

**Szegedi Tudományegyetem**  
**Informatikai Intézet**

## **Diplomamunka**

**Kun Réka Bianka**

**2025**

**Szegedi Tudományegyetem  
Informatikai Intézet**

**Mesterséges Intelligencia alapú azonosítás és  
ajánlórendszerek alacsony erőforrású  
környezetekben**

**Diplomamunka**

Készítette:

**Kun Réka Bianka**  
programtervező  
informatikus MSc szakos  
hallgató

Témavezető:

**Kiss-Vetráb Mercedes**  
PhD hallgató

**Szeged  
2025**

## Feladatkiírás

IDE KELL

## Tartalmi összefoglaló

- **A téma megnevezése:**

Mesterséges Intelligencia alapú azonosítás és ajánlórendszerek alacsony erőforrású környezetekben.

- **A megadott feladat megfogalmazása:**

Egy olyan ajánlórendszer megvalósítása, amely a felhasználók érdeklődési köreit, preferenciáit és korábbi aktivitásait figyelembe véve személyre szabott eseményajánlásokat nyújt, elősegítve így a számukra legmegfelelőbb rendezvények gyors és egyszerű megtalálását. Ezáltal nemcsak az események keresése válik egyszerűbbé, hanem a felhasználói élmény is fokozódik, mivel a rendszer lehetővé teszi számukra, hogy gyorsabban rátaláljanak azokra a rendezvényekre, amelyek valóban felkeltik az érdeklődésüket. Továbbá, az arcfelismerési funkció integrálása lehetőséget biztosít arra, hogy a felhasználók gyorsan és egyszerűen lépjenek be az alkalmazásba, elfelejtve a hagyományos bejelentkezési folyamatokat, mint a jelszavak vagy bonyolult hitelesítési lépések. Ezáltal nemcsak kényelmesebbé válik az alkalmazás használata, hanem egy biztonságos és zökkenőmentes élményt is biztosít a felhasználók számára.

- **A megoldási mód:**

Az alkalmazás fejlesztése során fontos volt számomra, hogy a felhasználók személyre szabott, gyors és releváns eseményajánlásokat kapjanak, anélkül hogy az ajánlórendszer működése kizárólag a szerveroldalon történjen. Ennek megvalósításához egy gépi tanulási modellt integráltam közvetlenül az alkalmazásba, amely képes az eszközön futva, valós időben feldolgozni az adatokat és eseményeket kategorizálni. Ez a megközelítés csökkenti a szerverre nehezedő terhelést, gyorsabb válaszidőt biztosít, és fokozza a felhasználói élményt.

- **Alkalmazott eszközök, módszerek:**

Az alkalmazás elkészítéséhez a Flutter mobilalkalmazás-fejlesztési keretrendszert használtam, amely a Dart programozási nyelven alapul. Az adatok kezelésére a Google Firebase adatbázis-szolgáltatását alkalmaztam a Firestore-t. A projekt kódjának fejlesztésére a Google Colabot is igénybe vettem, amely egy könnyen használható, felhő alapú fejlesztői környezetet biztosít, így biztosítva a zökkenőmentes és hatékony munkát a kód írása és tesztelése során.

- **Elért eredmények:**

Egy olyan ajánlórendszer és arcazonosításon alapuló hitelesítés, amely megbízható és felhasználóbarát. Könnyen integrálódik az eszközökkel.

- **Kulcsszavak:**

Deep Learning, modell, arcfelismerés, felügyelt tanulás, ajánlórendszer, embedding.

## **Tartalomjegyzék**

<b>Feladatkiírás .....</b>	<b>3</b>
<b>Tartalmi összefoglaló .....</b>	<b>4</b>
<b>Tartalomjegyzék.....</b>	<b>6</b>
<b>Motiváció.....</b>	<b>8</b>
<b>1. Bevezetés .....</b>	<b>9</b>
1.1 A téma jelentősége és aktualitása.....	9
1.2 A gépi tanulásos ajánlórendszerek szerepe a modern technológiákban .....	9
1.3 Az arcfelismerés szerepe a modern technológiában.....	9
<b>2. A gépi tanulás .....</b>	<b>11</b>
2.1 A gépi tanulás alapjai és alapfogalmai .....	11
2.2 Alapvető algoritmusok és technikák.....	11
2.3 Felhasználási területek.....	13
<b>3. Ajánlórendszerek típusai és alapvető megközelítései .....</b>	<b>14</b>
3.1 Kollaboratív szűrés .....	15
3.2 Tartalom alapú ajánlás .....	17
3.3 Hibrid rendszerek .....	18
3.4 Használt módszerek .....	19
3.4.1 Cosinus hasonlóság.....	20
<b>4. Gépi tanulási módszerek alkalmazása az ajánlórendszerekben .....</b>	<b>23</b>
4.1 Felügyelt tanulás (Supervised Learning) .....	23
4.2 Felügyelet nélküli tanulás (Unsupervised Learning) .....	23
4.3 Mélytanulási megközelítések (Deep Learning).....	24

<b>4.4 Használt módszerek .....</b>	<b>26</b>
<b>4.4.1 MobileNetV2 modell alkalmazása mobil eszközön .....</b>	<b>26</b>
<b>4.4.2 Generatív nyelvi modell alkalmazása .....</b>	<b>29</b>
<b>5. Arcfelismerés .....</b>	<b>32</b>
<b>5.1 Arcfelismerés történeti áttekintése .....</b>	<b>32</b>
<b>5.2 Arcfelismerési rendszerek típusai .....</b>	<b>32</b>
<b>5.2.1 Arcdetektálás.....</b>	<b>33</b>
<b>5.2.2 Arcazonosítás .....</b>	<b>36</b>
<b>5.2.3 Arcverifikáció.....</b>	<b>37</b>
<b>5.2.4 Használt módszerek.....</b>	<b>38</b>
<b>5.3 Arcfelismerési technikák.....</b>	<b>41</b>
<b>5.3.1 Klasszikus képfeldolgozási módszerek.....</b>	<b>41</b>
<b>5.3.2 Modern megközelítések.....</b>	<b>43</b>
<b>Irodalomjegyzék.....</b>	<b>45</b>
<b>Nyilatkozat.....</b>	<b>48</b>
<b>Köszönetnyilvánítás .....</b>	<b>49</b>

## **Motiváció**

A technológiai rohamos fejlődése új lehetőségeket teremt az eseményszervezés területén, így a felhasználói élmény még inkább személyre szabottá válhat.

A felhasználók gyakran elvesznek a rengeteg elérhető esemény között, és nem mindig találják meg azokat, amelyek igazán érdeklik őket. Emellett a bejelentkezési folyamat során is felmerülhetnek nehézségeik, mint a jelszóemlékezés vagy az adatok újra beírása, ami zavaró és időigényes lehet. Ezáltal a felhasználói élmény csökkenthet, és a látogatók könnyen elfordulhatnak az alkalmazástól.

A célom egy olyan ajánlórendszer és arcazonosító rendszer megvalósítása volt, amelyek hatékonyan kezelik ezeket a problémákat a felhasználói élmény növelése érdekében.



# **1. Bevezetés**

## **1.1 A téma jelentősége és aktualitása**

## **1.2 A gépi tanulós ajánlórendszerek szerepe a modern technológiákban**

A gépi tanulós ajánlórendszerek jelentős szerepet játszanak a modern technológiákban, mivel személyre szabott élményeket biztosítanak a felhasználók számára. Ezek az algoritmusok a folyamatosan növekvő adatmennyiséget és a feldolgozási kapacitást kihasználva képesek a felhasználói szokások, preferenciák és viselkedési minták alapján releváns tartalmat, termékeket vagy szolgáltatásokat javasolni.

