



Mesterséges intelligencia

GKNB_INTM002

Baby menu AI beadandó

2025/2026/1

Készítette:

Székely Áron (ZS811M)

Tartalom

1.	Bevezetés.....	3
2.	Rendszer áttekintése.....	5
2.1	A rendszer célja	5
2.2	A rendszer működésének áttekintése	5
2.3	A rendszer fő komponensei	6
2.4	A működés logikai felépítése (adatfolyam)	7
3.	Követelményanalízis	8
3.1	Funkcionális követelmények	8
3.1.1	Ételadatbázis kezelése	8
3.1.2	Fuzzy értékelő rendszer	8
3.1.3	Ételajánlás evolúciós algoritmussal	8
3.1.4	Gépi tanulás felhasználói visszajelzések alapján.....	8
3.1.5	Kétfüles grafikus kezelőfelület (GUI)	9
3.2	Nem funkcionális követelmények.....	9
3.2.1	Használhatóság.....	9
3.2.2	Megbízhatóság	9
3.2.3	Teljesítmény	9
3.2.4	Rugalmasság és bővíthetőség	9
3.2.5	Összegzés	10
4.	A rendszer felépítése és megvalósítása.....	11
4.1	Rendszerarchitektúra	11
4.2	Adatkezelés és adatforrások.....	12
4.3	Választott technológiák indoklása	12
4.3.1	Python.....	12
4.3.2	Streamlit	13
4.3.3	JSON alapú adatkezelés	13
5.	Adatbázis	14
6.	Alkalmazás implementálása	15
6.1	Fuzzy logika	15
6.2	Egyszerű gépi tanulás	16
6.3	Evolúciós algoritmus	17
6.4	GUI	18
7.	Tesztelés	21
8.	Összegzés	23

1. Bevezetés

A csecsemők és kisgyermekes egészséges fejlődésének egyik legfontosabb alapja a megfelelő táplálkozás. A hozzátáplálás időszaka különösen nagy odafigyelést igényel, hiszen a baba szervezete ilyenkor ismerkedik meg az új ízekkel, állagokkal és tápanyagokkal. Ezzel párhuzamosan egyre nagyobb szerepe van annak is, hogy a szülők olyan ételeket válasszanak, amelyek nemcsak táplálóak és életkori szempontból megfelelőek, hanem biztonságosak is a baba esetleges allergiái vagy érzékenységei mellett.

A szülők számára azonban gyakran nehézséget jelent, hogy minden szempontot egyszerre figyelembe vegyenek:

- milyen alapanyagokat ehet már a gyermek¹,
- mely ételek okozhatnak allergiás reakciót,
- milyen állag felel meg az aktuális fejlettségi szintnek,
- milyen ételeket kedvelt vagy utasított el korábban.

A megfelelő döntés sokszor időigényes és utánajárást igényel, miközben a különböző táplálási ajánlások egymásnak is ellentmondhatnak.

A modern technológia lehetőséget nyújt arra, hogy ezt a folyamatot intelligens, személyre szabott ajánlórendszer támogassa. Jelen dokumentációban egy olyan Python alapú alkalmazást mutatok be, amely a baba életkora, ételpreferenciái, allergiái és a múltbeli visszajelzések alapján állít össze napi étrendajánlást. A projekt célja egy olyan döntéstámogató rendszer létrehozása, amely segíti a szülőket abban, hogy gyorsan és megbízható módon válasszanak a babának megfelelő ételek közül.

A rendszer működése több mesterséges intelligencia alapú megközelítés együttes alkalmazásán alapul. Az ételek értékelését fuzzy logika végzi, amely lehetővé teszi az életkorhoz, az étel állagához (pépes, szilárd stb.) és egyéb tényezőkhez kapcsolódó fokozatos döntéshozatalt. A felhasználói visszajelzések feldolgozásával a rendszer egyszerű gépi tanulási mechanizmust valósít meg, amely idővel képes felismerni a kedvelt és kevésbé kedvelt ételeket. Ezekre az értékelésekre építve az alkalmazás evolúciós algoritmust alkalmaz, amely több lehetséges ételkombináció közül kiválasztja az adott napra legmegfelelőbb menüt.

A rendszerhez tartozik egy könnyen kezelhető grafikus felület is, ahol a szülő:

- ételajánlást kérhet a babának,

¹ <https://www.myplate.gov/life-stages/infants>

- vagy új ételeket vihet fel a rendszerbe.

A dokumentáció célja, hogy részletesen bemutassa a fejlesztett rendszer felépítését, működését, a használt algoritmusokat, valamint a felhasználói felületet. A fejezetek lépésről lépésre ismertetik az alkalmazott módszereket, a megvalósítás során hozott döntéseket és a tesztelési eredményeket.

2. Rendszer áttekintése

A Baby Menu AI egy olyan ajánlórendszer, amely a babatáplálás során felmerülő döntési nehézségek kezelésére fókuszál. A rendszer célja nem egyetlen szempont optimalizálása, hanem több, egymással összefüggő tényező együttes figyelembevétele a napi étrend összeállításakor.

A következő fejezet a megoldandó probléma részletes bemutatására, valamint a rendszerrel szemben megfogalmazott célokra és követelményekre koncentrálni, különös tekintettel azokra a kihívásokra, amelyek a hagyományos, szabályalapú megközelítések számára nehezen kezelhetők.

2.1 A rendszer célja

A Baby Menu AI célja, hogy támogassa a szülőket a hozzátáplálás időszakában azzal, hogy automatikusan megfelelő ételeket ajánl a baba életkora, allergiái, kedvelt ételállagai és korábbi visszajelzései alapján. A rendszer személyre szabott döntéstámogatást nyújt, miközben csökkenti a szülőkre háruló terhet, és egyszerűsíti a napi ételválasztást.

