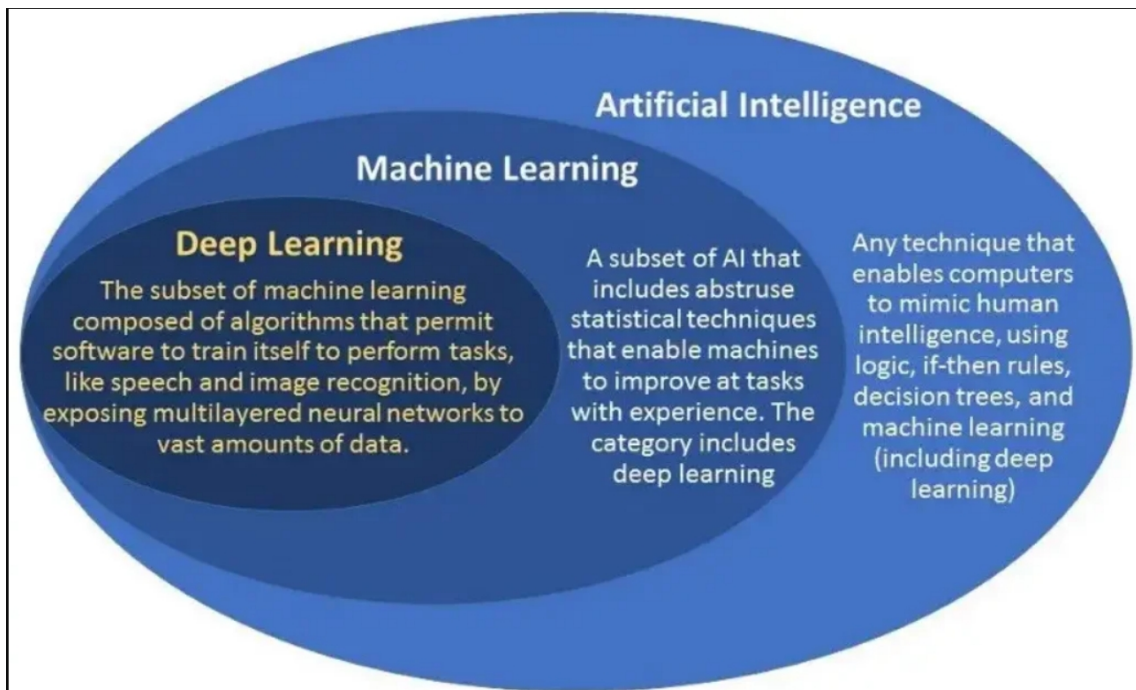
A mélytanulás forradalmasította a számítógépes modellek adatfeldolgozását, lehetővé téve azok számára, hogy bonyolultabb absztrakciókat tanuljanak meg több rétegű feldolgozás révén. Ez a technológia kiemelkedő eredményeket ért el különböző területeken, mint például a beszédfelismerés, a tárgyfelismerés, a gyógyszerkutatás és a genomika. A mélytanulás a backpropagation algoritmus segítségével finomítja a modell belső paramétereit, amely lehetővé teszi az adatok rejtett struktúráinak feltárását.

A visszacsatolt neurális hálózatok (RNN-ek) kiválóan alkalmasak szekvenciális adatok, mint például szövegek és beszéd feldolgozására, mivel képesek figyelembe venni az adatok időbeli sorrendjét és összefüggéseit. Bár a mély neurális hálózatok, amelyek több rejtett réteget tartalmaznak, már régóta ismertek, hatékony alkalmazásukat korábban gátolta a megfelelő számítástechnikai kapacitás hiánya. Az utóbbi években azonban a hardveres fejlődés, különösen a GPU-k teljesítményének növekedése, lehetővé tette ezen modellek hatékony képzését és alkalmazását. Ennek köszönhetően a mélytanulás napjaink egyik legfontosabb és legígéretesebb technológiájává vált a mesterséges intelligencia területén.[2]

4.3 Mélytanulás kontra klasszikus gépi tanulási módszerek

- Előnyök:
 - Adatkezelési rugalmasság: A mélytanulási algoritmusok közvetlenül a nyers adatokból képesek tanulni, ezzel csökkentve a tulajdonságok előzetes kiválasztásának igényét.
 - Fokozott teljesítmény: A mélytanulás a képfeldolgozás és a hangfeldolgozás területén jelentősen túlszárnyalja a klasszikus gépi tanulási módszereket.
 - Általánosíthatóság: A neurális hálózatok általános modellként szolgálnak, melyeket számos különböző feladatra lehet használni.
- Hátrányok:
 - Nagy adatigény: A hatékony működés érdekében szükség van bő adاتمennyiségre, habár az adatelőkészítés továbbra is alapvető lehet.
 - Magas számítási költségek: A mélytanulási modellek működtetése számottevő számítástechnikai kapacitást igényel, amely jelentősen meghaladja a klasszi-

- kus módszerek igényeit.
- Átláthatósági problémák: A mélytanulási modellek gyakran "fekete doboz" módon működnek, ami ellentétben áll a klasszikus módszerekkel, ahol a modellek és azok összefüggései könnyebben megérthetők.
 - További megfontolások:
 - Modellezési pontosság: A mélytanulás, különösen a konvolúciós és rekurens neurális hálózatok alkalmazásával, képes összetett minták és időbeli sorozatok modellezésére, ami számos új felhasználási területet nyit meg.
 - Adaptív képesség: A mélytanulási modellek rugalmasan igazodnak az új adatokhoz és a tanulási folyamat során folyamatosan pontosítják a modell paramétereit, ezáltal növelve az előrejelzések pontosságát.
 - Interdiszciplináris alkalmazások: A mélytanulás széleskörű alkalmazása az egészségügytől a pénzügyi ágazaton át a gyártástechnológiáig terjed, ahol hozzájárul a döntéshozatali folyamatok automatizálásához és az operációs hatékonyság javításához.



4.2. ábra: Mélytanulás kontra klasszikus gépi tanulási

[5]

4.3.1 Következtetés

A mélytanulás képessége arra, hogy közvetlenül a nyers adatokból tanuljon, csökkenti a jellemzők előzetes megalkotásának szükségességét, különösen a képek és hangok összetett adatstruktúrái esetében. Habár ez jelentős előnnyel jár, az eredményes tanuláshoz szükséges adatmennyiség korlátozó tényezővé válhat. A mélytanulási modellek magas számítási igénye további kihívást jelent, mivel ezek a modellek lényegesen több erőforrást igényelnek, mint a hagyományos gépi tanulási technikák. Ennek eredményeképpen, bár a mélytanulás bizonyos területeken kimagasló teljesítményt nyújt, költsége és összetettsége korlátozhatja annak széleskörű alkalmazását.

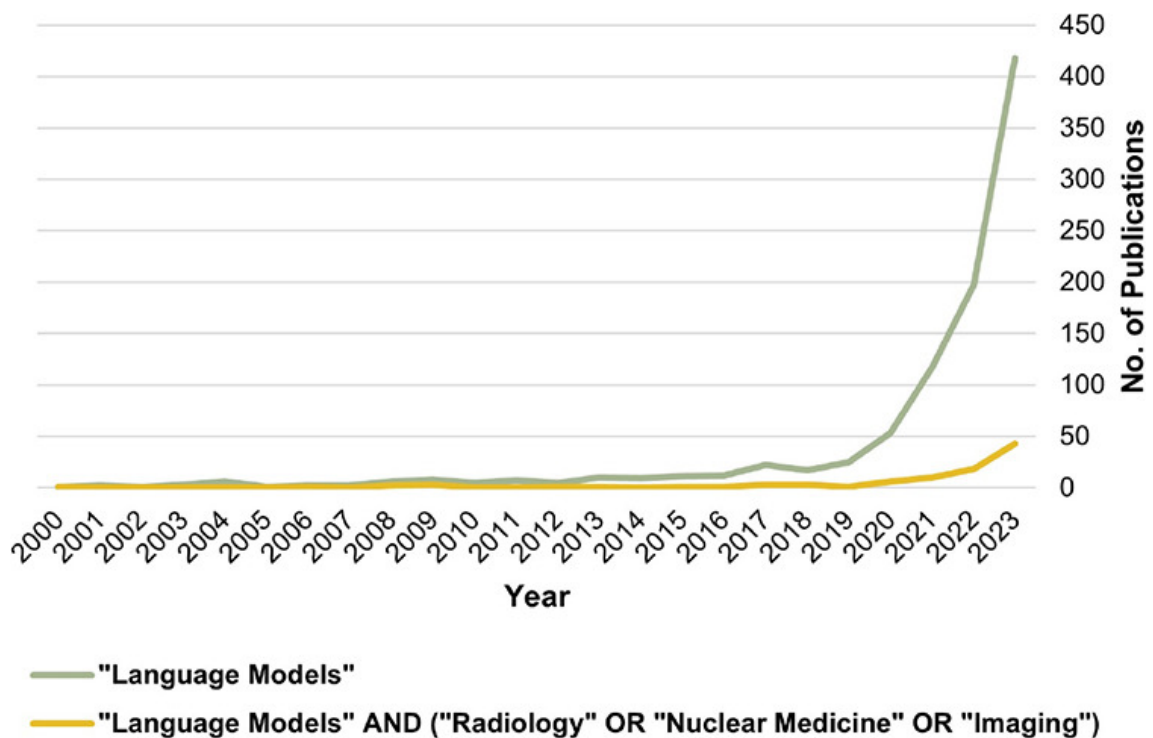
5 NYELVI MODELLEK

A mesterséges intelligencia terén a nagy nyelvi modellek (LLM) megjelenése forradalmasította az orvostudományt és különösen a radiológiát, ami mostanra egyre kézzelfoghatóbbá vált. Nap mint nap újabb tanulmányok kerülnek publikálásra, amelyek ezeket a modelleket alkalmazzák a radiológiai gyakorlatban. Ahhoz, hogy a radiológusok biztonságosan és hatékonyan alkalmazhassák ezt az új technológiát, alaposan meg kell ismerniük a lényeges fogalmakat, megérteniük a műszaki hátteret, és tisztában kell lenniük a potenciális kockázatokkal és etikai kérdésekkel. Alapvetően a nyelvi modell egy szoftver, amelyet az emberi nyelv feldolgozására fejlesztettek ki, méretük és bonyolultságuk a kisebb, szabályalapú rendszerektől a fejlett, AI-alapú modellekig terjed. Ezzel szemben az LLM-ek, mint a nyelvi modellek kiemelkedő csoportja, méretükkel, bonyolultságukkal és olyan egyedülálló képességeikkel különböznek meg, amelyeket kisebb méretű társaik nem rendelkeznek. Ezek a mélytanulási architektúrákra épülő, óriási, több milliárd paraméterrel rendelkező modellek széles körben kiváló teljesítményt nyújtanak különféle NLP feladatokban, mint az összefoglalás, fordítás, hangulatanalízis és szövegalkotás. Egyszerűsítve, az LLM-ek előrejelezik a következő szót vagy tokent egy adott szövegsorozatban.[6]

Az első nyelvi modellek között szerepelt az 1960-as években fejlesztett ELIZA, egy korai „chatbot”, amely előre definiált szabályokra és mintaillesztésre támaszkodott az emberi párbeszéd szimulációjához. Bár ELIZA és társai képességei korlátozottak voltak, és nehezen birkóztak meg az emberi nyelv bonyolultságával és finomságaival, ezek a korai modellek elindították a kutatásokat és fokozták az érdeklődést a természetes nyelvfeldolgozás (NLP) iránt. [7]

Az áttörés a nagy nyelvi modellekben az 1990-es években történt meg, az internet megjelenésével és a fejlett számítástechnikai képességekkel, amelyek lehetővé tették a bővülő szöveges adatbázisok elérhetőségét a képzéshez. Kiemelten, a hosszú rövid távú memória (LSTM) hálózat 1997-es bevezetése mérföldkőnek számít a modern nagy nyelvi modellek elődei számára. A technológiai haladás további lendületet vett, és 2017-ben

csúcsosodott ki az „Attention Is All You Need” című úttörő tanulmányban, amely bemutatta a transzformátoros hálózatokat. Ezt követően, 2018-ban a generatív előképzett transzformátor (GPT) és a transzformátorokon alapuló kétirányú kódoló reprezentációk (BERT) megjelenése jelentős fordulópontot jelentett a nyelvfeldolgozás terén, és bevezette a nagy nyelvi modellek korát. Ettől kezdve a nagy nyelvi modellek folyamatosan bővültek, és egyre nagyobb népszerűsége tettek szert mind a nagyközönség, mind az akadémiai közösség körében.[8]