## **1.3 Az arcfelismerés szerepe a modern technológiákban**

Az arcfelismerés a modern technológiák egyik kiemelkedő és sokoldalúan alkalmazható területe, amely jelentős hatással van számos iparágra és mindennapi életünkre. Az alábbiakban három fő szerepét emelem ki:

### **1. Biztonság és azonosítás**

Az arcfelismerés megbízható és érintésmentes megoldást kínál a személyazonosításra.

- Alkalmazási példák:

- Beléptetőrendszerek, például repülőtereken és vállalati környezetben.
- Okoseszközök feloldása (pl. Face Id).
- Bűnüldözési célok, például gyanúsítottak azonosítása vagy eltűnt személyek felkutatása.

Ezek a rendszerek nemcsak kényelmesebbé teszik a mindennapi tevékenységeket, hanem növelik a biztonságot is.

### **2. Személyre szabott felhasználói élmény**

Az arcfelismerés lehetővé teszi, hogy a technológiai rendszerek a felhasználók egyedi igényeihez igazodjanak.

- Alkalmazási példák:

- Kereskedelmi platformokon az ügyfelek vásárlási szokásai alapján személyre szabott ajánlatokat kínál.

- Szórakoztatóiparban, például streaming szolgáltatások vagy játékok során, az egyéni preferenciák figyelembevételével testreszabott tartalmat biztosít.

Ez a technológia elősegíti az intuitívabb és élvezetesebb felhasználói élmény kialakítását.

### 3. Automatizálás és hatékonyság növelése

Az arcfelismerő rendszerek képesek nagyméretű adatbázisokat gyorsan és pontosan feldolgozni, ami számos iparágban javítja a hatékonyságot.

- Alkalmazási példák:

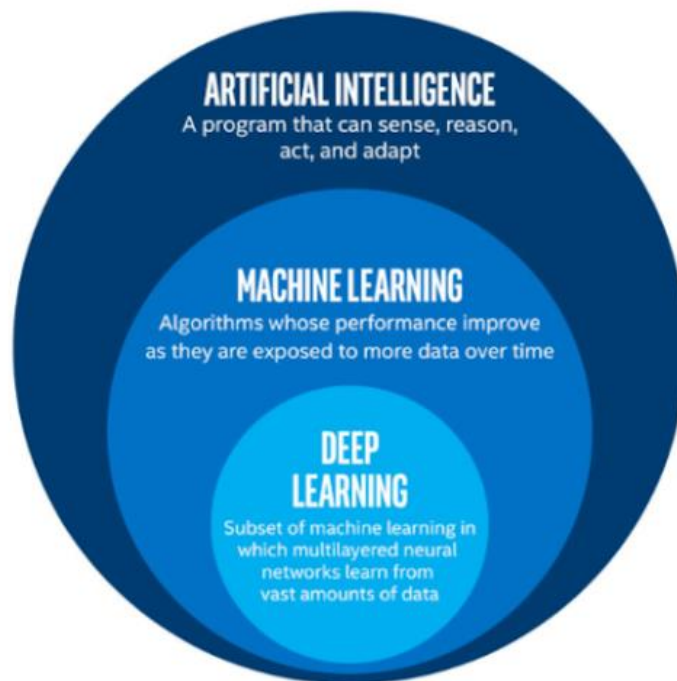
- Egészségügyben a betegek azonosítása és bizonyos betegségek felismerése.
- Marketingben az ügyfelek érzelmi reakcióinak elemzése reklámok vagy termékek iránt.
- Közlekedésben az intelligens megfigyelőrendszerek a forgalom szabályozásában és a balesetek elemzésében.

Az automatizáció révén az arcfelismerés csökkenti az emberi erőforrás-igényt és növeli a folyamatok pontosságát.

## 2. A gépi tanulás

### 2.1 A gépi tanulás alapjai és alapfogalmai

A gépi tanulás a mesterséges intelligencia (MI) egy részterülete, (ábra) amely adatállományokon betanított algoritmusokat használ önállóan tanuló modellek létrehozására. Ezek a modellek képesek előrejelzéseket készíteni és információkat osztályozni emberi beavatkozás nélkül.



2.1 ábra - A gépitánuulás és a Mesterséges Intelligencia kapcsolata

A gépi tanuulást napjainkban számos kereskedelmi célra alkalmazzzák, például termékek ajánlására a vásárlók korábbi vásárlásai alapján, tőzsdei ingadozások előrejelzésére, vagy szövegek fordítására egyik nyelvről a másikra.

### 2.2 Alapvető algoritmusok és technikák

A gépi tanulás alapvető algoritmusai és technikái azok az eszközök, amelyek lehetővé teszik különböző problémák megoldását, mint például osztályozás, előrejelzés, mintázatafelismerés vagy adatbányászat.

Ezek az algoritmusok az adatok feldolgozására, elemzésére és értelmezésére szolgálnak, és az alkalmazott technikák alapvetően meghatározzák, hogy a gépi tanuulási modellek milyen gyorsan és hatékonyan képesek a kívánt eredményeket elérni.

Ilyen algoritmusok a következők:

- **Lineáris Regresszió (Linear Regression)**

Működés: A lineáris regresszió egy alapvető algoritmus, amelyet elsősorban előrejelzések készítésére alkalmaznak. Célja, hogy egy lineáris kapcsolatot találjon a bemeneti változók és a kimeneti változó között. Az algoritmus egy egyenest illeszt az adatokhoz, amely a legjobban képes előre jelezni a kívánt kimenetet.

Alkalmazás: Használható például ingatlanárak, részvények árai vagy bármely olyan probléma esetén, ahol a kimeneti érték folyamatos és lineárisan összefügg a bemeneti változókkal.

- **Logisztikus Regresszió (Logistic Regression)**

Működés: A logisztikus regresszió egy osztályozási algoritmus, amely a kimeneti változót valószínűségi értékként kezeli. Ez a módszer különösen hasznos olyan esetekben, amikor a kimenet két kategóriába sorolható, mint például „igen” vagy „nem”.

Alkalmazás: Alkalmas osztályozási problémák megoldására, például spam-e-mailek felismerésére vagy betegségek előrejelzésére.

- **Neurális Hálózatok (Neural Networks)**

Működés: A neurális hálózatok a biológiai idegrendszer működését modellezzik, és több rétegből állnak, mint például bemeneti, rejtett és kimeneti rétegek. Az adatok áthaladnak ezeken a rétegeken, ahol minden réteg különböző mintázatokat ismer fel, lehetővé téve ezzel a komplex problémák megoldását.

Alkalmazás: Képfeldolgozás, hangfelismerés, természetes nyelv feldolgozás, valamint számos más bonyolult problémák megoldása során alkalmazzák.

Jellemzők: Rendkívül erőteljesek, de jelentős számítási kapacitást igényelnek.

- **K-Legközelebbi Szomszédok (K-Nearest Neighbors, KNN)**

Működés: A KNN egy egyszerű, de hatékony algoritmus, amely a legközelebbi szomszédok elvén alapul. Az új adatpontokat a legközelebbi K szomszédjuk alapján osztályozza a már meglévő adatok figyelembevételével. Az osztályozás a szomszédok többségi osztálya szerint történik.

Alkalmazás: Klasszifikációs és regressziós problémák megoldására, például ajánlórendszerekben vagy ügyfélszegmentálásban.