2.2 A rendszer működésének áttekintése

A program működése egymásra épülő lépésekből áll. A folyamat a felhasználói bemenetek megadásával kezdődik, amely során a szülő rögzíti a baba életkorát, az esetleges allergiákat és az ételállagokra vonatkozó preferenciákat.

A rendszer ezt követően a rendelkezésre álló ételeket fuzzy logika segítségével értékeli, ahol az életkori alkalmasság, az állaghoz való illeszkedés, az étel egészségességének becslése, valamint a korábbi visszajelzésekből tanult preferencia együttesen határozzák meg az egyes ételek pontszámát. A program egy függvény segítségével az étel nevében szereplő kulcsszavak alapján becsli meg az étel egészségességét. A rendszer pozitív pontszámot ad zöldségeknek, gyümölcsöknek és sovány húsoknak, míg csökkenti az értéket a kevésbé kedvező ételek esetén. Az eredmény egy 0 és 1 közötti egészségességi érték, amely a fuzzy logikai értékelés egyik bemeneti paramétere. Minta:

```
def _health_membership(food_name):  
    name = food_name.lower()  
    score = 0.0  
    if "zöldség" in name or "főzelék" in name:  
        score += 0.4
```

```
if "gyümölcs" in name or "alma" in name or "körte" in name or "banán" in name:  
    score += 0.3  
if "csirke" in name or "pulyka" in name or "hal" in name or "lazac" in name:  
    score += 0.3  
if "tészta" in name or "palacsinta" in name or "gombóc" in name:  
    score -= 0.2
```

1. kódrészlet: A program a kulcsszavak alapján egészségesnek ítélt ételeket kedvezően pontozza

A fuzzy értékek alapján az evolúciós algoritmus több lehetséges menüt generál és optimalizál, majd kiválasztja a legmegfelelőbb napi ételkombinációt. A felhasználói visszajelzések egyszerű tanulási mechanizmuson keresztül beépülnek a rendszer működésébe, így az ajánlások idővel személyre szabottabbá válnak.

2.3 A rendszer fő komponensei

A legfontosabb modulok:

- **foods.json**

Az alkalmazás ételadatbázisa, amely az egyes ételek alapvető jellemzőit (életkori ajánlás, állag, allergének) tartalmazza, és bemenetként szolgál az ajánlórendszer számára.

- **fuzzy.py**

A fuzzy logikán alapuló értékelő modul, amely az ételek alkalmasságát több tényező együttes figyelembevételével, 0-1 közötti folytonos pontszám formájában határozza meg.

- **feedback.py**

Egy egyszerű gépi tanulási megoldás, amely a felhasználói visszajelzések alapján folyamatosan frissíti az ételekhez tartozó preferenciákat, így az ajánlások idővel egyre jobban igazodnak a felhasználói igényekhez.

- **evolution.py**

Az optimalizációért felelős modul, amely az előzetesen értékelt ételekből genetikus elvek alapján keres egy lehető legjobban illeszkedő napi menüösszeállítást.

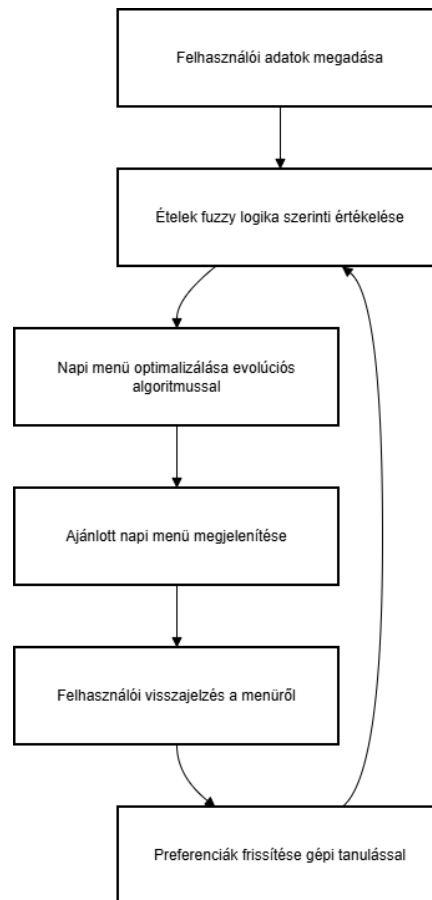
- **recommender.py**

A rendszer központi vezérlőmodulja, amely a felhasználói bemenetek alapján elindítja az ajánlási folyamatot, és összekapcsolja a fuzzy értékelést, a tanulási mechanizmust és az evolúciós algoritmust.

- **gui.py**

A grafikus felhasználói felület, amely lehetővé teszi az alkalmazás egyszerű használatát, az ajánlások megjelenítését és az adatok bevitelét.

2.4 A működés logikai felépítése (adatfolyam)



1. ábra: A program működési logikája (saját készítés)

Az ábra a Baby Menu AI működésének logikai adatfolyamát szemlélteti. A folyamat a felhasználói adatok megadásával kezdődik, amely során a szülő rögzíti a baba életkorát, allergiáit és az ételállagokra vonatkozó preferenciákat. Ezek az adatok bemenetként szolgálnak az ételek fuzzy logika szerinti értékeléséhez, ahol az egyes ételek alkalmassága több tényező együttes figyelembevételével kerül meghatározásra.