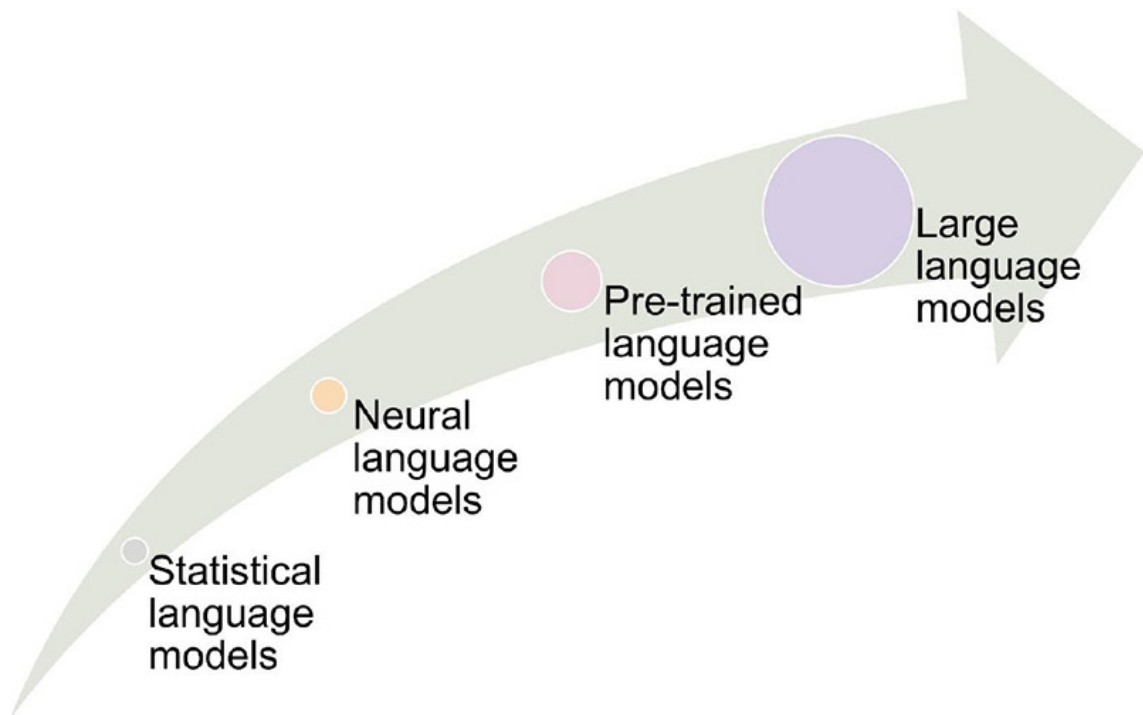
5.1. ábra: LLM-ek elterjedésének grafikonja

[9]

5.1 A nagy nyelvi modellek alapjai

A nyelvmodellálás technikai fejlődése három fő szakaszra bontható: statisztikai nyelvmodellek, neurális nyelvmodellek. Az utóbbiakat előzetesen tanítják be felügyelet nélküli módszerekkel (azaz címkézetlen adatokból tanulnak) nagy mennyiségű szöveges anyagon, és sokféle feladatban alkalmazhatóak anélkül, hogy alapjától újra kellene őket képezni. A zero-shot és few-shot tanulási képességek révén a PLM-ek képesek az általánosításra és az új kihívásokhoz való alkalmazkodásra további tanulás nélkül, vagy csak minimális további képzéssel. A kutatások szerint a PLM-ek méretezése az adatmennyiség vagy a modell méretének növelésével gyakran fokozza a modell hatékonyságát az

adott feladatokban. Ezek a nagy méretű PLM-ek meglepően eltérő viselkedést mutatnak a kisebb méretű modellekhez képest, és előtérbe kerülnek különféle összetett feladatok megoldásában, mint például a kontextusfüggő tanulás, az utasítások követése, és a lépésről lépésre való következtetés. Ezek a nagyobb PLM-ek kontextusfüggő tanulással képesek a kívánt eredmények előállítására további képzés vagy gradiens-frissítések nélkül, és képesek új feladatokhoz utasításokat adni explicit példák nélkül. Ezt figyelembe véve a tudományos közösség megalkotta az LLM (nagy nyelvi modellek) kifejezést ezekre a több százmilliárd paramétert tartalmazó óriás PLM-ekre.[10] [11]



5.2. ábra: Nyelvi modellek technikai fejlesztési szakaszai.

[9]

5.2 LLM-ek alapfogalmai

Az LLM-ek általában transzformátor-alapú architektúrákat használnak, amelyek számítástechnikai szempontból jelentősen párhuzamosíthatók. A transzformátorok alapvetően kódolókból és dekódolókból állnak, melyek mindegyike egyedi figyelmi mechanizmussal bír. Ez a figyelmi mechanizmus lényegében egy skalárszorzat művelet, amely hasonlósági értékeket állít elő, lehetővé téve a modell számára, hogy bizonyos bemenetekre nagyobb figyelmet fordítson másokkal szemben, függetlenül attól, hogy hol helyezkednek el a bemeneti sorozatban. Ezáltal a modell képes jobban megérteni a környezetét. Ezenkívül, a visszatérő neurális hálózatokkal ellentétben, a figyelmi mechanizmus lehetővé teszi a mo-

dell számára, hogy a teljes mondatot vagy akár egy teljes bekezdést egyszerre szemléljen, nem csak egyes szavakat külön-külön.

5.3 LLM metrikák

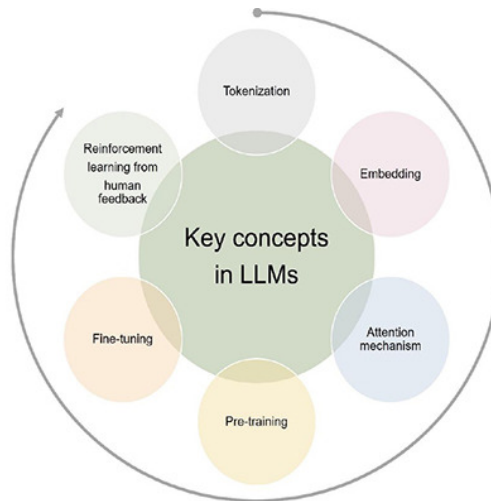
A természetes nyelvfeldolgozás a nagy nyelvi modellek sikerének köszönhetően jelentős áttörést ért el. Az LLM-ek jelentős figyelmet kaptak a tudományos körökben és az iparban, köszönhetően sokoldalú alkalmazásaiknak a szöveggenerálás, a kérdések megválaszolása és a szövegösszegzés terén. Ahogy az NLP területe fejlődik, egyre több, különféle technikákat alkalmazó és specifikus korpuszokon képzett, terület-specifikus LLM-ek jelennek meg, e modellek teljesítményének értékelése kulcsfontosságúvá válik. A teljesítmény méréséhez elengedhetetlen, hogy átfogó képet kapjunk a meglévő mutatókról. Az értékelési folyamat során az LLM-ek teljesítményét számszerűsítő metrikák kiemelt szerepet játszanak. [12]

Teljesítmény értékelési metrikák:

- Pontosság
- Precizitás
- Visszahívás
- F1-Score
- ROUGE
- BLEU

Statisztikai Értelmezés:

- Eloszlások
- Variancia
- Konfidencia Intervallumok



5.3. ábra: Kulcsfogalmak az LLM-ekben.

[9]

5.3.1 FAISS

A gépi tanulás kulcsfontosságú szerepet játszik számos valós probléma megoldásában, különösen az osztályozási és klaszterezési feladatok terén. A K-legközelebbi szomszéd (KNN) algoritmus, amelyet széles körben alkalmaznak szövegosztályozásban, adatbányászatban és prediktív elemzésben, nagy számítási teljesítményt igényel. A big data korszakában a modern adatközpontok közelítő technikákkal használják ezt az algoritmust, hogy a napi nagy terhelésű munkafolyamatokat hatékonyabban kezeljék. Ennek ellenére a nagy dimenziójú adathalmazokon végzett lekérdezések továbbra is számítási és energiafogyasztási kihívásokat jelentenek.

Ebben a tanulmányban bemutatunk egy új megközelítést, amely egy hardveresen gyorsított, közelítő KNN algoritmust alkalmaz a FAISS keretrendszerben, FPGA-OpenCL platformokon a felhőben. Ez az első olyan megoldás, amely az FPGA architektúrát integrálja ebbe a keretrendszerbe, és demonstrálja, hogyan csökkenthetők az indexépítési idők nagy méretű bemenetek esetén, miközben a rendszer teljesítménye meghaladja a hagyományos nagy teljesítményű rendszerekét. Az AWS cloud F1 példányokon végzett kísérletek során a megoldás 98-szoros gyorsítást ért el egy szál CPU-hoz képest, és 2,1-szeres sebességnövekedést mutatott egy 36 szál Xeon CPU-val szemben. Emellett a rendszer teljesítmény/watt aránya 3,5-szörös volt a CPU-hoz képest, és 1,2-szeres egy Kepler osztályú GPU-hoz viszonyítva.[13]

5.3.2 Fuizz

A fuzzy logika alkalmazása az adatok lekérdezésében és a természetes nyelvfeldolgozásban (NLP) az utóbbi években jelentős érdeklődést váltott ki. Zadeh (1965) bevezette a fuzzy halmazok fogalmát, amely az adatok pontatlanságának és bizonytalanságának kezelésére szolgált. Ezen az alapon Bezdek (1981) olyan fuzzy klaszterezési algoritmusokat dolgozott ki, amelyek számos területen elterjedtek.

Smith et al. (2021) és Johnson (2022) tanulmányai igazolták a fuzzy keresési technikák hatékonyságát az egészségügyi adatok kezelésében. Smith et al. kiemelte a megközelítő egyezés-algoritmusok, például a Levenshtein-távolság előnyeit, amelyek javították az elektronikus egészségügyi nyilvántartások keresési pontosságát. Johnson kiterjesztette ezt, és a fuzzy logikát alkalmazta a gyógyszerészeti adatbázisokban, hangsúlyozva az adatok megbízhatóságának és elérhetőségének javulását.

Wang et al. (2019) az NLP és a fuzzy logika integrációját vizsgálta strukturálatlan adatok kezelésére az ellátási lánc területén. Eredményeik szerint ezek kombinálása jelentősen növeli az adatlekérdezés hatékonyságát.

Mindezek ellenére a fuzzy logika gyógyszerészeti ellátási láncban történő alkalmazásáról kevés kutatás készült. Ez a tanulmány ezen a területen kíván hozzájárulni, fuzzy keresési technikák bevezetésével optimalizálva az adatlekérdezési folyamatokat, elősegítve a jobb készletkezelést és döntéshozatalt.[14]

5.4 Pandas

A Pandas egy Python-alapú adatkezelő könyvtár, amely kifejezetten hatékony eszközt biztosít strukturált adatok kezelésére és elemzésére. Leggyakrabban táblázatos adatokkal dolgozik, mint például CSV, Excel vagy SQL-adatbázisokból származó adatok, és lehetővé teszi azok könnyű betöltését, manipulálását és elemzését. Két fő adatszerkezettel dolgozik: a DataFrame, amely táblázatos adatokat reprezentál sorokkal és oszlopokkal, valamint a Series, amely egy egydimenziós adatsor. A Pandas rendkívül rugalmas és erős szűrési, rendezési és aggregálási funkciókat kínál, így ideális adat-előkészítéshez és elemzési feladatokhoz. Különösen hasznos nagy adathalmazok kezelésében, mivel optimalizált adatfeldolgozási műveleteket biztosít. Emellett könnyen integrálható más Python-könyvtárakkal, például a NumPy-val vagy a Matplotlib-pel, így egy adatvezérelt projektnél alapvető eszköz. A Pandas támogatja a hiányzó adatok kezelését és tisztítását, ami kritikus az elemzések pontosságának biztosításához. Összességében a Pandas a Python egyik legnépszerűbb és legszélesebb körben használt könyvtára, amely jelentősen egyszerűsíti az adatokkal végzett munkát.[15]

5.4.1 Matplotlib

A Matplotlib egy Python-alapú eszköz, amely kiváló minőségű, 2D-s grafikonok készítésére alkalmas. Támogatja az interaktív és nem interaktív ábrázolást, valamint lehetőséget nyújt a képek különböző formátumokban történő mentésére, például PNG-ben vagy PS-ben. Számos grafikus felületet képes használni (például GTK+, wxWidgets, Qt), és változatos diagramtípusokat kínál, mint például vonaldiagramok, oszlopdiagramok, kördiagramok, hisztogramok és még sok más. Különösen nagy előnye, hogy könnyen használható, testreszabható és rugalmas.