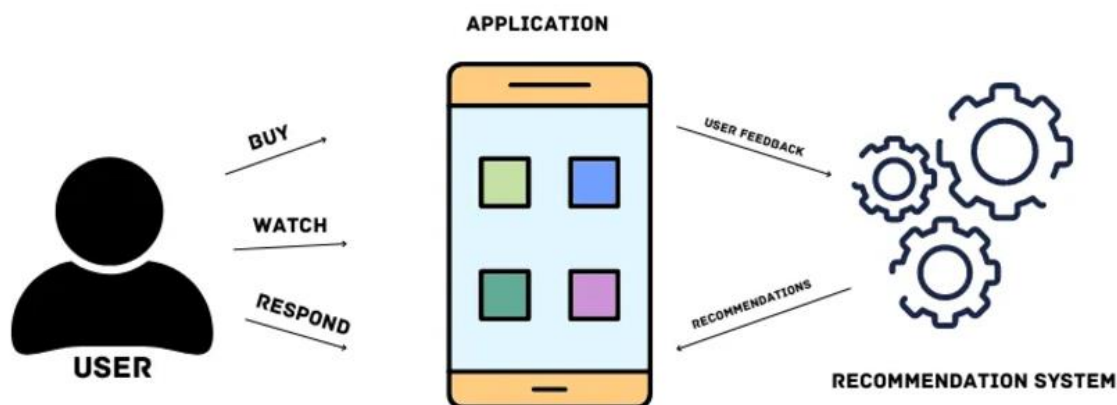
## 2.3 Felhasználási területek

A gépi tanulás napjainkban a mesterséges intelligencia legszélesebb körben használt technológiája. A mindennapi életben számos területen találkozhatunk vele, például:

- Ajánlórendszerek, amelyek személyre szabottan javasolnak termékeket, zenéket vagy műsorokat olyan platformokon, mint az Amazon, a Spotify vagy a Netflix.
- Beszédfelismerő rendszerek, amelyek lehetővé teszik, hogy hangfelvételeket szöveges formátumba alakítsunk át.
- Csalásmegelőző rendszerek a banki szektorban, amelyek automatikusan észlelik és jelzik a gyanús tranzakciókat.
- Önvezető autók és vezetést segítő technológiák, mint például a holtterfigyelő vagy az automatikus vészfékező rendszer, amelyek a közlekedés biztonságát szolgálják.

### 3. Ajánlórendszerek típusai és alapvető megközelítései

Az ajánlórendszerek olyan algoritmusokon és technológiákon alapulnak, amelyek célja, hogy a felhasználóknak személyre szabott javaslatokat kínáljanak különféle termékek, szolgáltatások vagy tartalmak tekintetében. Ezek a rendszerek kiemelten fontosak az e-kereskedelemben, a szórakoztatóiparban, valamint számos egyéb területen, ahol a releváns ajánlások javíthatják a felhasználói élményt és növelhetik az üzleti sikereket.



3.1 ábra - Általános ajánlási rendszer folyamata

Az ajánlórendszerek három fő megközelítést alkalmaznak: kollaboratív szűrés, tartalomalapú ajánlás és hibrid modellek.

A tartalomalapú szűrés és a kollaboratív szűrés egyaránt arra törekszik, hogy személyre szabott ajánlásokat nyújtson a felhasználóknak, azonban eltérő módszereket alkalmaznak.

A tartalomalapú szűrés az események jellemzőit elemzi, és a felhasználó által korábban kedvelt eseményekhez hasonlóakat ajánl, míg a kollaboratív szűrés más, hasonló érdeklődésű felhasználók választásaira épít. A kollaboratív szűrés előnye, hogy változatosabb és újabb eseményeket is képes ajánlani, még akkor is, ha a felhasználó korábban nem találkozott azokkal, míg a tartalomalapú szűrés hajlamos inkább a már ismert érdeklődési körhöz hasonló eseményeket ajánlani. Ugyanakkor a tartalomalapú módszer független a többi felhasználótól, így kevesebb adat mellett is jól működik, míg a kollaboratív szűrés hatékonysága a rendelkezésre álló felhasználói adatok mennyiségétől függ.



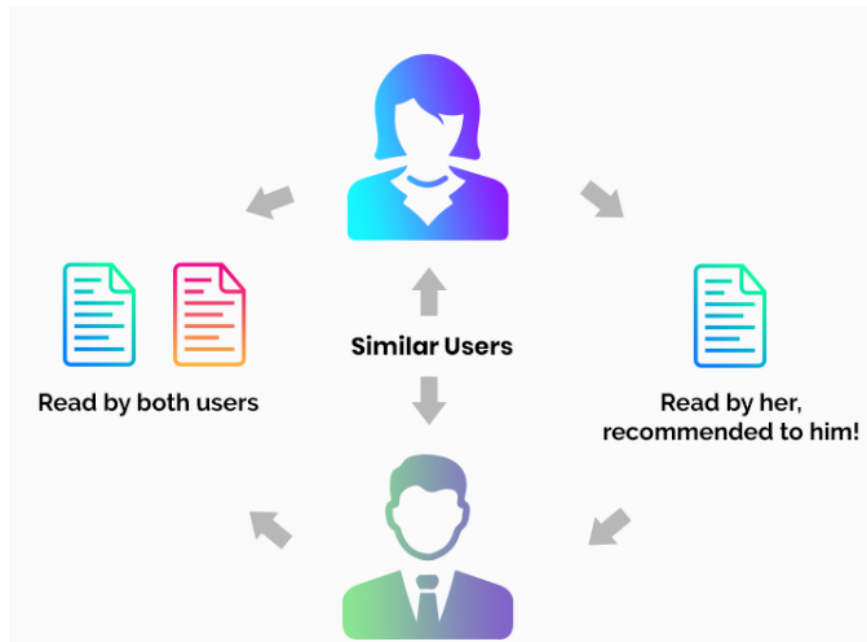
3.2 ábra - Tartalom alapú és Kollaboratív szűrés

A legjobb eredményt gyakran a két módszer kombinálásával, egy hibrid ajánlórendszer kialakításával lehet elérni, amely ötvözi a tartalomalapú szűrés precizitását és a kollaboratív szűrés változatosságát. Így a felhasználók nemcsak az érdeklődési körükhöz leginkább illeszkedő eseményeket találhatják meg könnyebben, hanem új, releváns ajánlásokat is kaphatnak más, hasonló preferenciájú emberek választásai alapján.

### 3.1 Kollaboratív szűrés

A kollaboratív szűrés azon a feltételezésen alapul, hogy a hasonló érdeklődési körű felhasználók hajlamosak hasonló döntéseket hozni.

Az algoritmus a rendelkezésre álló viselkedési adatokat – például vásárlásokat, értékeléseket, kattintásokat vagy tartalomfogyasztási szokásokat – használja, hogy összefüggéseket találjon a felhasználók vagy elemek között. Ezen adatok alapján az algoritmus megjósolja, milyen termékeket vagy tartalmakat találhat vonzónak egy adott felhasználó.

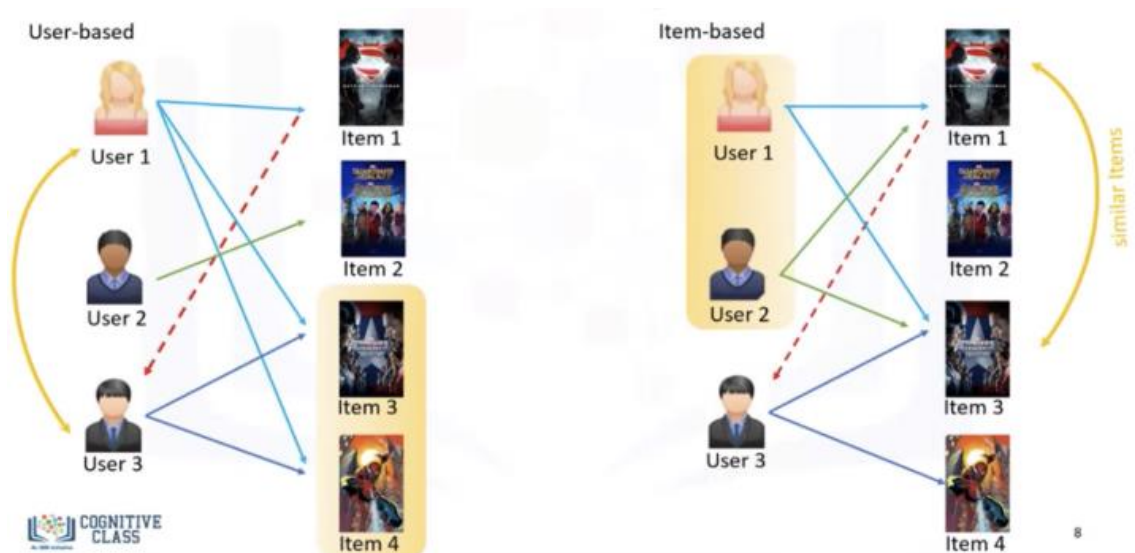


3.3 ábra - Kollaboratív szűrés

A kollaboratív szűrés két fő módszert használhat:

- Felhasználó-alapú megközelítés: Az algoritmus azokat a felhasználókat elemzi, akik hasonló érdeklődési körrel vagy preferenciákkal rendelkeznek, és az ő döntéseik alapján ajánl új elemeket. Például, ha egy felhasználó egy adott filmet kedvelt, akkor olyan filmek kerülnek számára ajánlásra, amelyeket más, hasonló ízlésű felhasználók szintén kedveltek.
- Elem-alapú megközelítés: Ebben a módszerben az algoritmus az elemek közötti hasonlóságokat vizsgálja. Például, ha egy felhasználó vásárolt egy bizonyos típusú laptopot, a rendszer olyan további termékeket javasol, amelyek hasonló vásárlási minták alapján népszerűek más felhasználók körében.





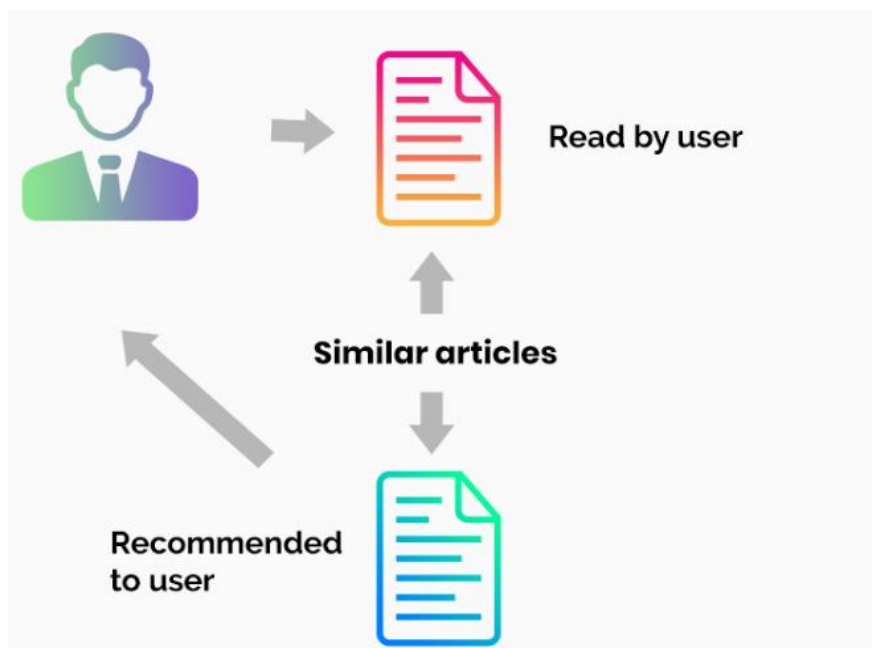
3.4 ábra - Felhasználó és elem alapú megközelítés

A kollaboratív szűrés alapvető eleme a modern ajánlórendszereknek, mivel egyszerű és hatékony megoldást nyújt a személyre szabott ajánlások készítésére anélkül, hogy az ajánlott tartalmakról részletes információval kellene rendelkezni.

### 3.2 Tartalom alapú ajánlás

A tartalomalapú ajánlás azon alapul, hogy az algoritmus a termékek vagy tartalmak tulajdonságait – mint például a cím, szerző, műfaj, kulcsszavak, vagy bármilyen egyéb jellemző – elemzi.

Az algoritmus a felhasználó korábbi interakcióit, például kedvenc termékeit, filmjeit vagy cikkeket, figyelembe véve ajánl új tartalmakat, amelyek hasonlóak azokhoz, amiket már kedvelt vagy megtekintett. A cél az, hogy a felhasználó érdeklődési köréhez passzoló, releváns ajánlásokat nyújtson, figyelmen kívül hagyva más felhasználók preferenciáit.



3.5 ábra - Tartalom alapú ajánlás

A tartalomalapú ajánlási rendszerek széles körben elterjedtek különböző online platformokon, például e-könyv ajánló rendszerekben, mint a Goodreads, ahol az olvasók érdeklődési körének megfelelő könyveket ajánlanak. Hasonlóan, film- és sorozat ajánló rendszerek, mint a Netflix, Hulu vagy Amazon Prime Video, szintén ezt a módszert alkalmazzák, hogy a felhasználók számára releváns tartalmakat kínáljanak a korábbi megtekintések és preferenciák alapján. Ezen kívül hírszolgáltatók, mint a Google News vagy a Flipboard is tartalomalapú ajánlásokat használnak, hogy személyre szabott híreket és cikkeket kínáljanak a felhasználóknak.

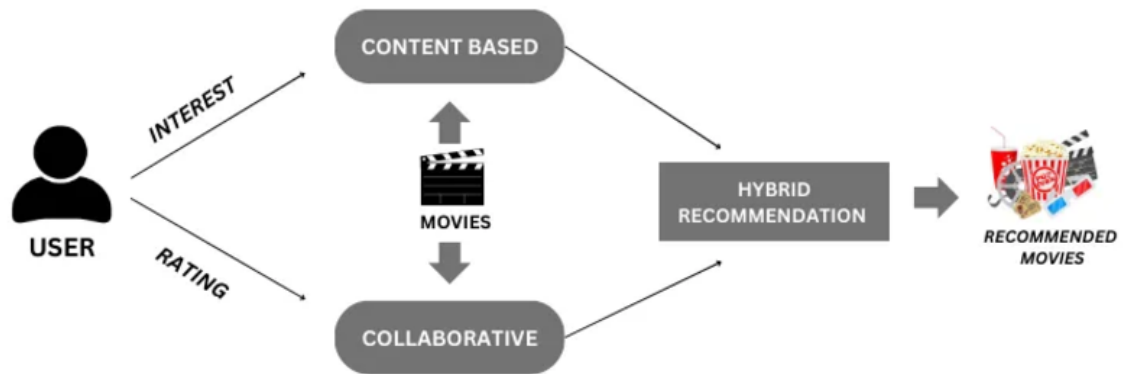
A zenei platformok, mint a Spotify és az Apple Music, szintén alkalmazzák ezt a megközelítést, hogy a felhasználók korábbi zenei preferenciái alapján új zeneszámokat vagy albumokat ajánljanak.

### 3.3 Hibrid rendszerek

A hibrid rendszerek a kollaboratív szűrés és a tartalomalapú ajánlás előnyeit ötvözik, hogy javítsák az ajánlások pontosságát és kiküszöböljék az egyes módszerek hátrányait. Az integráció különféle módon valósulhat meg, például a két megközelítés eredményeinek kombinálásával vagy párhuzamos alkalmazásával, hogy mindkét módszer erősségei érvényesülhessenek.

Ezen kívül, a hibrid rendszerek képesek dinamikusan alkalmazkodni a változó felhasználói preferenciákhoz, és figyelembe venni a felhasználói viselkedést, valamint a

termékek vagy tartalmak specifikus jellemzőit. Ezáltal a rendszer képes a különböző ajánlási források előnyeit egyesíteni, és olyan személyre szabott javaslatokat kínálni, amelyek szélesebb körű és pontosabb élményt nyújtanak.



3.6 ábra - Hibrid ajánlórendszer

Az ilyen rendszerek különösen hasznosak lehetnek azokban az esetekben, amikor egyik megközelítés önállóan nem elegendő, például új felhasználók vagy termékek esetén, vagy ha az ajánlások személyre szabása további finomhangolást igényel.