A fuzzy pontszámok alapján az evolúciós algoritmus optimalizálja a napi menüt, majd a legmegfelelőbb ételkombináció megjelenik a felhasználó számára. A felhasználó ezt követően visszajelzést adhat a javasolt menüről, amelyet a rendszer eltárol és feldolgoz. A visszajelzések egy egyszerű gépi tanulási mechanizmuson keresztül frissítik az ételekhez tartozó preferenciaértékeket, amelyek beépülnek a következő fuzzy értékelési folyamatba. Ennek eredményeként a rendszer idővel egyre pontosabb és személyre szabottabb ajánlásokat képes készíteni.

3. Követelményanalízis

A rendszer tervezésekor szükséges meghatározni azokat az elvárásokat, amelyek alapján a rendszer megfelelően és biztonságosan működhet. A követelmények két fő kategóriába sorolhatók: funkcionális és nem funkcionális követelmények. A funkcionális követelmények a rendszer által nyújtott szolgáltatásokat írják le, míg a nem funkcionális követelmények a minőségre, teljesítményre és használhatóságra vonatkoznak².

3.1 Funkcionális követelmények

3.1.1 Ételadatbázis kezelése

A Baby Menu AI egyik alapvető funkcionális eleme az ételadatbázis kezelése. A rendszernek képesnek kell lennie egy legalább száz elemből álló ételadatbázis betöltésére, amely az egyes ételek nevét, ajánlott életkorát, állagát és az esetleges allergéneket tartalmazza. A grafikus felületen keresztül a szülőnek lehetősége van új ételek felvitelére is, amelyek mentést követően azonnal megjelennek a későbbi ajánlásokban.

3.1.2 Fuzzy értékelő rendszer

Az ételek értékelését egy fuzzy logikán alapuló modul végzi, amely minden ételhez egy 0 és 1 közötti pontszámot rendel. A pontszám meghatározása során a rendszer figyelembe veszi az életkori megfelelést, az étel állagának illeszkedését a baba preferenciáihoz, az egészségesség becslését kulcsszóalapú névelemzéssel, valamint a felhasználói visszajelzésekből származó tanult preferenciaértéket.

3.1.3 Ételajánlás evolúciós algoritmussal

A rendszer három ételből álló menüket generál, amelyeket a fuzzy pontszámok átlagával értékel. Az algoritmus több generáción keresztül alkalmaz szelekciót, keresztezést és mutációt³, majd kiválasztja a legjobb megoldást, amely megjelenik a felhasználói felületen.

3.1.4 Gépi tanulás felhasználói visszajelzések alapján

² <https://www.geeksforgeeks.org/software-engineering/functional-vs-non-functional-requirements/>

³ Tamilselvi, S. (2022): Introduction to Evolutionary Algorithms.

A felhasználói visszajelzések kezelése egyszerű gépi tanulási mechanizmuson alapul. A szülő a javasolt menüről pozitív vagy negatív értékelést adhat, amelynek hatására az érintett ételekhez tartozó preferenciaértékek módosulnak. Ezek a tanult értékek a fuzzy értékelés részeként kerülnek felhasználásra a későbbi ajánlások során.

3.1.5 Kétfüles grafikus kezelőfelület (GUI)

A rendszer grafikus felülete két elkülönített funkcionális területet biztosít. Az egyik az ételajánlás megjelenítésére és a visszajelzések rögzítésére szolgál, míg a másik lehetőséget ad új ételek strukturált rögzítésére, kiegészítve a helyes adatbevitelhez szükséges tájékoztatással.

3.2 Nem funkcionális követelmények

3.2.1 Használhatóság

A rendszer használhatóságának biztosítása érdekében a grafikus felület kialakítása egyszerű és áttekinthető, így informatikai előképzettség nélkül is könnyen kezelhető. A funkciók logikusan elkülönülnek, és egyértelműen elérhetők a felhasználó számára.

3.2.2 Megbízhatóság

Megbízhatósági szempontból alapvető elvárás, hogy a rendszer ne javasoljon olyan ételt, amely a baba számára megjelölt allergént tartalmaz. A fuzzy értékelésnek minden esetben érvényes, 0 és 1 közötti eredményt kell adnia, míg a gépi tanulási modulnak biztosítania kell a visszajelzések megőrzését.

3.2.3 Teljesítmény

Teljesítmény szempontjából elvárás, hogy az ajánlások rövid időn belül, jellemzően egy másodpercen belül megjelenjenek. Az evolúciós algoritmusnak néhány tucat generáción belül elfogadható megoldást kell találnia, miközben a JSON alapú adatkezelés azonnali visszajelzést biztosít.

3.2.4 Rugalmasság és bővíthetőség

A rendszer felépítése rugalmas és bővíthető. Új ételek, fuzzy szabályok vagy tanulási paraméterek hozzáadása a meglévő struktúra módosítása nélkül is megvalósítható. Az

adatkezelés fájlalapú, ember által olvasható formátumban történik, és a rendszernek képesnek kell lennie az esetleges adatfájl-hibák kezelésére anélkül, hogy működésképtelenné válna.

3.2.5 Összegzés

A Baby Menu AI a követelmények biztosítják, hogy a rendszer megbízhatóan kezelje az ételadatokat, képes legyen személyre szabott ajánlások készítésére, valamint a felhasználói visszajelzések alapján folyamatosan alkalmazkodjon. A meghatározott elvárások egyértelmű alapot teremtenek a rendszer megvalósításához és későbbi értékeléséhez.