A Matplotlib sokoldalúsága lehetővé teszi, hogy interaktív környezetekben és grafikus megjelenítést nem igénylő szkriptekben egyaránt használható legyen. Használható beágyazottan grafikus alkalmazásokban vagy weboldalakon, de akár interaktív módon is a Python tolmács vagy az IPython segítségével. [16]

5.4.2 Streamlit

A Streamlit egy Python-alapú, nyílt forráskódú keretrendszer, amely egyszerűsíti az interaktív webalkalmazások készítését, különösen az adatelemzéssel és gépi tanulással foglalkozó fejlesztők számára. A Streamlit legfőbb előnye, hogy minimális kódmódosítással lehetővé teszi a Python szkriptek interaktív vizualizációját és felhasználóbarát kezelőfelületek kialakítását. Az alkalmazások létrehozása során nem szükséges HTML, CSS vagy JavaScript ismerete, mivel a Streamlit egyszerű API-t biztosít az interaktív elemek, például csúszkák, gombok és fájlfeltöltők hozzáadásához.

A Streamlit "script-first" megközelítést alkalmaz, ami azt jelenti, hogy a fejlesztők a Python kódot közvetlenül írják, és a keretrendszer automatikusan generálja az ehhez tartozó UI-t. Ez különösen hatékony prototípusok gyors elkészítéséhez vagy olyan alkalmazások fejlesztéséhez, ahol az adatok vizualizálása és manipulálása kulcsfontosságú. A Streamlit támogatja az élő újratöltést, így a fejlesztők azonnal láthatják a kódjukban történt változásokat, ami gyors iterációt tesz lehetővé.

A fejlesztők számára további előny, hogy a Streamlit natív támogatást nyújt az olyan népszerű Python-könyvtárakhoz, mint a Pandas, Matplotlib, Plotly és mások, így az adatmanipuláció és vizualizáció egyszerűen integrálható az alkalmazásokba. Könnyen telepíthető, és egyetlen parancssori utasítással indítható. Az alkalmazások helyileg futtathatók, de könnyedén telepíthetők felhőbe, például a Streamlit Community Cloud segítségével, vagy bármely más webhosting szolgáltatásra. [17]

5.4.3 Tokenizáció

A tokenizálás a szöveg apróbb részekre, azaz tokenekre bontását jelenti, amelyeket a nyelvi modellek feldolgozhatnak. A beágyazás a szavak vagy adatok geometriai reprezentációját, például vektoros megjelenítését foglalja magában. A figyelmi mechanizmus lehetőséget ad arra, hogy a modellek kiemelten kezeljék az adatok bizonyos szegmenseit. Az előtanítás a modell alapképzését jelenti nagy adathalmazon, amelyet számos feladattípus elvégzésére terveznek, újraépítés nélkül. A finomhangolás a modellek speciális feladatokhoz történő igazítását célozza a jobb eredmények érdekében. Az emberi visszajelzésen alapuló megerősítési tanulás egy olyan gépi tanulási stratégia, amely megerősítési tanulási technikákat és emberi felügyeletet integrál. [9]

You are reading a paper from Diagnostic and Interventional Radiology.

[1639, 389, 3555, 257, 3348, 422, 31549, 15132, 290, 4225, 20405, 5325, 12371, 13]

5.4. ábra: Tokenizáció.

[9]

5.4.4 Transzformerek

Egy alapvető transzformátor modell esetén a szövegbemenetet tokenekre bontják, amelyeket numerikus kódolás után beágyazásokra alakítanak át. A pozicionális jelölésekkel ellátott beágyazásokat a modell kódolója átfogó reprezentációvá alakítja. A dekóder az előző szavak alapján előrejelzi a következő szót, a kimeneti beágyazásokat lineáris réteg és softmax függvény segítségével alakítja vissza, létrehozva a kimeneti valószínűségi eloszlást. Ez az autoregresszív generálás folytatódik, amíg a teljes kimenet elkészül.

A maszkolt többfejes figyelemréteg különbözteti meg a transzformátor modellt más kódoló-dekódoló architektúráktól, biztosítva, hogy minden token kapcsolatban álljon az előző tokenekkel. Ez a réteg megelőzi, hogy a modell hozzáférjen a jövőbeni tartalmakhoz, megőrizve a sorozat ok-okozati viszonyait. A többfejes figyelemréteg párhuzamosan működő figyelemmechanizmusokat tartalmaz.

Az LLM-ek képesek külső eszközöket használni, például számológépeket vagy keresőmotorokat, hogy kezeljék a szövegen túlmutató feladatokat. Ezek a modellek integrálhatók külső alkalmazásokkal, mint a LangChain, jelentősen bővítve funkcionalitásukat..[18]

5.5 A Nagy Nyelvi Modellek áttekintő elemzése

5.5.1 ChatGPT

A ChatGPT egy mesterséges intelligencia által vezérelt chatbot, amely gépi tanulási technikákkal válaszol a kérdésekre beszélgetés formájában. A GPT a Generative Pre-trained Transformer rövidítése, amely az adatsorok közötti mintázatok felismerésére szolgál. ChatGPT képes megjegyezni, amit a felhasználó korábban mondott a beszélgetés során, ami elősegíti a folytonos párbeszédet. 2023 márciusában, az OpenAI által kifejlesztett GPT-4 bevezetésével a ChatGPT is jelentős frissítést kapott további funkciókkal. Konkrétan, a felhasználók mostantól párhuzamosan vihetnek be szövegeket és vizuális képeket, lehetővé téve összetettebb multimodális feladatok elvégzését, mint például a képfeliratok készítése, diagramok értelmezése, vagy tudományos cikkek összefoglalása.[19]

5.5.2 Bard

A Google Bard szintén egy mesterséges intelligencia által vezérelt chatbot, amely természetes nyelvfeldolgozással és gépi tanulással szimulálja az emberi beszélgetéseket, válaszokat keresve az internetről. Eredetileg a LaMDA modellt használta párbeszédész alkalmazásokhoz, de most a Google következő generációs nyelvi modelljét, a PaLM 2-t (Pathways Language Model) alkalmazza. A Google egy blogbejegyzésben kijelentette, hogy ez a modell jobban teljesít hétköznapi logikai és matematikai feladatokban, valamint gyorsabb is, mint korábbi verziói. A Bard részletesebb válaszokat nyújt, mint egy hagyományos Google keresés, a nagy nyelvi modell segítségével.

A Bard fő célja, hogy egyszerű válaszokkal szolgáljon egy keresőmotor eredményoldala helyett, hasonlóan a digitális asszisztensekhez, mint Alexa és Siri, de linkeket is biztosít további információk gyűjtéséhez. Személyes asszisztensként is működik, segítve a felhasználókat például nyaralások foglalásában, meglévő foglalások keresésében és étkezési tervezésben. A Bard beszélgetési vagy természetes nyelvi lekérdezéseket használ a kereséshez, a hagyományos keresőmotorok kulcsszavai helyett. A Bard világszerte elérhető a nyilvánosság számára, három nyelven, 180 országban.[19]

5.5.3 Összehasonlítás

A Bard és a ChatGPT közötti fő különbség a mögöttük álló nagy nyelvi modellek (LLM) felépítésében rejlik. Az LLM-ek célja az emberi nyelven történő szöveg megértése, értelmezése és generálása. Minél jobban képzettek ezek a modellek, annál pontosabb és érthetőbb tartalmat képesek előállítani.

A Bard és a ChatGPT eltérő LLM-eken fut, mindkettőt különböző adatkészleteken képezték ki:

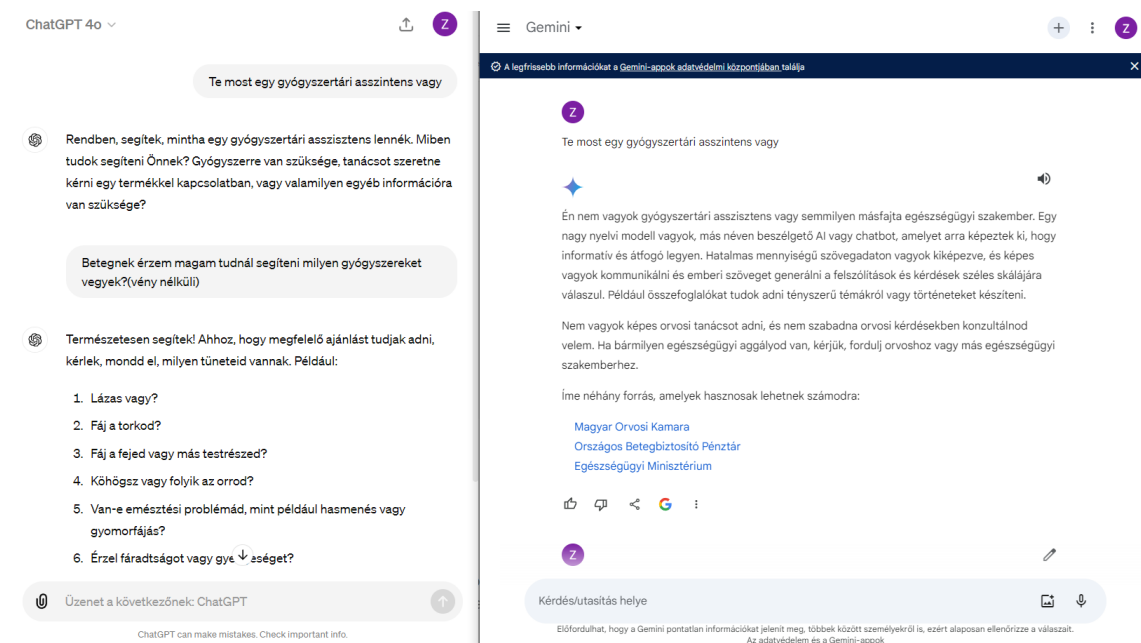
A Bard a PaLM 2-t használja, amelyet specifikus adatokra képeztek ki a párbeszéd és a kódolási képességek javítása érdekében, beleértve a tudományos dolgozatokat, matematikai kifejezéseket és forráskódokat. Ez a chatbot valós idejű internet-hozzáféréssel is rendelkezik, ami lehetővé teszi, hogy naprakész válaszokat adjon.

A ChatGPT a GPT-3.5-öt alkalmazza, amely egy előre meghatározott adatkészleten alapul, és 2021 óta nem frissítették. Habár nem rendelkezik aktuális információkkal, kiterjedt adatkészlete lehetővé teszi, hogy szinte bármilyen témában részletes válaszokat adjon. A ChatGPT előfizetői havi 20 dollárért hozzáférhetnek a GPT-4-hez is, amely a GPT-3.5 nagyobb és frissített verziója.

Mind a Bard PaLM, mind a ChatGPT GPT LLM-jei az interneten található cikkekből, könyvekből és weboldalakból származó adatokon tanulnak, tehát válaszaik az interneten elérhető információkon alapulnak.

Az OpenAI GPT-4 azonban több adatforráshoz fér hozzá, és több mint 95 nyelven tud válaszokat generálni, szemben a PaLM 2 által támogatott 40 nyelvvel. Ha a Bard nem talál választ egy kérdésre, az interneten kereshet, hogy megtalálja a választ, vagy megfelelő weboldalakat ajánlhat különböző nyelveken.[20]

Itt összehasonlítottam a két említett programot felhasználói szemmel. Az eredmény az lett hogy a ChatGPT tudott gyógyszerügyi asszisztens is lenni, a Gemini viszont nem.



5.5. ábra: ChatGPT kontra Gemini felhasználói felületen

Mivel a projektet úgy szeretném használni, hogy az OpenAI segítségével, API hívásokon keresztül ezért ezt kódba is foglaltam, és kipróbáltam úgy mit tapasztalok. A tapasztalatom az volt, hogy az API hívások jóval gyorsabbak, ha hibázok a hibák gyakran könnyebben megfejthetőek mintha a felhasználói felület adná, válaszok struktúráltak és előre meghatározott formátumúak.

Van is egy példa, ahol megkértem, hogy legyen egy gyógyszerértési asszisztens, és soroljon nekem össze-vissza gyógyszereket.

```
1 import os
2 OPENAI_API_KEY=os.environ['OPENAI_API_KEY']
3
4
5
6
7
8 from openai import OpenAI
9 client = OpenAI()
10
11
12 completion = client.chat.completions.create(
13     model="gpt-3.5-turbo",
14     messages=[
15         {"role": "system", "content": "You are a pharmacy assistant, you need to say some medicine."},
16         {"role": "user", "content": "Give me medicine names"}
17     ]
18 )
19
20 print(completion.choices[0].message)
```

PROBLÉMÁK KIMENET TERMINÁL PORTS JUPYTER HIBAKERESÉSI KONZOL

PS C:\Users\36789> & C:\Users\36789\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\36789\OneDrive\Isztali gép\szakdoga\openai-test.py"

ChatCompletionMessage(content="Sure, here are some common medication names:\n\n1. Acetaminophen (Tylenol)\n2. Ibuprofen (Advil, Motrin)\n3. Omeprazole (Prilosec)\n4. Lisinopril (Zestril, Prinivil)\n5. Atorvastatin (Lipitor)\n6. Amoxicillin\n7. Metformin (Glucophage)\n8. Albuterol (Proair, Ventolin)\n9. Prednisone\n10. Levothyroxine (Synthroid) \n\nPlease note that these are just examples and it's important to consult with a healthcare professional before taking any medication.", role="assistant", function_call=None, tool_calls=None)

PS C:\Users\36789>

5.6. ábra: OpenAI Gyógyszer sorolás API-val

Az RLHF (Reinforcement Learning from Human Feedback) célja, hogy az LLM-ek (nagy nyelvi modellek) emberközpontú asszisztensként működjenek, segítőkészen, őszintén és ártalmatlanul. A folyamat során először az LLM-eket nagy, alacsony minőségű adathalmazokon képzik a szövegkiegészítés optimalizálása érdekében. Ezt követően felügyelt finomhangolással javítják a modellek párbeszédképességeit magas minőségű, párbeszédre optimalizált adatokkal.