### 3.4 Használt módszerek

A rendszer megvalósítása során fontos szerepet játszott a megfelelő módszer kiválasztása annak érdekében, hogy a rendszer minél jobban a felhasználók igényeire összpontosítson. A fejlesztett rendszerben a tartalom alapú ajánlás módszere került alkalmazásra, amely lehetővé tette, hogy a felhasználók egyéni preferenciáik és érdeklődési köreik alapján személyre szabott eseményeket kapjanak.

Az ajánlórendszer fejlesztése során két eltérő megközelítést alkalmaztam annak érdekében, hogy a felhasználók számára minél pontosabb és relevánsabb ajánlásokat biztosítsak. Az egyik módszer a Cosinus hasonlóságon alapul, amely a felhasználók korábbi interakcióit és az események jellemzőit összevetve határozza meg a hasonlóság mértékét. A másik megközelítés a MobileNetV2 modell alkalmazása, amely lehetővé teszi a vizuális tartalmak, mint az eseményképek hatékony feldolgozását még alacsony erőforrású eszközökön is.

### 3.4.1 Cosinus hasonlóság

A fejlesztés során először a Cosinus hasonlóság módszerét használtam mivel ez egy hatékony és széles körben alkalmazott eljárás a tartalomalapú ajánlórendszerekben.

A módszer lényege, hogy az események jellemzőit és a felhasználók korábbi interakcióit vektoros reprezentációk (embeddingek) segítségével ábrázolja. Ezeket a vektorokat összehasonlítva számítja ki a hasonlósági értéket, amely megmutatja, mennyire illeszkedik egy adott esemény a felhasználó érdeklődési köréhez.

A Cosinus hasonlóság számításához alkalmazott embeddingek generálása egy Google Colab környezetben lett megvalósítva, ahol a Firestore adatbázisból lekérdezett események leírásait a TF-IDF vektorizáló segítségével numerikus vektorokká alakítom. A leírások alapján generált embeddingek ezután visszakerülnek az adatbázisba, így minden esemény rendelkezik egy egyedi, vektoros reprezentációval, amely lehetővé teszi a pontosabb és relevánsabb ajánlások nyújtását a felhasználók számára. A folyamat automatizálása érdekében a GitHub Actions-t alkalmaztam, amely óránként végrehajtja a számításokat, biztosítva ezzel az adatok folyamatos és naprakész frissítését.

```
collection = firestore_client.collection("events")
docs = collection.stream()

data = []
doc_ids = []

for doc in docs:
    doc_data = doc.to_dict()
    data.append({
        "id": doc.id,
        "description": doc_data.get("description"),
    })
    doc_ids.append(doc.id)

df = pd.DataFrame(data)

tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(df['description'])

embeddings = tfidf_matrix.toarray()

collection = firestore_client.collection("events")

for i in range(len(doc_ids)):
    doc_ref = collection.document(doc_ids[i])

    doc = doc_ref.get()

    if doc.exists and 'embedding_field' not in doc.to_dict():
        doc_ref.update({
            "embedding_field": embeddings[i].tolist()
        })
```

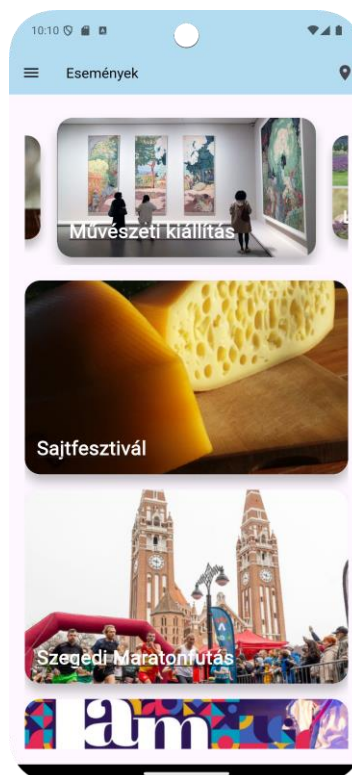
3.7 ábra – Az alkalmazáshoz használt embedding számítás

A Cosinus hasonlóság számítása közvetlenül az alkalmazásban történik, ahol a rendszer a felhasználó által korábban részt vett események leírásaiból létrehozott embeddingeket hasonlítja össze a többi esemény leírását reprezentáló embeddingekkel.

```
class CosineSimilarity {  
  
    double cosineSimilarity(List<double> vec1, List<double> vec2) {  
        double dotProduct = 0.0;  
        double magnitude1 = 0.0;  
        double magnitude2 = 0.0;  
        for (int i = 0; i < vec1.length; i++) {  
            dotProduct += vec1[i] * vec2[i];  
            magnitude1 += vec1[i] * vec1[i];  
            magnitude2 += vec2[i] * vec2[i];  
        }  
        if (magnitude1 == 0 || magnitude2 == 0) return 0.0;  
        return dotProduct / (sqrt(magnitude1) * sqrt(magnitude2));  
    }  
}
```

3.8 ábra – Az alkalmazásban használt Cosinus hasonlóság számítás

A kiszámított hasonlósági értékek alapján az alkalmazás a főképernyőn, a CarouselSlider widget segítségével, dinamikusan jeleníti meg a felhasználó számára ajánlott eseményeket.



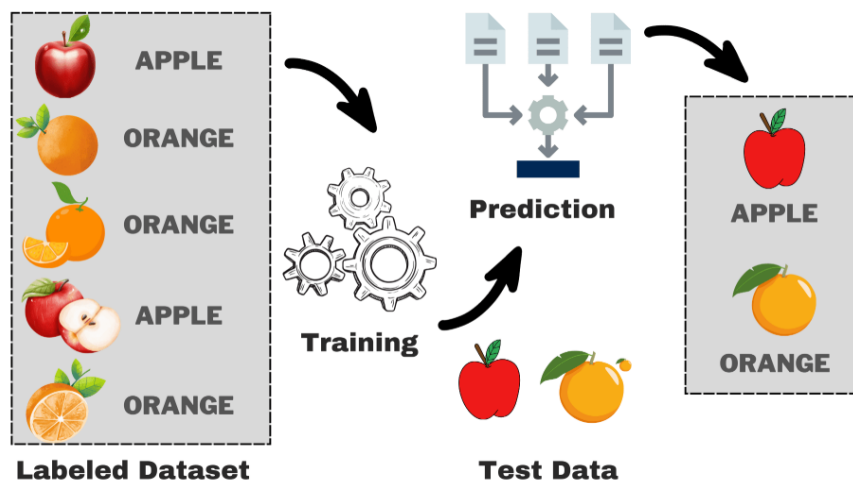
3.9 ábra – Az alkalmazásban megjelenített ajánlott események Cosinus hasonlósággal

A CarouselSlider lehetővé teszi, hogy a felhasználó könnyedén végig pörgessen az ajánlott eseményeken, és az ajánlások folyamatosan frissülnek a felhasználói aktivitások és preferenciák változásainak megfelelően. Az egyes események képekkel és rövid leírásokkal jelennek meg, hogy még könnyebbé váljon a felhasználók számára az érdeklődésüknek leginkább megfelelő események gyors kiválasztása.

## 4. Gépi tanulási módszerek alkalmazása az ajánlórendszerekben

### 4.1 Felügyelt tanulás (Supervised Learning)

A felügyelt gépi tanulás során az algoritmusokat címkézett adathalmazokon tanítják, amelyek az egyes adatokat leíró címkéket tartalmaznak. Ez azt jelenti, hogy az algoritmus olyan adatokkal dolgozik, amelyekhez egy „válasz kulcs” is tartozik, megmutatva, hogyan kell az adatokat értelmezni. Például egy algoritmus gyümölcsökről készült képeket kap, amelyek címkékkel jelzik az egyes gyümölcs típusokat, így később képes lesz új képek alapján azonosítani a gyümölcsöket.