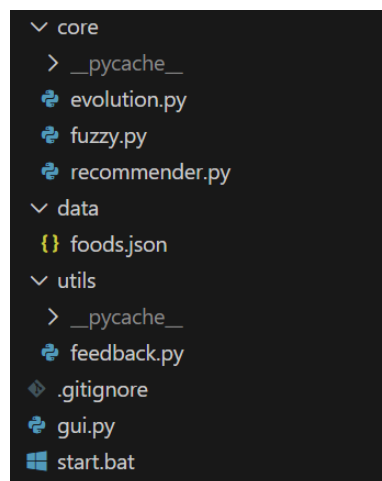
4. A rendszer felépítése és megvalósítása

A Baby Menu AI tervezése során kiemelt szempont volt az átlátható és moduláris felépítés kialakítása. A rendszer nem monolitikus alkalmazásként valósul meg, hanem egymástól jól elkülöníthető komponensekből áll⁴, amelyek mindegyike egy meghatározott feladatot lát el. Ez a megközelítés elősegíti a karbantarthatóságot, a bővíthetőséget, valamint a különböző mesterséges intelligencia módszerek világos elkülönítését.

A fejezet célja a rendszer architektúrájának, adatkezelési megoldásainak és az egyes modulok együttműködésének bemutatása.

4.1 Rendszerarchitektúra

Az ábra a rendszer moduláris felépítését szemlélteti:



2. ábra: A projekt mappastruktúrája

A **core** mappa tartalmazza az ajánlórendszer központi logikáját. Az `evolution.py` az evolúciós algoritmus megvalósításáért felel, amely a napi menü optimalizálását végzi. A `fuzzy.py` modul végzi az ételek fuzzy logika szerinti értékelését, míg a `recommender.py` gondoskodik arról, hogy a rendszer minden része együtt dolgozzon, és elindítja az ajánlások kiszámítását.

A **data** mappában található `foods.json` fájl az ételadatbázist tartalmazza, amely az ajánlórendszer működésének alapját képezi. A **utils** mappa a kiegészítő segédmodulokat

⁴ <https://www.geeksforgeeks.org/system-design/component-based-architecture-system-design/>

foglalja magában, ezen belül a `feedback.py` kezeli a felhasználói visszajelzéseket és a tanult preferenciaértékek számítását.

A rendszer indítását és használatát a gyökérkönyvtárban elhelyezett `gui.py` biztosítja. A `start.bat` fájl a Windows környezetben történő egyszerű indítást segíti elő, lehetővé téve, hogy az alkalmazás konzolos parancsok nélkül is elindítható legyen.

4.2 Adatkezelés és adatforrások

A Baby Menu AI fájlalapú adatkezelést alkalmaz. Az ételek adatait a `foods.json` fájl tartalmazza, amely az egyes ételek nevét, ajánlott életkorát, állagát és az esetleges allergéneket tárolja.

A felhasználói visszajelzések külön fájlban, a `feedback.json`-ben kerülnek rögzítésre. A rendszer ezekből számítja ki az egyes ételekhez tartozó preferenciaértékeket. A JSON formátum választását az egyszerűség, az átláthatóság és a kézi szerkeszthetőség indokolta, mivel a projekt mérete nem igényel komplex adatbázis-megoldást.

4.3 Választott technológiák indoklása

A Baby Menu AI megvalósítása során olyan technológiákat választottam, amelyek jól alkalmasak mesterséges intelligenciához kapcsolódó számítások elvégzésére, ugyanakkor lehetővé teszik egy egyszerűen kezelhető grafikus felület gyors elkészítését is. Fontos szempont volt számomra, hogy a fejlesztés során olyan, már ismert és bevált technológiákkal dolgozzak, amelyekkel korábban is volt tapasztalatom, így a rendszer átlátható felépítésű, könnyen karbantartható és stabil működésű lett.

4.3.1 Python

A rendszer alapjául a Python programozási nyelv szolgál, amely széles körben elterjedt a mesterséges intelligencia és adatfeldolgozás területén. A Python előnye az egyszerű szintaxis, a gyors fejlesztési ciklus és a jól strukturálható moduláris felépítés⁵, amely lehetővé teszi a fuzzy logika, az evolúciós algoritmus és a tanulási komponens elkülönített megvalósítását. A nyelv rugalmassága jól illeszkedik a kísérletező jellegű fejlesztéshez és a szakdolgozati környezethez.

⁵ <https://www.geeksforgeeks.org/artificial-intelligence/python-ai/>

4.3.2 Streamlit

A grafikus felület kialakításához a Streamlit keretrendszer került kiválasztásra. A Streamlit egyik legnagyobb előnye, hogy lehetővé teszi interaktív webes felületek gyors létrehozását kizárólag Python kód segítségével, külön frontend technológiák alkalmazása nélkül⁶. Ez jelentősen könnyíti és gyorsítja a fejlesztést, miközben jól strukturált és áttekinthető felhasználói felület hozható létre.

Kiemelt szempont volt a Streamlit beépített állapotkezelési mechanizmusa (session_state⁷), amely lehetővé teszi a felhasználói interakciók állapotának megőrzését az alkalmazás újratöltése során. Ennek segítségével a rendszer képes kezelni például azt, hogy egy adott menü már értékelésre került-e, vagy hogy a visszajelző üzenetek és ajánlások konzisztensen jelenjenek meg. Az állapotkezelés kulcsszerepet játszik abban, hogy a felhasználói visszajelzések egyszer kerüljenek feldolgozásra, és a rendszer ne kerüljön hibás vagy ellentmondásos állapotba.

4.3.3 JSON alapú adatkezelés

Az ételek és a felhasználói visszajelzések tárolására JSON formátum került alkalmazásra. Ez a megoldás egyszerű, jól olvasható és könnyen kezelhető, különösen kisebb méretű adathalmazok esetén. A JSON formátum használatával az adatok egyszerű fájlokban tárolhatók, így nincs szükség külön adatbázisra, ez a projekt céljainak és méretéhez teljesen megfelelő megoldás.