Az emberi visszajelzések alapján jutalommodellt alkalmaznak, amely a modellek válaszait értékeli és pontozza. Az így kapott adatok segítségével megerősítő tanulási algoritmusokat, például a PPO (Proximal Policy Optimization) algoritmust használják a modellek válaszainak optimalizálására, hogy azok minél inkább megfeleljenek az emberi preferenciáknak.

Az RLHF alkalmazása kihívásokkal teli, mivel a jutalmazási rendszer tervezése, a környezeti interakció és az ügynökképzés komplex feladatok. Az AI-kutatók jelentős aka-

dályokkal szembesülnek az LLM-ek technikai összehangolásában és biztonságos működésének biztosításában. Ennek ellenére az RLHF stabil alkalmazása kulcsfontosságú az emberközpontú mesterséges intelligencia fejlesztésében.

Az RLHF technológiai megközelítései, mint a jutalommodellek és a PPO algoritmusok, kritikus szerepet játszanak az LLM-ek finomhangolásában és optimalizálásában. A nyílt forráskódú implementációk és műszaki jelentések közzététele fontos a további fejlődés támogatásához és a biztonságos alkalmazások kialakításához.[21]

5.6 A ChatGPT etikai kérdései az orvostudományban és az egészségügyben

A ChatGPT bevezetésével a Nagy Nyelvi Modellek (LLM-ek) nagy figyelmet kaptak az egészségügyben. Bár jelentős előnyökkel járhatnak, mint az adatkezelés, személyre szabott információk és döntéshozatali támogatás, számos etikai kérdés merült fel. Ezek közé tartozik a méltányosság, elfogultság, ártalommentesség, átláthatóság és adatvédelem. Különösen aggasztó a félrevezető vagy pontatlan információk előállítása. Az etikai iránymutatások és emberi felügyelet szükségessége egyértelmű. Fontos, hogy az etikai iránymutatások az elfogadható emberi felügyelet meghatározására összpontosítsanak az egészségügyi alkalmazásokban. Emellett kritikai vizsgálatra van szükség az LLM-ek kísérleti használatának szükségességéről és indokoltságáról.

6 HASONLÓ RENDSZEREK BEMUTATÁSA

Több olyan weboldalt is találtam mint például a Docus AI Health Assistant [22], Healthy [23], vagy akár az Ada [24], amelyek Nagy Nyelvi Modellek (LLM-ek) használatával nyújtanak orvosi diagnosztikát. Ezek az oldalak gyakran alkalmazzák az LLM-eket az egészségügyi adatok elemzésére, a betegségek felismerésére és a kezelési javaslatok megfogalmazására. Az ilyen alkalmazások célja, hogy gyorsabb és pontosabb diagnózisokat biztosítsanak, valamint hogy támogassák az orvosokat a döntéshozatalban, különösen bonyolult esetekben.

A szakdolgozatomban részletesen foglalkozom az LLM-ek orvosi diagnosztikai alkalmazásainak előnyeivel és kihívásaival, különös tekintettel a gyógyszerajánlásokra.. Kutatásom során elemzem, hogyan használják ezeket a modelleket a gyakorlatban, és milyen hatással vannak az orvosi ellátás minőségére, ami legjobb tudásom szerint az iparág segítségére lehet. Továbbá, Feltárom az etikai kérdéseket és a szabályozási kihívásokat is, Ugyanakkor feltárom az ilyen technológiák alkalmazásával járó veszélyeket is, például azt, hogy az AI rendszerek nem teljesen pontosak, és a hibás gyógyszeres kezelés komoly következményekkel járhatnak.

Célom, hogy átfogó képet nyújtsak arról, hogyan integrálhatók az LLM-ek az egész-

ségügyi rendszerekbe, és milyen lépések szükségesek ahhoz, hogy ezek a technológiák biztonságosan és hatékonyan támogassák az orvosi diagnosztikát. Remélem, hogy munkám hozzájárul az LLM-ek etikus és felelősségteljes használatának előmozdításához az egészségügyben.

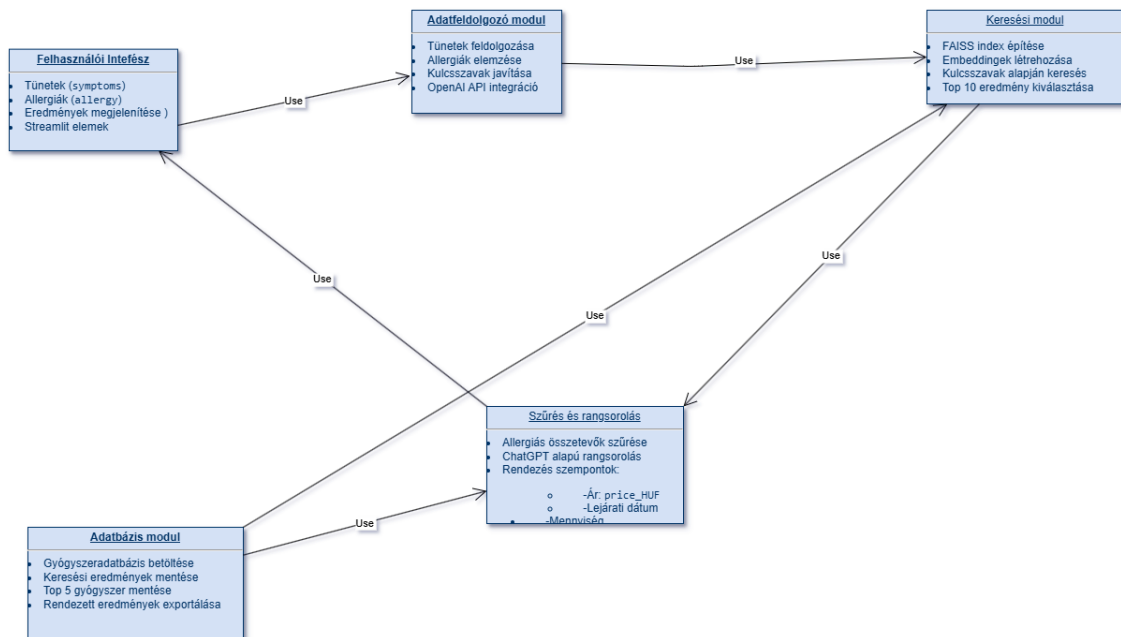
Docu AI Health Assistant [22] és a Healthy [23], amelyek a legközelebb állnak ahhoz a programhoz, amelyet szeretnék létrehozni, képesek a tünetek alapján meghatározni a lehetséges betegségeket, azonban nem nyújt gyógyszeres tüneti kezelést. Az általam tervezett projekt hasonló módon működne: a felhasználóktól bekérné a tüneteket, majd további pontosító kérdéseket tenne fel, hogy pontosabb diagnózist adhasson. Ez a megközelítés lehetővé teszi, hogy a rendszer alaposabb és megbízhatóbb eredményeket nyújtson, mintha csak egyszer kérné be a tüneteket. Ez a Weboldal sem rendelkezik saját AI-val, api hívásokon keresztül kommunikál egy másik GPT-alapú modellel, mint például a GPT-4. A Docu AI a felhasználók által megadott információkat és tüneteket elemzi, majd ezek alapján külső GPT-modelleket használ a válaszok generálására és egészségügyi tanácsok nyújtására. Céлом, hogy egy olyan eszközt hozzak létre, amely támogatja az gyógyszerészeket és a pácienseket a pontosabb diagnózis felállításában, miközben figyelembe veszi az egyedi tüneteket és azok részleteit.

Az Ada [24] saját mesterséges intelligenciával rendelkezik, amelyet orvosok és tudósok fejlesztettek ki házon belül. Ez az AI rendszer több mint 3600 betegséget és 10 000 tünetet képes felismerni és értékelni. Az Ada rendszer integrálható más egészségügyi platformokba is API-hívásokon keresztül, például az Epic EHR-be, lehetővé téve a felhasználók számára, hogy közvetlenül a platformon végezzenek tünetellenőrzéseket.

Az én rendszerem több szempontból is újszerű és előnyösebb lehet. Először is, az API-alapú működés révén a rendszerem rugalmasan integrálható a legmodernebb külső AI-modellekkel, például a legújabb GPT-alapú technológiákkal. Ez biztosítja, hogy mindig a legújabb és legjobb AI-algoritmusokat használhassam az adatok elemzésére és a válaszok generálására. Másodszor, a rendszerem vektoradatbázist használ, amely lehetővé teszi a hatékonyabb és pontosabb adatkezelést és keresést. A vektoradatbázis alkalmazása révén a rendszerem képes lesz a tünetek alapján személyre szabott gyógyszerajánlásokat is nyújtani a felhasználóknak. Ez nem csak az egészségügyi tanácsadás színvonalát emeli, hanem gyorsabb és relevánsabb válaszokat is biztosít a felhasználók számára.

Az a projekt, amelyet írok, hasonló módon fog működni mint a Docu AI Health Assistant [22], de néhány további funkcióval és fejlesztéssel szeretném jobbá tenni. Az alábbiakban néhány javaslat található arra, hogy miben lehetne fejleszteni a projektet: Részletesebb és interaktívabb tünetlekérdezés: A rendszer ne csak egyszer kérje be a tüneteket, hanem folyamatosan tegye fel pontosító kérdéseket, hogy minél részletesebb képet kapjon a felhasználó állapotáról. Ez növelheti a diagnózis pontosságát.

7 RENDSZERTERV A GYÓGYSZERAJÁNLÓ ALKALMAZÁSRÓL



7.1. ábra: Rendszerterv

7.1 A rendszer céljai

7.2 Technológiai eszközök

A gyógyszerajánló rendszer fejlesztéséhez a következő technológiai eszközöket és megoldásokat választottam, figyelembe véve a rendelkezésre álló erőforrásokat és a célkitűzéseket:

– Technológiák:

- Programozási nyelv: A Python kiváló választás volt a projekt megvalósításához, mivel egyszerű szintaxisa, széles körben elérhető könyvtárai és közösségi támogatása lehetővé tette a gyors fejlesztést és tesztelést. Az adatok kezelése, a mesterséges intelligencia modellek integrációja és a teljes backend logika implementálása során a Python robusztus és megbízható eszköznek bizonyult.
- Frontend keretrendszer: A Streamlit-et választottam a frontend megvalósításához, mivel a rendelkezésemre álló erőforrások miatt egy hagyományos weboldal fejlesztése jelentősen lassította volna a folyamatot, és a program teljesítményét is hátrányosan érintette volna. A Streamlit egyszerűségének köszönhetően az interaktív felületet gyorsan létre tudtam hozni, így minimalizálva a fejlesztési időt. Ezen kívül a Streamlit támogatja az interaktív adatbevitelt

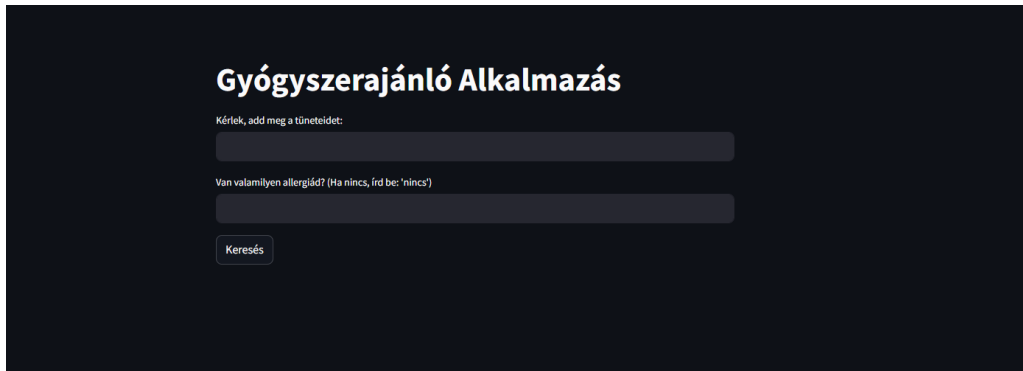
és az eredmények dinamikus megjelenítését, ami ideális választássá tette a gyógyszerértári alkalmazás számára.