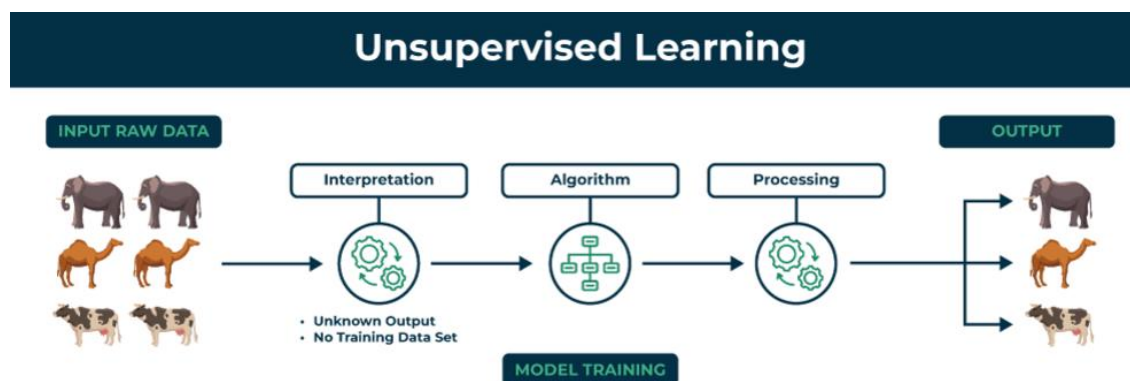
4.1 ábra - Felügyelt tanulás működése

A felügyelt gépi tanulás alkalmazása lehetővé teszi, hogy az algoritmusok az egyes jellemzők és azok kategóriái közötti összefüggéseket tanulják meg, és később ezeket a tudásokat új, még nem látott adatok értékelésére használják. A pontosabb predikciók érdekében a modell gyakran iteratív tanulási folyamaton megy keresztül, ahol folyamatosan finomítja a paramétereit a bemeneti és kimeneti adatok közötti kapcsolat jobb megértéséhez. Az ilyen típusú tanulás széleskörűen alkalmazható például orvosi diagnosztikában, pénzügyi előrejelzésekben vagy bármilyen más területen, ahol fontos a pontos osztályozás vagy előrejelzés.

### 4.2 Felügyelet nélküli tanulás (Unsupervised Learning)

A felügyelet nélküli gépi tanulás olyan módszer, amely címkézetlen adathalmazokat használ az algoritmusok betanítására. Ebben az esetben az algoritmus nem kap előre megadott címkéket, így önállóan kell felismernie a mintázatokat és összefüggéseket az

adatok között külső iránymutatás nélkül. Például egy ilyen algoritmus elemezheti egy közösségi média platform nagy mennyiségű címkézetlen felhasználói adatait, hogy viselkedési trendeket vagy felhasználói csoportokat azonosítson.

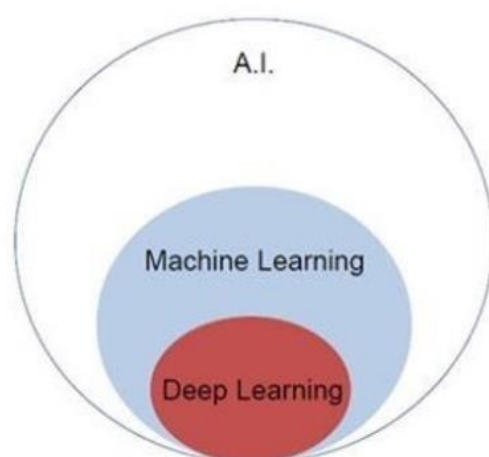


4.2 ábra - Felügyelet nélküli tanulás működése

A felügyelet nélküli tanulás különösen hasznos nagy adathalmazok esetén, ahol az adatok manuális címkézése túl költséges vagy időigényes lenne. Ilyen módszereket gyakran alkalmaznak kutatók és adatelemzők a rejtett struktúrák felfedezésére, piaci szegmensek azonosítására, anomáliák észlelésére, illetve szövegek és képek automatikus csoportosítására. Ez a megközelítés lehetővé teszi az adatok mélyebb megértését anélkül, hogy azokhoz előzetesen címkéket kellene rendelni.

#### 4.3 Mélytanulási megközelítések (Deep Learning)

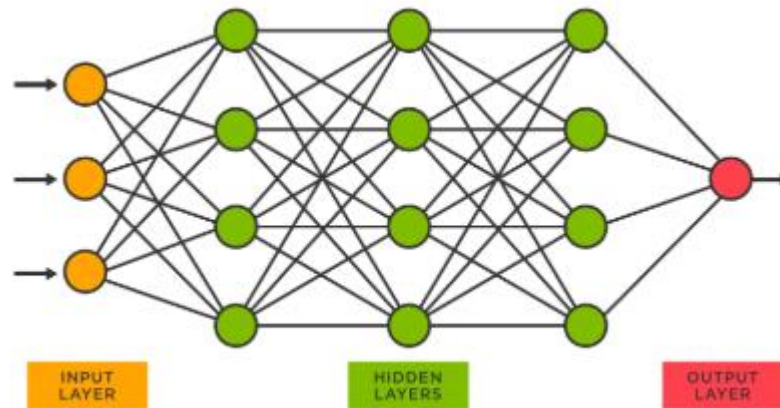
A mélytanulás (Deep Learning) a gépi tanulás egy speciális ága, amely mesterséges neurális hálózatok segítségével old meg bonyolult problémákat.



4.3 ábra - Mélytanulás és a gépi tanulás



A mélytanulás alapját a több rétegből álló neurális hálózatok képezik, amelyek nagy mennyiségű adat feldolgozására és értelmezésére képesek. Ezeknek a modelleknek a célja, hogy önállóan tanuljanak, mintázatok ismerjenek fel és döntéseket hozzanak, mindezt anélkül, hogy előre meghatározott szabályokat kellene alkalmazniuk.



4.4 ábra - Mélytanulási rétegek

A mélytanulási hálózat rétegei:

- **Bemeneti réteg (Input Layer):** Ez a réteg önálló számításokat nem végez, csupán továbbítja az adatokat a következő rétegek felé. A bemeneti réteg minden egyes bemeneti jellemzőhöz – például egy kép pixelértékeihez vagy egy szenzor mérési adataihoz – egy külön neuron kapcsolódik. Az adatokat vektorként vagy mátrixként ábrázolják, majd továbbítják a következő rétegekhez, amelyek a rejtett rétegek. Az előfeldolgozási műveletek, mint például a normalizálás, általában a bemeneti réteg előtt zajlanak, ezzel biztosítva az adatok megfelelő formátumát és minőségét a hatékony tanulási folyamat érdekében.
- **Rejtett rétegek (Hidden Layers):** A rejtett rétegek felelősek az adatok feldolgozásáért és a kívánt kimenet előállításáért. Ezek a rétegek végzik el a legösszetettebb számításokat, mivel minden egyes réteg az előző rétegekből származó kimeneti adatokat használja fel a további feldolgozáshoz. A rejtett rétegek hierarchikus struktúrában működnek: az alsóbb rétegek az egyszerűbb adatjellemzőket, például vonalakat vagy éleket azonosítanak, míg a magasabb rétegek egyre bonyolultabb mintázatok és összefüggések képesek felismerni, mint például tárgyakat, személyeket vagy egész jeleneteket.
- **Kimeneti réteg (Output Layer):** A kimeneti réteg a mélytanulási hálózat utolsó része, amely az előző rétegek által feldolgozott adatokat a kívánt formátumban adja ki. Ez a réteg végzi el a korábbi rétegek által generált eredmények végső

kiértékelését, és azokat a konkrét feladathoz igazítva alakítja át, például osztálycímkék (osztályozás esetén) vagy folytonos értékek (regresszió esetén) formájában. A kimeneti réteg biztosítja a hálózat választ, amelynek meg kell felelnie a várt célértékeknek.