⁶ <https://www.geeksforgeeks.org/python/a-beginners-guide-to-streamlit/>

⁷ https://docs.streamlit.io/develop/api-reference/caching-and-state/st.session_state

5. Adatbázis

A Baby Menu AI működésének alapját két JSON formátumú adatfájl képezi: a `foods.json`, amely az ételek listáját tartalmazza, valamint a `feedback.json`, amely a felhasználói visszajelzéseket tárolja. A rendszer szándékosan nem alkalmaz összetett adatbázis-megoldásokat, mivel a cél egy egyszerűen kezelhető, otthoni használatra optimalizált alkalmazás volt. A JSON fájlok ember által is könnyen olvashatók, rugalmasak és jól illeszkednek a projekt méretéhez.

A **`foods.json`** egy előre összeállított étellistát tartalmaz, ahol minden ételhez meg van adva a neve, a minimálisan ajánlott életkor, az állag és az esetleges allergének.

A **`feedback.json`** fájl a felhasználói visszajelzések alapján jön létre és frissül. Ebben az alkalmazás minden ételhez eltárolja a pozitív és negatív értékelések számát, valamint az ezekből számított preferenciaértéket. A rendszer tanul a szülői visszajelzésekből, és idővel egyre személyre szabottabb ajánlásokat ad.

Az adatszerkezet egyik legnagyobb előnye a bővíthetőség: a felhasználó új ételeket is felvehet a grafikus felületen keresztül, amelyek azonnal bekerülnek az ajánlási folyamatba.

6. Alkalmazás implementálása

6.1 Fuzzy logika

A Baby Menu AI rendszerben a fuzzy logika feladata az egyes ételek megfelelőségének rugalmas értékelése. A hagyományos, bináris döntések helyett a rendszer 0 és 1 közötti folytonos értékeket használ, mivel a valós életben előforduló táplálkozási döntések - különösen a hozzátáplálás időszakában - nem írhatók le éles határértékekkel⁸.

Első lépésként a rendszer az allergének jelenlétét vizsgálja. Amennyiben az étel olyan allergént tartalmaz, amelyet a felhasználó megjelölt, az étel automatikusan kizárásra kerül, függetlenül más fuzzy szabályoktól.

```
if set(food["allergens"]) & set(allergies):  
    return 0.0
```

2. kódrészlet: Allergént tartalmazó étel automatikus kizárását megvalósító kód

Az életkor megfelelősége nem éles határértékek alapján kerül meghatározásra. Ha a baba eléri az ételhez ajánlott minimális életkort, az életkori tagsági érték maximális, fiatalabb korban pedig fokozatosan csökken.

```
if age >= min_age:  
    return 1.0  
return max(0.0, 1 - (min_age - age) / 3)
```

3. kódrészlet: Minimális életkor elérésének vizsgálata

Az étel állagának értékelése szintén fuzzy módon történik. A rendszer figyelembe veszi a felhasználó által kedvelt ételállagokat, valamint a baba életkorához tartozó ideális állagot. Ez biztosítja, hogy az ajánlások alkalmazkodjanak mind a szülői beállításokhoz, mind a fejlődési szakaszokhoz.

A fuzzy értékelés kiegészül a korábbi felhasználói visszajelzések figyelembevételével. A pozitív visszajelzések növelik, míg a negatív értékelések csökkentik az adott étel megfelelőségi értékét. Ha egy ételhez még nem tartozik visszajelzés, a rendszer semleges értékkel számol.

A végső fuzzy pontszám az egyes részértékek súlyozott összegzésével jön létre, és 0 és 1 közötti értéket vesz fel. Ez az érték fejezi ki az adott étel általános megfelelőségét, és ez szolgál alapul a genetikus algoritmus számára a napi menü összeállításakor.

⁸ Russell, Stuart - Norvig, Peter: Mesterséges Intelligencia: Modern megközelítésben, Budapest, Hungarian Translation Panem Könyvkiadó, 2005.

```
score = (  
    0.4 * age_m +  
    0.25 * texture_m +  
    0.2 * health_m +  
    0.15 * feedback_m  
)
```

4. kódrészlet: A vizsgálati szempontok súlyozása a végső pontszám megállapításához

6.2 Egyszerű gépi tanulás

A rendszerbe egy egyszerű, visszajelzés-alapú tanulási mechanizmus került beépítésre. Ennek célja, hogy a rendszer a felhasználói visszajelzések alapján fokozatosan igazodjon az egyéni preferenciákhoz.

A tanulás alapját az képezi, hogy a felhasználó minden ajánlott menüt értékelhet pozitív („tetszett”) vagy negatív („nem tetszett”) visszajelzéssel. Ezek az értékelések ételenként kerülnek eltárolásra, és a rendszer a későbbi ajánlások során figyelembe veszi őket.

A felhasználói visszajelzések a feedback.json fájlban kerülnek rögzítésre. Minden ételhez eltárolásra kerül a pozitív és negatív értékelések száma, valamint egy preferenciaérték, amely 0 és 1 közötti értéket vesz fel:

```
data[name] = {  
    "positive": 0,  
    "negative": 0,  
    "preference": 0.5  
}
```

5. kódrészlet: Az ételekhez tartozó visszajelzési adatok inicializálása

Az alapértelmezett preferenciaérték 0.5, amely semleges állapotot jelent. Ez biztosítja, hogy a rendszer már az első használatkor is működőképes legyen, még korábbi visszajelzések hiányában is.