- Adatbázis: CSV fájlok és pandas: A rendszer adatainak tárolására CSV fájlokat használtam, amelyek jól skálázhatók és könnyen kezelhetők Python pandas könyvtárával. A pandas lehetővé tette az adatbázis gyors betöltését, szűrését és manipulációját. A top 5 gyógyszer kiválasztásához és az ár szerinti rendezéshez több megközelítést is kipróbáltam. A pandas kezdetben jól működött a kisebb adathalmazokon, de amikor bonyolultabb szűréseket és rendezéseket integráltam (például a FAISS keresés és a ChatGPT4minu rangsorolás után), a kód bonyolultabbá és kevésbé átláthatóvá vált. Ezért optimalizáltam a folyamatot, hogy az egyszerűbb megközelítéseket részesítsem előnyben, ahol csak lehetett, a gyorsaság és az olvashatóság érdekében.
- Keresés: FAISS: A FAISS indexet a tünetek alapján releváns gyógyszerek gyors azonosítására használtam. Ez a technológia különösen akkor hasznos, amikor a keresési folyamat nagyméretű adatbázison fut, mivel a FAISS az embeddingek hasonlóságát méri, és hatékonyan szűkíti le a találatokat. Az OpenAI embedding modellje által generált numerikus vektorok bekerülnek a FAISS indexbe, amely lehetővé teszi a tünetekhez leginkább illeszkedő gyógyszerek azonosítását. A FAISS nemcsak gyors keresést biztosít, hanem jól integrálható a Python ökoszisztémába is.
- OpenAI API: Az OpenAI API többféleképpen is támogatja a rendszert. A ChatGPT modellt a kulcsszógenerálásban és a releváns gyógyszerek rangsorolásában használtam. A kulcsszógenerálás során a ChatGPT biztosítja, hogy a felhasználó által megadott tünetekből releváns kulcsszavak szülessenek, még akkor is, ha a bevitel nem pontos vagy helyesírási hibákat tartalmaz. A rangsorolás szintén a ChatGPT4mini-re épül, amely a tünetek és gyógyszerek adatai alapján kiválasztja a legmegfelelőbb 5 gyógyszert, figyelembe véve a relevanciát. Az OpenAI embedding modellje kulcsszerepet játszik a FAISS indexek működésében. Az embeddingek numerikus vektorokká alakítják a szöveges adatokat, lehetővé téve a hasonlóságok gyors és pontos mérését. Ez különösen fontos a tünetekhez legjobban illeszkedő gyógyszerek azonosításában. Bár az OpenAI API használata némi teljesítménybeli kompromisszumot jelentett, a pontosság és az intelligens keresési eredmények jelentősen javultak, így megérte beépíteni a projektbe.

7.2.1 Egyszerű és gyors keresés

A rendszer célja, hogy a felhasználó által megadott tünetek alapján gyorsan és hatékonyan ajánlja a legrelevánsabb gyógyszereket. Az alkalmazás egyszerű, felhasználóbarát

felületet biztosít, amelyet kifejezetten gyógyszerértári munkások számára terveztek. Az intuitív kezelőfelület minimalizálja a kereséshez szükséges időt, és lehetővé teszi, hogy a gyógyszerértári dolgozók gyorsan megkapják a szükséges információkat, ezáltal hatékonyabb kiszolgálást biztosítanak az ügyfeleknek.



7.2. ábra: A felhasználói felület

7.2.2 Allergiák kezelése

A rendszer kiemelt figyelmet fordít a felhasználó által megadott allergiák kezelésére. A keresés során az allergén összetevőket tartalmazó gyógyszerek automatikusan kizárásra kerülnek, ezzel biztosítva, hogy az ajánlott gyógyszerek biztonságosan alkalmazhatók legyenek. Az allergiás összetevők szűrését mesterséges intelligencia támogatja, amely pontosan azonosítja az allergiákat és kizárja az érintett termékeket, ezáltal a gyógyszerértári munkások gyorsan és hatékonyan tudnak olyan gyógyszereket ajánlani, amelyek megfelelnek az ügyfél speciális igényeinek.

7.2.3 Rangsorolás

A rendszer a releváns gyógyszerek listáját tovább szűkíti és rangsorolja a legfontosabb kritériumok – ár, lejárat idő és mennyiség – alapján. Ez a megközelítés biztosítja, hogy a gyógyszerértári munkások a legjobb ár-érték arányú, friss készletből származó és megfelelő mennyiségű gyógyszereket ajánlhassák az ügyfeleknek. Az automatikus rangsorolás segít optimalizálni a döntéshozatalt, csökkenti a keresési időt, és megkönnyíti a legmegfelelőbb gyógyszerek kiválasztását.

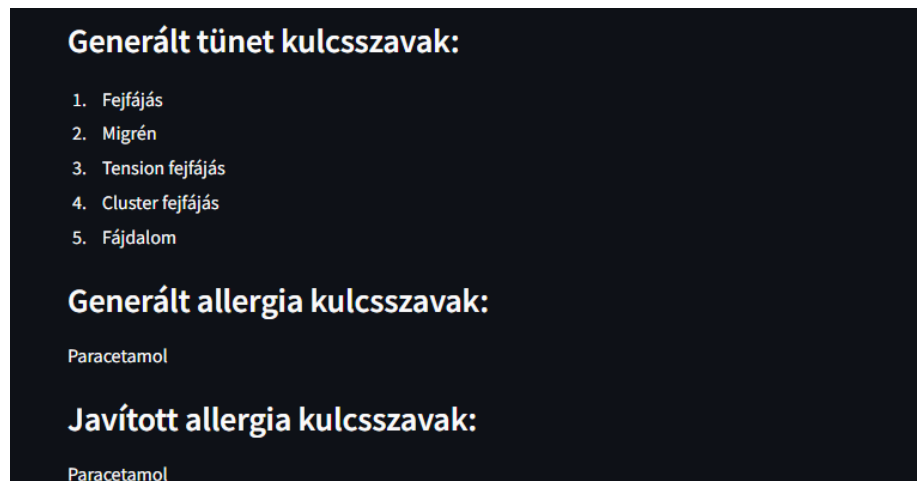
7.2.4 Könnyű hozzáférhetőség

Az alkalmazás böngészőben futtatható a Streamlit keretrendszer segítségével, amely egyszerű és gyors telepítést, valamint használatot tesz lehetővé. A felhasználóknak nincs szükségük bonyolult szoftverek telepítésére vagy speciális technikai ismeretekre; elegendő egy internetkapcsolattal rendelkező eszköz. Ezáltal a rendszer bárhol könnyen elérhető, ami különösen előnyös a gyógyszertárak dinamikus munkakörnyezetében, biztosítva a zökkenőmentes és hatékony működést.

7.3 Funkcionális követelmények

7.3.1 Adatbevitel és kulcsszógenerálás

Az alkalmazás lehetővé teszi a felhasználók számára, hogy egyszerűen megadják tüneteiket és allergiáikat szöveges formában. A tünetek alapján a rendszer releváns kulcsszavakat generál, amelyek segítik a pontos gyógyszerkeresést. Az allergiák megadásakor a felhasználók felsorolhatják az allergén összetevőket, vagy ha nincs ismert allergiájuk, egyszerűen beírhatják a „nincs” kulcsszót. A rendszer intelligensen kezeli az esetleges elírásokat és helyesírási hibákat, így a pontatlan bevitel sem akadályozza a pontos keresést. A kulcsszógenerálás folyamata során a ChatGPT mesterséges intelligencia segítségével releváns kulcsszavak kerülnek előállításra, amelyek a tünetek és allergiák alapján optimalizálják a keresési eredményeket. Ez biztosítja a gyors és hatékony adatfeldolgozást, miközben a felhasználók számára egyszerű és kényelmes adatbevitel áll rendelkezésre.



7.3. ábra: Kulcsszógenerálás

7.3.2 Keresés és szűrés

A rendszer a FAISS keresőindexet használja a gyógyszerek gyors és hatékony azonosítására a felhasználó által megadott tünetek alapján. Ez a megoldás biztosítja, hogy a keresés nagy mennyiségű adatban is pontos és gyors legyen. A keresési eredmények további szűrésen mennek keresztül: az allergiák mezőben megadott allergéneknek megfelelő gyógyszerek automatikusan kizárásra kerülnek. Ezáltal a felhasználó biztos lehet abban, hogy a kapott gyógyszerjavaslatok nem tartalmaznak olyan összetevőket, amelyek problémát okozhatnának. A keresés és szűrés folyamata a relevancia és a biztonság érdekében szorosan együttműködik.

FAISS Keresési Eredmények:

	name	ingredients	side_effects
49	Lidocaine	Lidocaine	Lehetséges mellékhatások: bőrreakciók, égő érzés
13	Gabapentin	Gabapentin	Lehetséges mellékhatások: szédülés, fáradtság
48	Pregabalin	Pregabalin	Lehetséges mellékhatások: szédülés, álmoság
18	Tramadol	Tramadol	Lehetséges mellékhatások: szédülés, álmoság
40	Oxycodone	Oxycodone	Lehetséges mellékhatások: székrekedés, álmoság
0	Paracetamol	Acetaminophen	Lehetséges mellékhatások: májkárosodás hosszú távú használat esetén
2	Aspirin	AcetylsalicylicAcid	Lehetséges mellékhatások: gyomorvérzés, allergiás reakciók
51	Meloxicam	Meloxicam	Lehetséges mellékhatások: gyomorfájdalom, fejfájás
1	Ibuprofen	Ibuprofen	Lehetséges mellékhatások: gyomorirritáció, vérzés
38	Naproxen	Naproxen	Lehetséges mellékhatások: gyomorfájdalom, hányinger

7.4. ábra: FAISS Keresési Eredmények

7.3.3 Rangsorolás és adatok megjelenítése

A rendszer a releváns gyógyszereket egy mesterséges intelligenciával támogatott rangsorolási folyamat során szűkíti le a legmegfelelőbb öt ajánlatra, amely a felhasználói igények és a biztonsági szempontok egyensúlyára épül. Ehhez az OpenAI GPT-4Mini modellt használja, amely a felhasználó által megadott tünetek alapján elemzi és értékeli a keresési eredményeket. A modell figyelembe veszi a tünetekhez kapcsolódó relevanciát, a hatóanyagok és összetevők megfelelőségét, valamint a felhasználó által megadott allergiákat, biztosítva ezzel, hogy a javasolt gyógyszerek nemcsak hatékonyak, hanem biztonságosak is legyenek. A rangsorolási algoritmus kifejezetten a legjobban illeszkedő gyógyszereket emeli ki, szinonimák és hasonló kifejezések figyelembevételével, még akkor is, ha a felhasználó tünetei hiányosan vagy pontatlanul lettek megadva.

Miután a gyógyszereket a relevancia szerint leszűkítette, a rendszer ár, lejáratidő és elérhetőség alapján tovább rendezi az eredményeket. Ez a lépés biztosítja, hogy a felhasználók számára költséghatékony, aktuális és azonnal hozzáférhető termékek kerüljenek ajánlásra. Az ár-érték arány különös hangsúlyt kap a rangsorolásban, hogy a felhasználókat ne csak a legjobb gyógyászati megoldásokkal, hanem gazdaságos opciókkal is segítse. A lejáratidő figyelembevétele pedig garantálja, hogy a vásárlók mindig friss és használható készítményeket kapjanak, míg az elérhetőség alapján a rendszer biztosítja, hogy csak

ténylegesen készleten lévő gyógyszerek kerüljenek a listára.

A rangsorolt eredmények a felhasználói élmény maximalizálása érdekében CSV formátumban kerülnek mentésre. Ez a fájlformátum könnyen kezelhető, archiválható és egyszerűen megosztható a gyógyszerértékelő dolgozók között, támogatva a további nyilván tartási és elemzési munkafolyamatokat. Az alkalmazás az eredményeket egy tiszta, jól áttekinthető táblázat formájában jeleníti meg a felhasználói felületen. A táblázat színezése és vizuális elrendezése segít kiemelni a legfontosabb információkat, például az árat, a lejárat időt és az elérhető mennyiséget, lehetővé téve a gyors és egyszerű döntéshozatalt.

A rendszer a CSV fájlok mentése mellett garantálja azok könnyű hozzáférhetőségét és használhatóságát további feldolgozásra. Ezáltal a gyógyszerészek és a gyógyszerértékelő dolgozói számára biztosítja a gyors, pontos és átlátható adatkezelést, amely megkönnyíti a munkafolyamatokat. A rangsorolási algoritmus nemcsak a felhasználói preferenciákat veszi figyelembe, hanem a gyógyszerek biztonságát és elérhetőségét is garantálja. Az adatok egy privát GitHub repositoryban tárolódnak, amelyhez csak hozzáadott személyek férhetnek hozzá, így a biztonságos hozzáférés is biztosított. Ez a megközelítés jelentősen csökkenti az adminisztratív terheket, gyorsítja a gyógyszerek kiválasztását, és hozzájárul a gyógyszerértékelő ügyfélszolgálatának minőségi javításához. Az egész rendszer moduláris felépítésének köszönhetően könnyen adaptálható és bővíthető, így hosszú távon is támogatja a hatékony gyógyszerajánlási folyamatokat. A főbb modulok a következők: Felhasználói Interfész Modul, amely a felhasználói interakciókat kezeli; Adatfeldolgozó Modul, amely a tünetek elemzéséért és a kulcsszavak kinyeréséért felelős; Keresési Modul, amely a gyógyszeradatbázisban történő gyors keresést valósítja meg, például FAISS technológiával; és a Szűrés és Rangsorolás Modul, amely allergének és egyéb kritériumok alapján szűri és rangsorolja a gyógyszereket. Ez a felépítés biztosítja a rendszer rugalmasságát és hatékonyságát a változó igények mellett.