## **4.4 Használt módszerek**

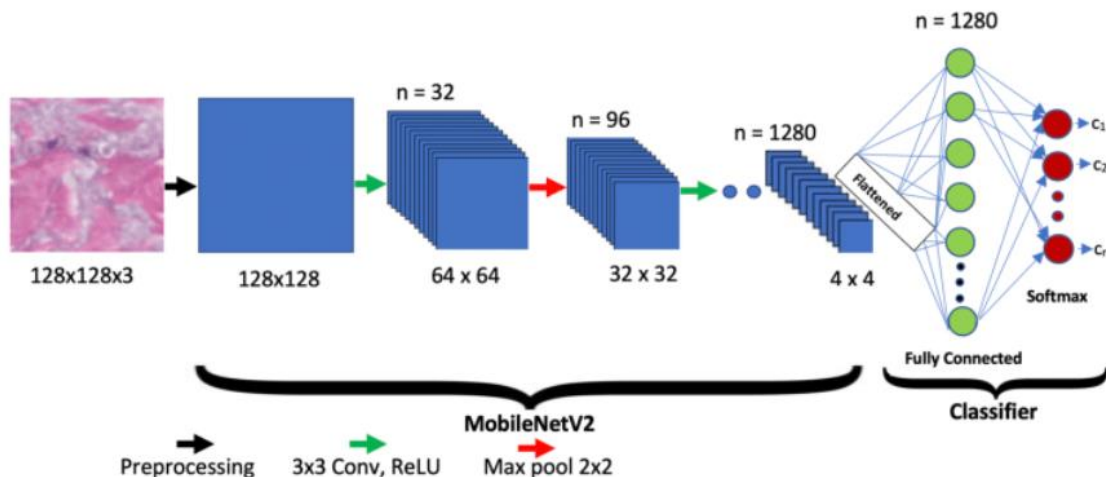
### **4.4.1 MobileNetV2 modell alkalmazása mobil eszközön**

Az alkalmazásban használt MobileNetV2 modell egy előre betanított konvolúciós neurális hálózat (CNN), amelyet a képek hatékony és gyors feldolgozására terveztek, különösen mobil és beágyazott eszközökön történő futtatásra. A modell a mélytanulás (Deep Learning) egyik felügyelt tanulási (Supervised Learning) módszerét alkalmazza, ahol a betanítás során minden képhez ismert címke tartozik. A tanulási folyamat során a modell megtanulja az egyes vizuális jellemzők és a hozzájuk tartozó címkék közötti kapcsolatokat.

A MobileNet V2 architektúráját úgy tervezték, hogy magas feldolgozási teljesítményt nyújtson, miközben alacsony erőforrásigénnyel működjön, így különösen alkalmas mobil és beágyazott rendszerek számára. A hálózat egy 224x224 pixeles RGB képet vár bemenetként, amelyet először egy hagyományos konvolúciós réteg dolgoz fel lépésközzel (stride) 2-vel, 32 csatornára növelve a bemenet mélységét.

A modell legfontosabb építőelemei az úgynevezett inverted residual blokkok, amelyek három fő rétegből állnak:

- Kiterjesztő réteg – egy 1x1 konvolúció, amely megnöveli a csatornák számát a kiterjesztési tényezővel, ezt egy ReLU6 aktivációs függvény követi.
- Depthwise konvolúciós réteg – ez a réteg minden csatornán külön-külön végez térbeli konvolúciót, szintén ReLU6 aktivációval.
- Leképező réteg – egy újabb 1x1 konvolúció, amely visszacsökkenti a csatornaszámot. Ez a réteg már nem használ aktivációs függvényt, lineárisan működik.



4.5 ábra - MobileNetV2 modell architektúrája

A legtöbb blokkban található egy úgynevezett shortcut (átugró) kapcsolat, amely közvetlenül összeköti a blokk bemenetét és kimenetét, amennyiben azok dimenziói megegyeznek. Ez elősegíti a gradiens hatékony áramlását a hálózaton belül, ezáltal javítva a modell tanulási képességét. A rétegek sorát egy globális átlag pooling zárja, amely az aktivációkat 1x1 térbeli méretre sűríti, így egy tömör jellemzővektort hoz létre. A végső, teljesen összekapcsolt réteg pedig az adott képhez tartozó kategóriához rendeli hozzá az osztálypontszámokat.

A MobileNetV2 modellt optimalizált formátumban .tflite (TensorFlow Lite) kiterjesztéssel integráltam az alkalmazásba. Ez a formátum kifejezetten mobilos környezetekre lett kifejlesztve, mivel jelentősen kisebb memória- és számítási kapacitást igényel, mint a hagyományos .pb (TensorFlow Protocol Buffer) vagy .h5 (Keras) modellek. A .tflite formátum lehetővé teszi, hogy a modell közvetlenül a mobilkészítőn fusson, internetkapcsolat nélkül is, így biztosítva gyors válaszidőt. Emellett más bináris modellformátumok is elérhetők, például a .bin kiterjesztésű fájlok, amelyeket különböző gépi tanulási keretrendszerek, mint az OpenVINO, CoreML vagy NCNN, használnak mobil- és edge-eszközökön történő futtatásra. Ezek a formátumok szintén alacsony erőforrásigényű működést és hatékony memóriakezelést biztosítanak, így lehetővé téve a gépi tanulási modellek gyors és valós idejű működését kis teljesítményű környezetekben is.

Az alkalmazásban a modell akkor kerül alkalmazásra, amikor a felhasználó új eseményt hoz létre, és ehhez képet is csatol.

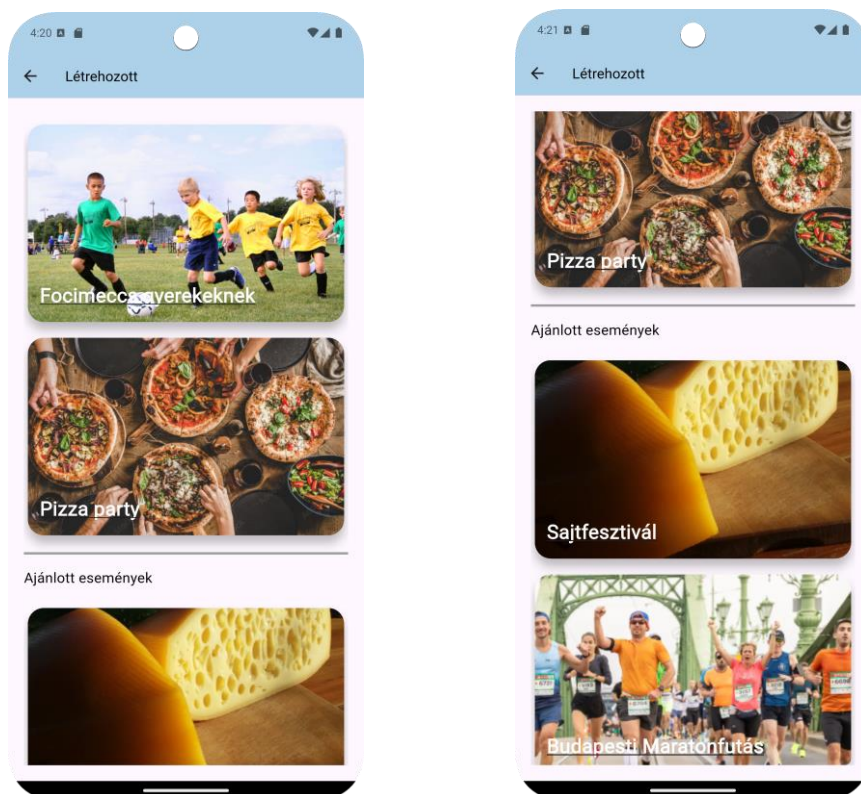
4.6 Az alkalmazás új esemény képernyője

A modell a feltöltött képekből releváns vizuális jellemzőket nyer ki, és ezek alapján az eseményeket automatikusan kategóriákba sorolja, például „Sport”, „Zene” vagy „Művészet”, a képen felismerhető tárgyak alapján. A kategorizálás egy előre meghatározott címke–kategória leképezés szerint történik. A leképezések egy részét a 3. 11. ábra szemlélteti.

```
void initializeLabelMapping() {
    labelToEventCategory = {
        'baseball': 'Sport',
        'basketball': 'Sport',
        'tennis ball': 'Sport',
        'football helmet': 'Sport',
        'soccer ball': 'Sport',
        'rugby ball': 'Sport',
        'golf ball': 'Sport',
        'golf cart': 'Sport',
        'sports car': 'Sport',
        'snowmobile': 'Sport',
        'volleyball': 'Sport',
        'swimming trunks': 'Sport',
        'accordion': 'Music',
        'acoustic guitar': 'Music',
        'bassoon': 'Music',
        'cello': 'Music',
        'drum': 'Music',
        'drumstick': 'Music',
        'electric guitar': 'Music',
    }
}
```