A tanulás egy egyszerű statisztikai szabály alapján történik. Minden új visszajelzés után a preferenciaérték újraszámításra kerül a pozitív visszajelzések arányaként:

```
data[name]["preference"] = round(pos / total, 3)
```

6. kódrészlet: A preferenciaérték frissítése a felhasználói visszajelzések alapján

Ez azt jelenti, hogy minél több pozitív visszajelzést kap egy étel a negatívakhoz képest, annál magasabb lesz a hozzá tartozó preferenciaérték. A módszer előnye, hogy a változás lépcsőzetes, így egyetlen értékelés nem torzítja el jelentősen a rendszer működését.

A tárolt preferenciaértékek közvetlenül beépülnek az ajánlási folyamatba. Az ajánlórendszer a fuzzy logika során figyelembe veszi ezeket az értékeket, majd az evolúciós

algoritmus a módosított pontszámok alapján választja ki az optimális napi menüt. Az ajánlás meghívása a következő módon történik:

```
score, menu = evolve(foods, age, allergies, prefs)
```

7. kódrészlet: Az evolúciós algoritmus meghívása és az ajánlott menü lekérése

Ez biztosítja, hogy a felhasználói visszajelzések ténylegesen befolyásolják a jövőbeli ajánlásokat.

6.3 Evolúciós algoritmus

Az evolúciós algoritmus feladata, hogy az előzetesen értékelt ételekből összeállítsa a lehető legjobb napi menüt. Ennek során több különböző menüváltozatot hoz létre, majd ezek közül kiválasztja azokat, amelyek a legjobban illeszkednek a megadott szempontokhoz, és ezeket tovább javítja. A működés alapja hasonló a természetes kiválasztódáshoz: a jobb megoldások „túlélnek”, a gyengébbek fokozatosan kiesnek.

A folyamat elején a rendszer véletlenszerűen állít össze több különböző menüt. Ezek a kezdeti menük eltérő ételkombinációkat tartalmaznak, ami biztosítja, hogy az algoritmus többféle lehetőséget is megvizsgáljon, mielőtt kiválasztja a legmegfelelőbbet.

```
population = []
for _ in range(pop_size):
    population.append(random.sample(foods, menu_size))
```

8. kódrészlet: Kezdeti menüpopuláció véletlenszerű létrehozása

Az egyes menük értékelése egy fitness függvény segítségével történik, amely a fuzzy logika által számított pontszámokra épül. A fitness érték azt fejezi ki, hogy az adott menü összességében mennyire felel meg a baba életkorának, preferenciáinak és egészségügyi szempontjainak.

```
def fitness(menu, age, allergies, prefs):
    scores = [fuzzy_score(f, age, allergies, prefs) for f in menu]
    return sum(scores) / len(scores)
```

9. kódrészlet: A menü fitness értékének kiszámítása a fuzzy pontszámok átlaga alapján

A szelekció során a rendszer a legjobb fitness értékkel rendelkező menüket választja ki, és ezekből hoz létre új megoldásokat. Az új menük keresztezés és mutáció segítségével jönnek létre. A keresztezés során két jó megoldás elemei kombinálódnak, míg a mutáció véletlenszerű változtatásokat vezet be annak érdekében, hogy az algoritmus ne ragadjon le lokális optimumoknál.

A generációk ismétlődése során a populáció minősége fokozatosan javul. A folyamat végén a rendszer a legmagasabb fitness értékkel rendelkező menüt választja ki ajánlásként, amely így az evolúciós keresés eredményeként jön létre.

```
scored_population = [(menu, fitness(menu, age, allergies, prefs)) for menu in population]
best_menu = max(scored_population, key=lambda x: x[1])[0]
return best_menu
```

10. kódrészlet: a program kiértékeli az összes menüt, majd kiválasztja és visszaadja a legmagasabb pontszámú menüt, vagyis az adott feltételeknek legjobban megfelelő megoldást.

6.4 GUI

A Baby Menu AI a Streamlit keretrendszerre épül. A cél az volt, hogy egy olyan webalkalmazást hozzak létre, amelyet bárki képes önállóan használni, informatikai előképzettség nélkül is. A fejlesztett GUI két fő részre, két külön fülre oszlik, amelyek egymástól jól elkülönülő célt szolgálnak: az egyik a napi ételajánlást jeleníti meg, a másik új ételek felvitelére ad lehetőséget.

Baba Étél Ajánló

Ez az alkalmazás abban segít, hogy a baba életkorának és egyéni igényeinek megfelelő ételeket válasszon. Add meg a baba életkorát hónapokban, jelöld meg az esetleges allergiákat és a kedvelt ételállagokat, majd kattints a „Menü ajánlása” gombra. Az ajánlás után visszajelzést adhatsz, amely segíti a rendszer tanulását.

Étel ajánlás

Új étel felvittele

Baba életkora (hónap):

8

-

+

Allergiák:

Choose options

▼

Kedvelt állagok:

püré

×

+

▼

Menü ajánlása

Ajánlott napi menü:

• Zöldséges halpüré (püré, min. 8 hó)

• Zöldséges csirkepüré (püré, min. 8 hó)

• Almapüré (püré, min. 6 hó)

Illeszkedés a megadott adatokhoz: 85.7%

Tetszett a menü

Nem tetszett a menü

3. ábra: A program ételajánló felülete

Az első fül, az „Étel ajánlás”, a rendszer központi funkcionalitását jeleníti meg. Itt a szülő megadhatja a baba életkorát, esetleges allergiáit, valamint azt, hogy milyen ételállagokat részesít előnyben a gyermek. A felület úgy lett kialakítva, hogy minden bemenet egyértelmű,

könnyen kitölthető legyen. Miután a szülő megadta az adatokat, egyetlen gombnyomással elindítható az ajánlási folyamat.