Top 5 Gyógyszer (Rendezett):				
	name	ingredients	side_effects	symptoms
3	Tramadol	Tramadol	Lehetséges mellékhatások: szédülés, álmoság	fájdalom
4	Oxycodone	Oxycodone	Lehetséges mellékhatások: székrekedés, álmoság	fájdalom
9	Naproxen	Naproxen	Lehetséges mellékhatások: gyomorfájdalom, hányinger	gyulladás,fájdalom
6	Aspirin	AcetylsalicylicAcid	Lehetséges mellékhatások: gyomorvérzés, allergiás reakciók	fejfájás,szívritmuszavar
8	Ibuprofen	Ibuprofen	Lehetséges mellékhatások: gyomorirritáció, vérzés	gyulladás,izomfájdalom

A rendezett top 5 gyógyszer mentése megtörtént a 'sorted_top_5_results.csv' fájlba.

7.5. ábra: Top 5 Gyógyszer

7.4 Nem-funkcionális követelmények

A teljesítmény szempontjából a cél az, hogy a keresési lekérdezések átlagos esetben kevesebb mint 5 másodperc alatt eredményt adjanak. Ez a gyors válaszidő kulcsfontosságú a gyógyszertári dolgozók számára, akik gyakran időnyomás alatt dolgoznak. A FAISS keresőmotor és az OpenAI API hatékony kombinációja garantálja, hogy a rendszer nagy mennyiségű adatot is gyorsan és pontosan tudjon feldolgozni. Az algoritmusok optimalizálása minimalizálja a késleltetéseket, így a teljesítmény nagyobb adatbázisokkal is fenntartható.

A rendszer hibatűrő képessége lehetővé teszi, hogy az esetleges hibás adatbevitel vagy üres keresési feltételek ne akadályozzák a felhasználói élményt. Hibás vagy pontatlan adatok megadása esetén figyelmeztető üzenet jelenik meg, amely útmutatást nyújt a helyes adatbevitelhez. Az elírásokat és helyesírási hibákat a rendszer automatikusan javítja, így a keresési folyamat zökkenőmentesen működik. A rendszer biztosítja, hogy hibás adatbevitel esetén sem omlik össze, hanem megfelelő alternatív eredményeket vagy hiba-üzeneteket nyújt a felhasználóknak.

A rendszer úgy lett tervezve, hogy az adatbázis bővítése és a felhasználói terhelés növekedése esetén is megfelelő teljesítményt nyújtson. A FAISS indexek hatékony működé-

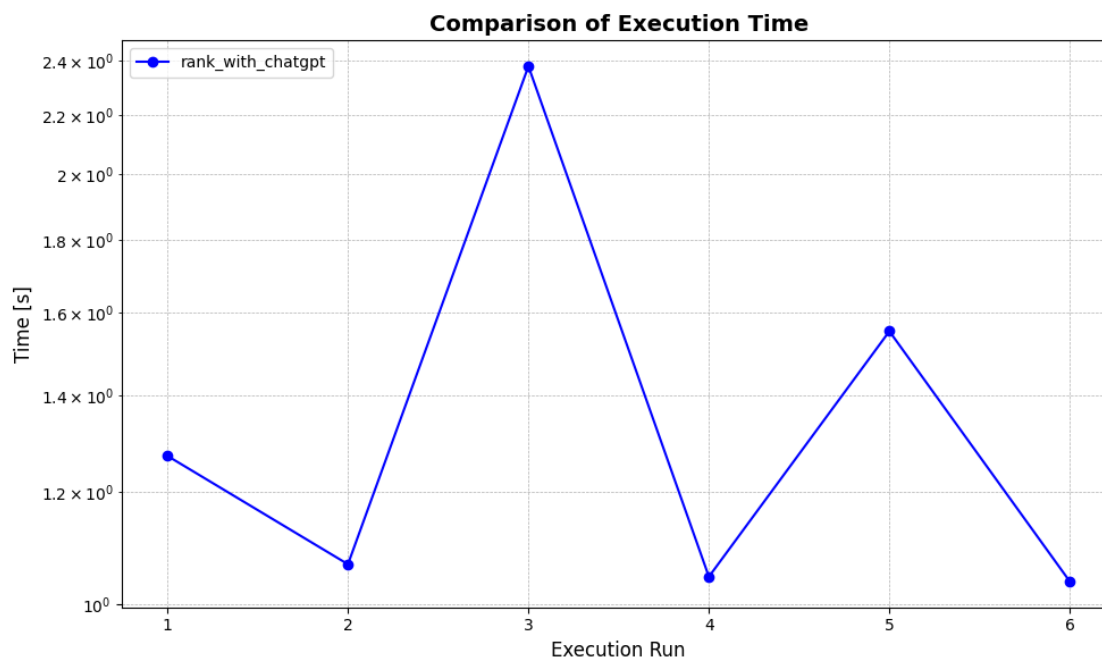
se és a párhuzamos műveletek támogatása révén a rendszer képes több egyidejű lekérdezést is gyorsan feldolgozni. Az OpenAI API terhelésének növekedése szintén kezelhető, így a rendszer alkalmas nagyobb felhasználói bázis kiszolgálására is. A felhőalapú környezet, például a Streamlit Cloud, további rugalmasságot biztosít a skálázhatósághoz.

A biztonság szintén kiemelt prioritás. Az OpenAI API kulcsok és egyéb érzékeny adatok titkosítva kerülnek tárolásra, és kizárólag a megfelelő jogosultságokkal rendelkező folyamatok férhetnek hozzájuk. A rendszer nem tárol tartósan érzékeny adatokat, ezzel minimalizálva az adatvédelmi kockázatokat. Érttem itt ezalatt azt hogy a rendszerhez általános tünetekkel dolgozik nem, személyi adatok nélkül. A titkosítás alkalmazása biztosítja, hogy az adatok védettek legyenek az illetéktelen hozzáféréssel szemben, ezt szintén a privát GitHub repositoryval oldottam meg.

A megbízhatóságot és a karbantarthatóságot a kódstruktúra és a részletes dokumentáció támogatja. A rendszer logolási funkciói segítenek a hibák gyors azonosításában és javításában, ezáltal minimalizálva a leállások esélyét. A fejlesztések és bővítések egyszerűen implementálhatók, így a rendszer hosszú távon is fenntartható.

A felhasználói élményt az egyszerű és intuitív kezelőfelület biztosítja, amely gyors adatbevitelre és eredmények megjelenítésére lett optimalizálva. Az eredmények jól strukturált táblázatos formában jelennek meg, amely segíti a gyógyszerértári dolgozókat a hatékony döntéshozatalban.

Összességében a rendszer biztosítja, hogy gyors, megbízható, skálázható és biztonságos legyen, miközben a felhasználók számára könnyen használható felületet nyújt. Ezáltal nemcsak az aktuális igényeknek, hanem a jövőbeni fejlesztési lehetőségeknek is megfelel.



7.6. ábra: Futás idő

7.5 Architektúra

A rendszer három fő modulból áll: felhasználói interfész, backend logika és adatbázis. A felhasználói interfész a Streamlit keretrendszeren alapul, amely egyszerű és intuitív adatbevitelt biztosít. A backend logika tartalmazza a kulcsszógenerálást, FAISS keresést, szűrést és rangsorolást, amelyek a gyógyszerek pontos azonosítását és ajánlását végzik. Az adatbázis a gyógyszerek adatait tartalmazza CSV formátumban, beleértve a neveket, tüneteket, összetevőket, árakat és lejárat dátumokat, így a rendszer gyorsan hozzáférhet a szükséges információkhoz.

7.5.1 Felhasználói interfész

A rendszer felhasználói interfészét a Streamlit keretrendszer biztosítja, amely egyszerűen kezelhető, gyorsan implementálható és intuitív megoldásokat kínál. Az interfész fő feladata az adatbevitel és az eredmények megjelenítése, mindezt letisztult és átlátható formában. A felhasználók szöveges mezőkben adhatják meg tüneteiket és allergiáikat, amelyek az ajánlások kiindulópontjaként szolgálnak. Az űrlap kitöltése után egy gomb segítségével indítható el a keresés, amely azonnal megjeleníti az eredményeket.

Az interfész valós időben jelenít meg figyelmeztető üzeneteket, ami azt jelenti, hogy a rendszer azonnal reagál a felhasználó által bevitt adatokra, amint azok hibásak, hiányosak

vagy nem felelnek meg az elvárt formátumnak. Ez a funkció biztosítja, hogy a felhasználó már az adatbevitel során kapjon visszajelzést, például ha a tünetek mező üresen marad, a rendszer azonnal figyelmeztetést küld: „Kérlek, add meg a tüneteidet!”. Hasonló módon, ha az allergiák mezőbe helytelen adatot visznek be, például nem konkrét összetevőt, az interfész rögtön jelez: „Kérlek, adj meg érvényes allergiát, vagy írd be, hogy 'nincs'.” Ez a valós idejű működés jelentős különbséget jelent az olyan rendszerekkel szemben, amelyek csak egy adott művelet, például a „Keresés” gomb megnyomása után ellenőrzik az adatokat. Az ilyen nem valós idejű rendszerek esetén a felhasználónak előbb el kell végeznie az adatbevitel teljes folyamatát, majd a hibák kijelzése után vissza kell térnie az adatok javításához, és ismét el kell indítania a folyamatot. Ez a megközelítés nemcsak időigényesebb, hanem nagyobb frusztrációt is okozhat a felhasználók számára, különösen akkor, ha gyakran ismétlődő hibákról van szó.

A valós idejű figyelmeztetések használata jelentősen javítja a felhasználói élményt, mivel az azonnali visszajelzés lehetővé teszi a hibák gyors kijavítását még a folyamat korai szakaszában, anélkül, hogy újra kellene indítani a keresési műveletet. Ez a megközelítés nemcsak időt takarít meg, hanem csökkenti a tanulási görbét, mivel a rendszer egyértelmű útmutatást nyújt a felhasználók számára. Ezen túlmenően, a valós idejű visszajelzés segít a helyes adatok biztosításában, ami elengedhetetlen a pontos keresési eredmények eléréséhez. A felhasználók hibáinak minimalizálásával a rendszer megbízhatósága és hatékonysága is növekszik, amely különösen fontos egy olyan kritikus területen, mint a gyógyszerügyi működés.

A felhasználói felület a figyelmeztető üzeneteken túl jól strukturált eredménytáblázatot is biztosít, amely tartalmazza a gyógyszerek nevét, a hozzájuk kapcsolódó tüneteket, az árukat, a lejárat dátumukat és egyéb fontos jellemzőket. Ez a táblázatos formátum lehetővé teszi, hogy a gyógyszerügyi dolgozók gyorsan és könnyen értelmezhető formában kapjanak információt, ami nagymértékben hozzájárul a hatékonysághoz. A felület modern és felhasználóbarát kialakítása csökkenti a betanulási időt, így a dolgozók szinte azonnal használni tudják a rendszert anélkül, hogy hosszas képzésre lenne szükségük. A Streamlit technológia dinamikus és interaktív funkciói további előnyt biztosítanak azáltal, hogy a rendszer minden igényt kielégít, miközben stabilan működik a gyógyszerárak gyakorlati környezetében is. A Streamlit által nyújtott lehetőségek, például az `st.warning()` és az `st.error()` funkciók, segítenek a valós idejű hibakezelés és figyelmeztetések hatékony megvalósításában, így az interfész nemcsak intuitív, hanem megbízható és gyors is. Ez az átgondolt megközelítés biztosítja, hogy a rendszer ne csak a felhasználói élményt javítsa, hanem megfeleljen az egészségügyi környezet magas szintű elvárásainak is.

7.5.2 Backend logika

A rendszer backend logikája felel a keresési folyamatok és az ajánlások pontos és hatékony végrehajtásáért. A kulcsszógenerálás módszer a GPT-4o-mini mesterséges intelligencia modell segítségével a felhasználó által megadott tünetekből és allergiákból releváns kulcsszavakat állít elő. Ez a lépés biztosítja, hogy a keresés szinonimákkal és releváns kulcsszavakkal egészül ki, még akkor is, ha a bevétel hibás vagy hiányos.

A FAISS (Facebook AI Similarity Search) technológia végzi a kulcsszavakhoz kapcsolódó szöveges adatok embeddingekkel való feldolgozását, amelyeket az OpenAI embedding modell generál. Ezek az embeddingek a FAISS indexbe kerülnek, lehetővé téve a gyors és hatékony keresést még nagy adatbázisok esetén is. A hasonlósági mérések alapján a FAISS rangsorolja a gyógyszereket, hogy a felhasználó tüneteire legjobban illeszkedő találatok kerüljenek előre.

A keresési eredmények egy szűrési módszerrel futnak át, amely kizárja azokat a gyógyszereket, amelyek az allergiák mezőiben megadott összetevőket tartalmazzák. Ez a funkció garantálja, hogy a javasolt gyógyszerek nem tartalmazzanak allergén anyagokat. A rendszer összeveti a gyógyszerek összetevőit a felhasználói allergiákkal, és kizárja a nem megfelelő találatokat.

A rangsorolás módszer a GPT-4o-mini modellt használja, amely a szűrt gyógyszerek listáját a felhasználói tünetek relevanciája alapján rangsorolja. A rangsorolás során a rendszer figyelembe veszi a gyógyszerek árát, lejárat dátumát és elérhetőségét, hogy a felhasználók számára a legmegfelelőbb opciók kerüljenek előtérbe.

Az adatmentés modul gondoskodik a keresési eredmények CSV fájlba történő mentéséről, támogatva a dokumentációt és a gyógyszer-tárolási nyilvántartást. A mentett adatok ár, lejárat idő és elérhetőség szerint rendezettek, így megkönnyítik a további feldolgozást.

A rendszer az OpenAI GPT-4o-mini modellt, a FAISS index és a pandas könyvtár modern technológiáira épül, amelyek biztosítják a gyors, pontos és skálázható működést. Az architektúra felépítése lehetővé teszi további fejlesztéseket, például új keresési vagy szűrési kritériumok bevezetését. Ezáltal a rendszer rugalmasan alkalmazkodik a felhasználói igényekhez, miközben a háttérben zajló összetett folyamatokat egyszerűen és intuitívan teszi elérhetővé.

7.5.3 Adatbázis

A rendszer adatbázisa CSV fájl formátumban tárolja a gyógyszeradatokat, biztosítva a könnyű hozzáférhetőséget és karbantarthatóságot. Az adatbázis tartalmazza a gyógyszerek nevét (name), amely az egyértelmű azonosítást szolgálja. A symptoms oszlopban a

gyógyszerek által kezelt tünetek listája található, lehetővé téve a releváns kereséseket. Az ingredients oszlop tartalmazza a gyógyszerek hatóanyagait, amelyeket a rendszer a szűrés során összevet a felhasználó által megadott allergiákkal.

A price-HUF oszlopban a gyógyszerek ára szerepel forintban, ami fontos szerepet játszik a rangsorolásban és az ár-érték arány optimalizálásában. A expiration-date a gyógyszerek lejárat dátumát tartalmazza, biztosítva, hogy csak friss és felhasználható termékek kerüljenek a felhasználók elé. A quantity oszlop a készleten lévő mennyiséget mutatja, amely szintén szempont az ajánlások során.

Ez a jól strukturált adatbázis lehetővé teszi, hogy a rendszer gyorsan és pontosan dolgozza fel a felhasználói lekérdezéseket, miközben egyszerűsíti a rendszer karbantartását és bővítését is.

7.6 Folyamatok részletes leírása

A felhasználó tüneteit egy egyszerű szöveges űrlapon keresztül adja meg, amit a Streamlit keretrendszer kezel. Az adatok helyességét a rendszer valós időben ellenőrzi, és az API-n keresztül továbbítja a további feldolgozáshoz.

A kulcsszógenerálás módszer az OpenAI GPT-4 modellt használja, amely releváns kulcsszavakat hoz létre a megadott tünetek alapján, szinonimákat és kapcsolódó kifejezéseket is figyelembe véve. A fuzzy matching algoritmus a generált kulcsszavakat ellenőrzi és javítja, hogy az esetleges helyesírási hibák ne befolyásolják a keresési eredményeket. Ehhez a fuzzywuzzy könyvtár process.extractOne függvénye biztosít gyors és pontos elírásjavítást.

A FAISS keresőmódszer a kulcsszavakhoz kapcsolódó szöveges adatokat embeddingek formájában dolgozza fel az OpenAI text-embedding-3-small modelljével. Az embeddingek numerikus vektorként kerülnek tárolásra egy FAISS indexben, amely gyors hasonlóságkeresést tesz lehetővé. A keresési lekérdezés numerikus vektorként adja át a FAISS indexnek, amely a legrelevánsabb találatokat rangsorolja. Ez a technológia különösen hatékony nagy mennyiségű adat esetén, mivel jelentősen csökkenti a keresési időt.

A szűrés lépés kizárja azokat a találatokat, amelyek nem relevánsak a felhasználó tünetei szempontjából. Ez a módszer Python pandas könyvtárát használja az adatok manipulációjára és gyors szűrésére. A szűrt eredményeket ezután a rangsorolási módszer dolgozza fel, amely szintén az OpenAI GPT-4 modellt használja a relevanciaelemzéshez. A rendszer kiválasztja a legjobb 5 gyógyszert, amelyeket a tünetek, ár, lejárat idő és elérhetőség alapján további sorrendbe állít.

Az utolsó lépés az eredmények megjelenítése és mentése. A rendszer a gyógyszerek

listáját átlátható táblázatos formában jeleníti meg, amely tartalmazza a gyógyszerek nevét, árát, lejárat dátumát és egyéb releváns információkat. A felhasználók számára ez könnyen értelmezhető, miközben az adatok egy CSV fájlba is mentésre kerülnek, ami archiválható vagy további feldolgozásra használható.

A rendszer felépítése modern technológiákra épül, beleértve az OpenAI GPT-4 modellt, a FAISS indexet és a pandas könyvtárat. Ez a megközelítés biztosítja a gyors, pontos és skálázható működést, amely megbízható eredményeket nyújt a felhasználók számára, miközben a technológiai alapok lehetővé teszik a további fejlesztéseket és a könnyű bővíthetőséget.

7.7 Kockázatok

A biztonság kiemelt prioritást élvez a rendszer fejlesztése során, különösen az érzékeny adatok, például az OpenAI API-kulcsok kezelése terén. Az API-kulcsok bizalmas információk, amelyek kiszivárgása vagy illetéktelen hozzáférése jelentős működési problémákhoz vezethet, beleértve az API-hozzáférés elvesztését vagy a szolgáltatás megszakadását. Ennek megelőzése érdekében az API-kulcsok titkosított formában kerülnek tárolásra, és kizárólag a rendszer működése során érhetők el. A GitHub privát repository-jának "Secrets" funkcióját használjuk a kulcsok biztonságos tárolására. Ez a megoldás biztosítja, hogy az érzékeny adatok csak kódolt formában létezzenek, és kizárólag a CI/CD (Continuous Integration/Continuous Deployment) pipeline futtatása során kerüljenek visszafejtésre. A telepítési folyamat során a kulcsokat közvetlenül az alkalmazásba integráljuk, így nem szerepelnek a forráskódban, és nem kerülnek semmilyen nyilvánosan elérhető helyre.

A GitHub "Secrets" funkciója AES-256 szabványú titkosítást használ az érzékeny adatok védelmére. Ez a titkosítási szabvány jelenleg az egyik legbiztonságosabb és legszélesebb körben használt módszer, amely garantálja, hogy az adatok illetéktelen kezekbe kerülése esetén se legyenek visszafejthetők. Az adatok dekódolása kizárólag a telepítési és futtatási folyamat során történik meg, így az érzékeny információk nem állnak rendelkezésre semmilyen más környezetben. A CI/CD pipeline automatizált folyamatai biztosítják, hogy az API-kulcsok csak a szükséges folyamatok számára legyenek elérhetők, és a telepítési folyamat befejeztével az érzékeny adatok automatikusan eltávolításra kerüljenek a futtatási környezetből.

A titkosítási protokollok alkalmazása nemcsak a tárolás során, hanem az adatok felhasználása közben is fontos szerepet játszik. Az alkalmazás az OpenAI API-ját hívja meg a kulcsszavak feldolgozása és a keresési eredmények rangsorolása érdekében. Fontos megérteni, hogy az OpenAI adatvédelmi irányelvei szintén garantálják az adatok biztonságát. Az OpenAI API-hívások során az alkalmazás által küldött adatokat csak a feldol-

gozás idejére tárolják, és azokat nem használják fel más célokra. Az OpenAI kijelenti, hogy a beküldött adatok nem kerülnek tartós tárolásra, és nem használják azokat modelljeik további tanítására. Ez biztosítja, hogy az adatok biztonságosan és kizárólag az adott művelet céljaira kerülnek felhasználásra.

Az OpenAI továbbá szigorú adatbiztonsági intézkedéseket alkalmaz az adatátvitel és a feldolgozás során. Az adatátvitel biztonságát a TLS (Transport Layer Security) protokoll garantálja, amely az iparág egyik legmegbízhatóbb és legelterjedtebb adatbiztonsági szabványa. Ez a protokoll titkosítja az adatokat az alkalmazás és az OpenAI szerverei között, megakadályozva, hogy az adatokat harmadik felek elfogják vagy módosítsák az átvitel során. A szerveroldali feldolgozás során az OpenAI a legújabb adatbiztonsági technológiákat és szabványokat alkalmazza, hogy megvédje az adatokat az illetéktelen hozzáféréstől.

A rendszer tehát nemcsak az API-kulcsok biztonságos tárolására és kezelésére fordít figyelmet, hanem az OpenAI szervereihez történő adatküldés és -fogadás során is garantálja a biztonságot. Az API-kulcsokat rendszeresen rotáljuk, vagyis időszakonként új kulcsokat generálunk és cserélünk az előzőek helyett. Ez további védelmet nyújt az esetleges kompromittálás ellen, mivel még egy kulcs kiszivárgása esetén is csak korlátozott ideig lenne használható. Az API-hívások naplózása lehetővé teszi a rendellenes működések vagy illetéktelen hozzáférések azonnali észlelését, és szükség esetén gyors beavatkozást.

A rendszer biztonsági stratégiája tehát kiterjed mind a helyi API-kulcsok kezelésére, mind az OpenAI szerverekkel való biztonságos kommunikációra. Az átgondolt titkosítási protokollok és az OpenAI adatvédelmi intézkedései biztosítják, hogy a felhasználók által megadott adatok biztonságban legyenek, miközben a rendszer megfelel a modern adatbiztonsági követelményeknek. Ezáltal garantált a rendszer megbízható és biztonságos működése, amely támogatja a felhasználók igényeit, és biztosítja az érzékeny adatok védelmét az egész folyamat során.

7.8 Nehézségek amikbe ütköztem

Amikor elkezdtem a Streamlit alkalmazás fejlesztését, az egyik fő nehézség az volt, hogy a Streamlit nem kezelte megfelelően az embeddingeket. Először a text-embedding-ada-002 modellt használtam, de hamar rájöttem, hogy ez bizonyos esetekben nem hozta az elvárt eredményeket, különösen nagyobb adathalmazokkal dolgozva. Ezért úgy döntöttem, hogy áttérek a text-embedding-ada-003-large modellre, amely már sokkal hatékonyabbnak bizonyult a szükséges feladatokhoz. Az átállás során azonban némi extra időbe telt a meglévő rendszer módosítása, hogy az új modell kimenetét megfelelően kezelje. Emellett kihívást jelentett az is, hogy a Streamlitben az embeddingekkel végzett műveletekhez optimalizálnom kellett a kódot, hogy az alkalmazás gyors és responszív maradjon. Bár az

átállítás eleinte frusztráló volt, végül elégedett voltam az eredménnyel, mivel az új modell pontosabb és stabilabb teljesítményt nyújtott.

A szavak pontos felismerése és értelmezése komoly kihívást jelentett a fejlesztés során. Próbáltam a lehető leghatékonyabb promptokat írni, hogy a rendszer jól értse a tüneteket, és pontosan generáljon új tüneteket, de hamar rájöttem, hogy ez nem hozta a várt eredményeket. Először a ChatGPT-3.5 modellt használtam, majd áttértem a GPT-4-re, bízva abban, hogy ez javít a pontosságon. Sajnos azonban a probléma továbbra is fennállt; a GPT-4 sem volt teljesen megbízható ebben a feladatban. Bár a GPT-4 használata minimálisan jobb eredményeket hozott, a futásidő jelentősen megnőtt, ami rontotta a felhasználói élményt. Ez a helyzet arra késztetett, hogy új megoldásokat keressek, például szótáralapú megközelítéseket vagy a promptok teljes átdolgozását, hogy javítsam a rendszer pontosságát és hatékonyságát.