A rendszer ezután megjeleníti a napi ajánlott menüt, amely három ételből áll. Minden étel mellett rövid leírás is található, az ajánlott kortól és az állagtól függően. A felhasználó egyetlen pillantással át tudja tekinteni, hogy a nap folyamán milyen ételeket javasol a rendszer. Az első fülön kap helyet a visszajelzési mechanizmus is: a szülő eldöntheti, hogy tetszett-e neki a menü vagy sem. Ez a visszajelzés azonnal bekerül a gépi tanulási modulba, így a rendszer következő ajánlásai már ennek a döntésnek az eredményét is figyelembe veszik. Ezzel a GUI nem csak egy statikus megjelenítő felület, hanem dinamikusan alakítja a program tanulási folyamatát is.

Baba Étel Ajánló ↗

Ez az alkalmazás abban segít, hogy a baba életkorának és egyéni igényeinek megfelelő ételeket válasszon. Add meg a baba életkorát hónapokban, jelöld meg az esetleges allergiákat és a kedvelt ételállagokat, majd kattints a „Menü ajánlása” gombra. Az ajánlás után visszajelzést adhatsz, amely segíti a rendszer tanulását.

Étel ajánlás Új étel felvitele

Étel neve:

Ajánlott életkor (hónap):

6 - +

Állag:

püré ▾

Allergének:

Choose options ▾

Étel mentése

4. ábra: Új étel felvitelét megvalósító felület

A második fül, az „Új étel felvitele”. Ennek segítségével a szülő bármikor bővítheti az adatbázist saját ételeivel, hiszen minden gyermek egyedi módon étkezik, és a háztartásokban készített ételek változatossága sokkal nagyobb, mint amit egy előre feltöltött lista lefedhet. A felületen található egy rövid tájékoztató, amely elmagyarázza, hogyan érdemes új ételt felvinni, milyen információkat kell megadni, és miért fontos ezek pontos kitöltése. A szülő itt megadhatja az étel nevét, a bevezetéshez ajánlott életkort, az étel állagát, valamint az esetleges allergéneket.

Az új étel felvitele után a rendszer azonnal frissíti a foods.json fájlt, és az új étel a következő ajánlások során már szerepelhet is. a rendszer a felhasználó igényeire szabható és folyamatosan bővíthető igény szerint.

A Streamlit alkalmazás előnye, hogy valós időben, interaktív módon jeleníti meg az eredményeket, így a szülő mindig látja, hogy a bevitt adatok hogyan befolyásolják az ajánlást.

Összességében a grafikus felület nem csupán a program esztétikai eleme, hanem annak egyik kulcsfontosságú komponense: biztosítja, hogy a szülők egyszerűen léphessenek interakcióba az AI-rendszerrel, és közvetlen visszajelzésükkel folyamatosan formálják annak működését.

7. Tesztelés

A program tesztelésének célja a működés helyességének, stabilitásának és megbízhatóságának ellenőrzése volt. Mivel a rendszer több mesterséges intelligencia módszert egyesít, a tesztelés nemcsak az egyes kimenetek ellenőrzésére, hanem a komponensek együttműködésének vizsgálatára is kiterjedt.

A tesztelés első lépéseként az egyes modulokat külön-külön vizsgáltam. A fuzzy logikai komponens esetében ellenőriztem, hogy az életkor, az állag és az allergének figyelembevételével minden esetben 0 és 1 közötti, értelmezhető pontszámot adjon vissza. Az egészségességének megállapítása során azt vizsgáltam, hogy az ételnevekhez kapcsolódó kulcsszavak megfelelően befolyásolják-e az értékelést. A fuzzy modul működése minden teszt során stabilnak bizonyult.

A tanulási mechanizmus tesztelése a felhasználói visszajelzések hatásának ellenőrzésére koncentrált. Vizsgáltam, hogy a pozitív és negatív értékelések megfelelően módosítják-e az ételekhez tartozó preferenciaértékeket, valamint, hogy ezek a változások fokozatosan, torzítás nélkül jelennek-e meg. A tesztek alapján a rendszer az elvárt módon reagált az ismétlődő visszajelzésekre.

Az evolúciós algoritmus működését különböző életkor- és preferenciakombinációk mellett teszteltem. Megfigyelhető volt, hogy a generációk előrehaladtával a menük megfelelőségi értékei javultak, miközben a rendszer nem hozott létre hibás vagy duplikált menüket. Az algoritmus stabilan és következetesen működött.

A grafikus felület tesztelése során a használhatóság és a hibakezelés került előtérbe. Ellenőriztem a hiányos bemenetek kezelését, valamint az új ételek felvitelének folyamatát. A felület minden esetben egyértelmű visszajelzést adott a felhasználónak, és nem vezetett hibás állapothoz.

Az integrációs tesztelés során a rendszer teljes működési folyamata került vizsgálatra. A tesztelési folyamat során kiderült, hogy a felhasználói visszajelzések tárolására szolgáló feedback.json fájl automatikus létrehozása és az abba történő adatmentés nem minden esetben működött megfelelően, különösen az első visszajelzés rögzítésekor. A hiba integrációs tesztek során vált láthatóvá, amikor a visszajelzések nem jelentek meg a későbbi ajánlásokban, annak ellenére, hogy a grafikus felületen az értékelés sikeresnek tűnt. A probléma oka a fájlkezelési logika hiányos állapotkezelése volt, amely nem kezelte megfelelően a nem létező fájl esetét. A hiba javítását követően a rendszer már megbízhatóan hozta létre a feedback.json fájlt, és a

felhasználói visszajelzések következetesen rögzítésre és feldolgozásra kerülnek, ami igazolta a tanulási folyamat helyes működését. Az ajánlások generálása, a visszajelzések feldolgozása és az adatbázis bővítése összhangban működött, és a tanulási folyamat hatása a későbbi ajánlásokban is érzékelhető volt.