Ezt a problémát végül a fuzzy matching rendszer bevezetésével oldottam meg, amely sokkal hatékonyabban ismerte fel a szavakat, még akkor is, ha azok nem pontos egyezést mutattak. A fuzzy rendszer segítségével a tünetek azonosítása és generálása pontosabbá vált, és a rendszer működése is jelentősen javult. Ezzel a megoldással a szófelismerési probléma lezártnak tekinthető, és továbbléphettem a fejlesztés következő lépéseire.

A következő problémám az API-kulcs titkosítása volt, amin sokat gondolkodtam, mivel biztosítani akartam, hogy az érzékeny adatok védettek legyenek. Végül a GitHub repository mellett döntöttem, ahol a kulcsot környezeti változóként kezelhetem, így nem kerül közvetlenül a kódba. Ez a megoldás nemcsak biztonságosabbá tette a rendszert, hanem a verziókövetés előnyeit is kihasználhattam. Így a fejlesztés és a biztonság szempontjait egyszerre tudtam megoldani, anélkül hogy a munkafolyamat bonyolultabbá vált volna.

A top 5 elem megjelenítése eleinte szintén kihívást jelentett. Először egyetlen prompttal próbáltam megoldani, de hosszas próbálkozás után nemcsak a futásidő növekedett jelentősen, hanem a kód is átláthatatlanabbá vált. Emiatt úgy döntöttem, hogy inkább a Pandas könyvtárat fogom használni, mivel az ideális az olyan fájlokhoz, mint amilyeneket én használok. A Pandas lehetővé tette az adatok könnyű rendezését és szűrését, így a top 5 elem megjelenítése egyszerűbbé és gyorsabbá vált, miközben a kód olvashatósága is javult.

8 AZ IRADALOMKUTATÁSRÓLÖSSZEGZÉS

8.1 Magyar összefoglaló

A gyógyszerajánló rendszer egy Python alapú alkalmazás, amely a Streamlit keretrendszerrel használja az intuitív és gyors felhasználói felület biztosítására. A háttérben a FAISS keresőmotor, valamint a ChatGPT által támogatott kulcsszógenerálás és rangsorolás működik, hogy a tünetek alapján a legrelevánsabb gyógyszereket ajánlja. A rendszer a CSV-alapú adatbázist használja a gyógyszerek és összetevők tárolására, miközben a szűrési módszer gondoskodik az allergén összetevők kizárásáról. Az ajánlások relevancia, ár, lejárat idő és elérhető mennyiség alapján kerülnek sorrendbe, így a felhasználók számára a legjobb választási lehetőségeket kínálja. A rendszer gyógyszerészek és egészségügyi szakemberek számára készült, akik szakszerűen tudják alkalmazni az ajánlásokat a páciensek biztonságos és hatékony kiszolgálása érdekében.

Az első és második félév során megszerzett szakmai tapasztalatok és elméleti tudás szervesen kapcsolódnak a rendszer megvalósításához. Az első félévben mélyrehatóan foglalkoztam a nagy nyelvi modellek, például a GPT-3 és a Google Bard működésének és alkalmazási lehetőségeinek tanulmányozásával. Megértettem, hogy ezek a modellek, több százmilliárd paraméterük révén, képesek kontextusfüggő tanulásra anélkül, hogy újra kellene őket képezni. A transzformátor-alapú architektúrák és a figyelmi mechanizmusok lehetővé teszik, hogy a szövegekkel hatékonyan dolgozzanak, ami az egészségügyi alkalmazások terén különösen hasznos.

A második félévben különös figyelmet fordítottam a ChatGPT és a Google Bard összehasonlítására, és megértettem azok egészségügyi alkalmazási lehetőségeit. Míg a ChatGPT folyamatos párbeszédre képes, a Bard valós idejű információkeresést végez az internetről, így naprakész adatokkal szolgál. Az ilyen modellek alkalmazása jelentős előnyöket kínál az adatkezelésben és a döntéshozatal támogatásában, azonban etikai kérdéseket is felvet, például az adatvédelem és pontosság terén. A tanulmányaim során olyan orvosi diagnosztikai alkalmazások működését is vizsgáltam, mint a Docus AI Health Assistant és az Ada, amelyek nagy nyelvi modelleket használnak a betegségek felismerésére. Ezek inspirálták az általam tervezett projektet, amely interaktív tünetlekérdezéssel és pontos diagnózissal kívánja támogatni a gyógyszerészek és a páciensek munkáját.

Összességében az első félévben szerzett technikai ismeretek – különös tekintettel a transzformátor-modellek működésére – és a második félévben elsajátított alkalmazási tapasztalatok megalapozták a projekt megvalósítását. A tanulmányok során szerzett tudás a technológiai lehetőségek és az etikai szempontok közötti egyensúly kialakítására is felhívta a figyelmemet, amely kritikus az egészségügyi rendszerek fejlesztése során.

8.2 Angol összefoglaló az iradalomkutatásról

fordítsd angolra A ChatGPT ezt mondta: ChatGPT The medication recommendation system is a Python-based application that utilizes the Streamlit framework to provide an intuitive and fast user interface. In the background, the FAISS search engine and ChatGPT-supported keyword generation and ranking work together to recommend the most relevant medications based on symptoms. The system uses a CSV-based database to store medication and ingredient information, while the filtering method ensures the exclusion of allergenic ingredients. Recommendations are ranked based on relevance, price, expiration date, and available quantity, offering users the best possible options. The system is designed for pharmacists and healthcare professionals, who can professionally apply the recommendations to ensure safe and effective service for patients.

The professional experiences and theoretical knowledge acquired during the first and second semesters are closely connected to the implementation of this system. In the first semester, I delved deeply into the operation and application possibilities of large language models, such as GPT-3 and Google Bard. I understood that these models, with their hundreds of billions of parameters, are capable of context-sensitive learning without needing retraining. Transformer-based architectures and attention mechanisms enable these models to work efficiently with text, which is particularly useful in healthcare applications.

In the second semester, I focused on comparing ChatGPT and Google Bard and explored their healthcare applications. While ChatGPT is capable of continuous dialogue, Bard performs real-time information retrieval from the internet, providing up-to-date data. The use of such models offers significant advantages in data management and decision support, but it also raises ethical issues, such as data privacy and accuracy. During my studies, I also examined the functionality of medical diagnostic applications like Docus AI Health Assistant and Ada, which use large language models to identify diseases. These inspired the project I designed, which aims to support pharmacists and patients through interactive symptom inquiry and accurate recommendations.

In summary, the technical knowledge acquired in the first semester—particularly regarding transformer models—and the practical experience gained in the second semester laid the foundation for implementing this project. The knowledge gained during these studies also highlighted the importance of balancing technological possibilities with ethical considerations, which is critical when developing healthcare systems.

9 IRODALOMJEGYZÉK

- [1] A. M. Turing, *Computing machinery and intelligence*. Springer, 2009.
- [2] IBM, *What Is Artificial Intelligence (AI)?*, <https://www.ibm.com/topics/artificial-intelligence>, Hozzáférés dátuma: 2021-10, 2021. okt.
- [3] Y. LeCun, Y. Bengio és G. Hinton, „Deep learning”, *nature*, 521. évf., 7553. sz., 436–444. old., 2015.
- [4] M. Jaiwai és tsai., „Automatized Educational Chatbot using Deep Neural Network”, *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, IEEE, 2021, 5–8. old. DOI: 10.1109/ECTIDAMTNCON51128.2021.9425716.
- [5] Unite.AI, *Gépi tanulás vs. mély tanulás – Főbb különbségek*, <https://www.unite.ai/hu/A-g%C3%A9pi-tanul%C3%A1s-%C3%A9s-a-m%C3%A9ly-tanul%C3%A1s-k%C3%B6z%C3%B6tti-f%C5%91-k%C3%BCl%C3%B6nb%C3%A9gek/>, Elérés dátuma: 2024-05-17, 2024. máj.
- [6] A. Hosny, C. Parmar, J. Quackenbush, L. H. Schwartz és H. J. Aerts, „Artificial intelligence in radiology”, *Nature Reviews Cancer*, 18. évf., 8. sz., 500–510. old., 2018.
- [7] H.-Y. Shum, X.-d. He és D. Li, „From Eliza to XiaoIce: challenges and opportunities with social chatbots”, *Frontiers of Information Technology & Electronic Engineering*, 19. évf., 10–26. old., 2018.
- [8] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan és D. S. W. Ting, „Large language models in medicine”, *Nature medicine*, 29. évf., 8. sz., 1930–1940. old., 2023.
- [9] T. A. D’Antonoli és tsai., „Large Language Models in Radiology: Fundamentals, Applications, Ethical Considerations, Risks, and Future Directions”, *Diagnostic and Interventional Radiology*, 30. évf., 2. sz., 80. old., 2024. márc. DOI: 10.4274/dir.2023.232417.
- [10] J. Wei, M. Bosma, V. Y. Zhao és tsai., „Finetuned language models are zero-shot learners”, *arXiv preprint arXiv:2109.01652*, 2021.
- [11] T. A. D’Antonoli és tsai., „Large language models in radiology: fundamentals, applications, ethical considerations, risks, and future directions”, *Diagnostic and Interventional Radiology*, 30. évf., 2. sz., 80–90. old., 2024. márc. DOI: 10.4274/dir.2023.232417.

- [12] T. Tang, H. Lu, Y. E. Jiang és tsai., „Not all metrics are guilty: Improving nlg evaluation with llm paraphrasing”, *arXiv preprint arXiv:2305.15067*, 2023.
- [13] D. Danopoulos, C. Kachris és D. Soudris, „Approximate similarity search with faiss framework using fpgas on the cloud”, *International Conference on Embedded Computer Systems*, Springer, 2019, 373–386. old.
- [14] P. Janardhana Rao, K. Nageswara Rao, S. Gokuruboyina és K. N. Neeraja, „An Efficient Methodology for Identifying the Similarity Between Languages with Levenshtein Distance”, *International Conference on Communications and Cyber Physical Engineering 2018*, Springer, 2024, 161–174. old.
- [15] W. McKinney és P. Team, „Pandas-Powerful python data analysis toolkit”, *Pandas—Powerful Python Data Analysis Toolkit*, 1625. évf., 2015.
- [16] N. Ari és M. Ustazhanov, „Matplotlib in python”, *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, IEEE, 2014, 1–6. old.
- [17] T. Richards, *Getting Started with Streamlit for Data Science: Create and deploy Streamlit web applications from scratch in Python*. Packt Publishing Ltd, 2021.
- [18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever és tsai., „Language models are unsupervised multitask learners”, *OpenAI blog*, 1. évf., 8. sz., 9. old., 2019.
- [19] S. K. Singh és tsai., „Chat GPT & Google Bard AI: A Review”, *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*, IEEE Xplore, 2023, 1–6. old. DOI: 10.1109/ICICAT57735.2023.10263706.
- [20] S. Zhao és tsai., „Carbon Tax vs. Carbon Trading in China: Which Is Better for Promoting Sustainable Development of Remanufacturing Companies?”, *Environmental Science and Pollution Research*, 31. évf., 11. sz., 16710–16724. old., 2024. febr. DOI: 10.1007/s11356-024-32127-1.
- [21] R. Zheng, S. Dou, S. Gao és tsai., „Secrets of rlhf in large language models part i: Ppo”, *arXiv preprint arXiv:2307.04964*, 2023.
- [22] Docus, *Docus AI Symptom Checker: Quick and Accurate Health Insights*, <https://docus.ai/symptom-checker>, Elérés dátuma: 2024. május 17.
- [23] LiveHealthily, *LiveHealthily*, Accessed: 2024-05-17, 2024. máj.
- [24] Ada, *Health. Powered by Ada*, <https://ada.com/>, Elérés dátuma: 2024. május 17.

10 ÁBRAJEGYZÉK

4.1. Gépi tanulás edzése címkézett adatokon, teljesítményértékeléssel.	9
4.2. Mélytanulás kontra klasszikus gépi tanulási	11
5.1. LLM-ek elterjedésének grafikonja	13
5.2. Nyelvi modellek technikai fejlesztési szakaszai.	14
5.3. Kulcsfogalmak az LLM-ekben.	16
5.4. Tokenizáció.	19
5.5. ChatGPT kontra Gemini felhasználói felületen	21
5.6. OpenAI Gyógyszer sorolás API-val	22
7.1. Rendszerterv	25
7.2. A felhasználói felület	27
7.3. Kulcsszógenerálás	29
7.4. FAISS Keresési Eredmények	30
7.5. Top 5 Gyógyszer	32
7.6. Futás idő	34

11 TÁBLAJEGYZÉK

4.1. Mesterséges intelligencia típusai	9
--	---

12 RÖVIDÍTÉSEK

UR	Universal Robots
RAG	Retrieval-Augmented Generation
LLM	Large Language Model
FAISS	Facebook AI Similarity Search
CHATGPT	Generative Pre-trained Transformer
API	Application Programming Interface
CVS	Chorionic villus sampling.