Összegzőként elmondható, hogy az elkészült rendszer a tesztelések alapján stabilan és megbízhatóan működik. Az alkalmazott mesterséges intelligencia módszerek egymást kiegészítve biztosítják a következetes ajánlásokat, míg a felhasználói felület támogatja az egyszerű és hatékony használatot.

8. Összegzés

A Baby Menu AI projekt célja egy olyan gyakorlati, könnyen használható ajánlórendszer létrehozása volt, amely képes támogatni a szülőket a hozzátáplálás időszakában felmerülő döntések meghozatalában. A rendszer tervezése során kiemelt szempont volt a biztonság, az életkori megfelelés és a személyre szabhatóság, miközben a megoldás technikai szempontból is átlátható és stabil maradt.

A fejlesztés során bebizonyosodott, hogy az egyszerűbb mesterséges intelligencia megoldások is hatékonyan alkalmazhatók valós problémák kezelésére. A fuzzy logika segítségével a rendszer képes kezelni azokat a bizonytalan, átmeneti helyzeteket, amelyek a hagyományos szabályalapú megközelítések számára nehezen kezelhetők. A visszajelzés-alapú tanulási mechanizmus lehetővé teszi, hogy az alkalmazás idővel alkalmazkodjon a felhasználói szokásokhoz, míg az evolúciós algoritmus hatékonyan kezeli a menüösszeállítás kombinatorikus jellegét.

A grafikus felület kialakítása során az egyszerűség és az érthetőség volt az elsődleges cél. Ennek eredményeként az alkalmazás technikai előképzettség nélkül is használható, ami különösen fontos egy családi, otthoni környezetben alkalmazott rendszer esetében. A felhasználói visszajelzések kezelése nemcsak a tanulási folyamatot támogatja, hanem aktív alakítói lehetünk ezáltal a program működésének.

Tanulásgként levonható, hogy a komplex mesterséges intelligencia módszerek nem feltétlenül igényelnek bonyolult megvalósítást: jól megválasztott, egyszerű eszközökkel is hatékony és hasznos rendszer építhető. A projekt azt is megmutatta, hogy a fuzzy logika, az adaptív tanulás és az evolúciós optimalizáció kombinációja különösen alkalmas döntéstámogató rendszerek kialakítására. A Baby Menu AI moduláris felépítése lehetőséget ad a további fejlesztésre, így a rendszer a jövőben új funkciókkal és fejlettebb tanulási megoldásokkal is bővíthető.

Hivatkozások

[1] *Infant Nutrition*. MyPlate.

Elérhető: <https://www.myplate.gov/life-stages/infants>

(Hozzáférés: 2025. 11. 25.)

[2] *Functional vs Non-Functional Requirements*. GeeksforGeeks.

Elérhető: <https://www.geeksforgeeks.org/software-engineering/functional-vs-non-functional-requirements/>

(Hozzáférés: 2025. 11. 30.)

[3] Tamilselvi, S. (2022): *Introduction to Evolutionary Algorithms*.

Elérhető:

https://www.researchgate.net/publication/364620637_Introduction_to_Evolutionary_Algorithms

(Hozzáférés: 2025. 11. 30.)

[4] *Component-Based Architecture – System Design*. GeeksforGeeks.

Elérhető: <https://www.geeksforgeeks.org/system-design/component-based-architecture-system-design/>

(Hozzáférés: 2025. 12. 01.)

[5] *Artificial Intelligence with Python*. GeeksforGeeks.

Elérhető: <https://www.geeksforgeeks.org/artificial-intelligence/python-ai/>

(Hozzáférés: 2025. 12. 01.)

[6] *A Beginner's Guide to Streamlit*. GeeksforGeeks.

Elérhető: <https://www.geeksforgeeks.org/python/a-beginners-guide-to-streamlit/>

(Hozzáférés: 2025. 12. 07.)

[7] *Session State in Streamlit*. Streamlit Documentation.

Elérhető: https://docs.streamlit.io/develop/api-reference/caching-and-state/st.session_state

(Hozzáférés: 2025. 12. 07.)

[8] Russell, S., Norvig, P. (2005): *Mesterséges intelligencia: Modern megközelítésben*.

Budapest: Panem Könyvkiadó.

Ábrajegyzék

1. ábra: A program működési logikája (saját készítés)	7
2. ábra: A projekt mappastruktúrája	11
3. ábra: A program ételajánló felülete	18
4. ábra: Új étel felvitelét megvalósító felület	19

Kódrészletek

1. kódrészlet: A program a kulcsszavak alapján egészségesnek ítélt ételeket kedvezően pontozza	6
2. kódrészlet: Allergént tartalmazó étel automatikus kizárását megvalósító kód	15
3. kódrészlet: Minimális életkor elérésének vizsgálata	15
4. kódrészlet: A vizsgálati szempontok súlyozása a végső pontszám megállapításához	16
5. kódrészlet: Az ételekhez tartozó visszajelzési adatok inicializálása	16
6. kódrészlet: A preferenciaérték frissítése a felhasználói visszajelzések alapján	16
7. kódrészlet: Az evolúciós algoritmus meghívása és az ajánlott menü lekérése	17

8. kódrészlet: Kezdeti menüpopuláció véletlenszerű létrehozása	17
9. kódrészlet: A menü fitness értékének kiszámítása a fuzzy pontszámok átlaga alapján	17
10. kódrészlet: a program kiértékeli az összes menüt, majd kiválasztja és visszaadja a legmagasabb pontszámú menüt, vagyis az adott feltételeknek legjobban megfelelő megoldást.	18