4.7 ábra – Az alkalmazásban használt címke-kategória leképezés

A felhasználó által létrehozott események egy külön felületen jelennek meg az alkalmazásban, ahol az egyes események alatt megjelennek az automatikusan generált ajánlott események is.



4.8 ábra – Az alkalmazásban megjelenített ajánlott események MobileNetV2 modell használatával

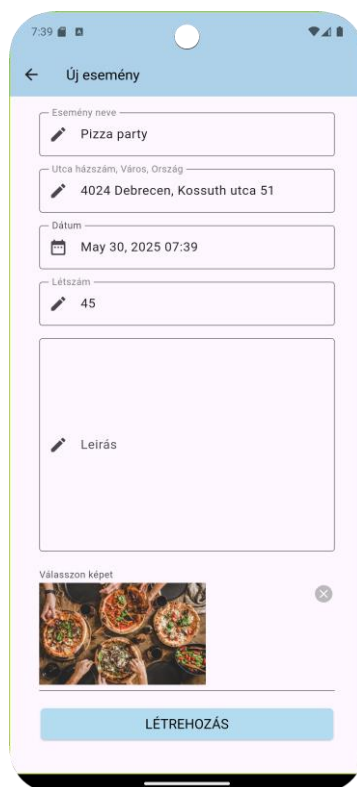
Ezek az ajánlások dinamikusan frissülnek az események kategóriája alapján, amelyet a feltöltött kép tartalma határoz meg. Az alkalmazás olyan további eseményeket jelenít meg, amelyek azonos kategóriába tartoznak, így tematikailag illeszkednek az adott eseményhez. Ez lehetővé teszi, hogy a felhasználók könnyedén felfedezzenek más, hasonló érdeklődési körbe tartozó programokat anélkül, hogy manuálisan kellene keresniük. Ennek eredményeként az alkalmazás személyre szabottabb és felhasználóbarátabb élményt nyújt.

#### 4.4.2 Generatív nyelvi modell alkalmazása

Az alkalmazás egyik különleges funkciója a generált szövegek automatikus előállítás, amelyhez a Puli GPT-2 nevű, magyar nyelvre finomhangolt generatív nyelvi modellt használtam. A modell a híres GPT-2 (Generative Pretrained Transformer 2)

architektúrán alapul, amelyet a Puli AI csapat adaptált és továbbfejlesztett magyar nyelvi környezetre, nagy mennyiségű magyar szövegen történő tanítással.

A fejlesztés során a Google Colab felhőalapú környezetében sablonos kérésformulák (promptok) segítségével generáltam leírásokat előre megadott eseménynemekhez a modell felhasználásával. Az elkészült leírásokat egy JSON fájlba mentettem, ahol minden eseménynévhez hozzárendeltem a hozzá tartozó leírást egy kulcs-érték párokból álló struktúrában. Az alkalmazás ezeket az előre generált leírásokat használja fel azokban az esetekben, amikor a felhasználó új eseményt hoz létre, de nem ad meg saját leírást.



4.9 ábra - Az alkalmazás új esemény képernyője

Ilyenkor a rendszer fuzzy szövegillesztési módszerrel megkeresi az eseménynévhez leginkább hasonlító előre generált esemény nevet, és ha talál megfelelő egyezést, automatikusan beilleszti annak leírását a mezőbe. Ha nincs megfelelő találat, a rendszer hibaüzenetet jelenít meg, amely figyelmezteti a felhasználót, hogy a leírás nem maradhat üresen. A kódban a `StringSimilarity.findBestMatch()` függvény végzi az eseménynév és az elérhető eseménykulcsok közötti összehasonlítást. Ez visszaadja a legjobb egyezést, amelyhez egy ún. rating érték (hasonlósági pontszám) is tartozik. Amennyiben ez az érték 0.4-nél kisebb, a találat nem számít elég jónak, így nem rendel leírást az eseményhez.

```
String? getDescriptionForEvent(String eventName) {
    if (eventDescriptions.isEmpty) {
        return null;
    }
    var bestMatch = StringSimilarity.findBestMatch(eventName, eventDescriptions.keys.toList());
    if (bestMatch.bestMatch.rating! < 0.4) {
        return null;
    }
    final bestMatchName = bestMatch.bestMatch.target;
    return eventDescriptions[bestMatchName];
}
```

4.10 ábra - Az alkalmazásban használt fuzzy szövegillesztés

Ez a megközelítés rugalmasabbá teszi a rendszer használatát, hiszen nem szükséges az eseményneveket pontosan megadni – elég, ha azok kellően hasonlítanak egy már ismert eseményre.

## **5. Arcfelismerés**

### **5.1 Arcfelismerés történeti áttekintése**

Az arcfelismerés kutatása a 20. század közepén indult, amikor megjelentek az első matematikai modellek és algoritmusok, amelyek célja az arcok azonosítása volt. Az 1960-as években kezdődtek az első kísérletek, de akkoriban az arcfelismerő rendszerek még meglehetősen korlátozottak voltak és nem rendelkeztek nagy pontossággal. A 1990-es évek közepére, a számítógépes látás fejlődése és a gépi tanulás elterjedése révén jelentős előrelépések történtek, és az olyan arcfelismerő algoritmusok, mint az eigenfaces, egyre megbízhatóbbá váltak.

A 2000-es évek elejére az arcfelismerési technológia szélesebb körben elterjedt, és számos iparági alkalmazás vált elérhetővé, például biztonsági rendszerekben, biometrikus azonosításban és arcfelismerő szoftverekben, melyeket okostelefonokban és közlekedési rendszerekben is használtak. A legújabb áttörés a mélytanulási algoritmusok bevezetésével történt, amelyek jelentősen növelték az arcfelismerés pontosságát. A konvolúciós neurális hálózatok (CNN) alkalmazása lehetővé tette, hogy a rendszerek automatikusan tanuljanak az arcok jellemzőiről a képeken, így a felismerés még gyorsabbá és megbízhatóbbá vált.

Manapság az arcfelismerés széleskörűen használt technológia, amely jelen van a mobiltelefonokban, közlekedési rendszerekben, biztonsági intézkedésekben és számos más iparági alkalmazásban.

### **5.2 Arcfelismerési rendszerek típusai**

Az arcfelismerési rendszerek különböző technológiai megközelítéseken alapulnak, amelyek különböző módon végzik el az arcok azonosítását és elemzését. Ezek a rendszerek különböző módszereket alkalmaznak, amelyek az adatforrástól és az alkalmazási környezettől függően változnak. A legelterjedtebb típusok közé tartoznak a geometriai alapú rendszerek, a textúra-alapú rendszerek, a statisztikai alapú rendszerek, valamint a mélytanulásra épülő rendszerek.

- 1) Geometriai alapú rendszerek: Ezek a rendszerek az arc jellegzetes geometriai vonásait, mint például a szemek, orr és száj elhelyezkedését elemzik. Az arc felismerése a jellemzők, mint például a távolságok és arányok pontos mérésén

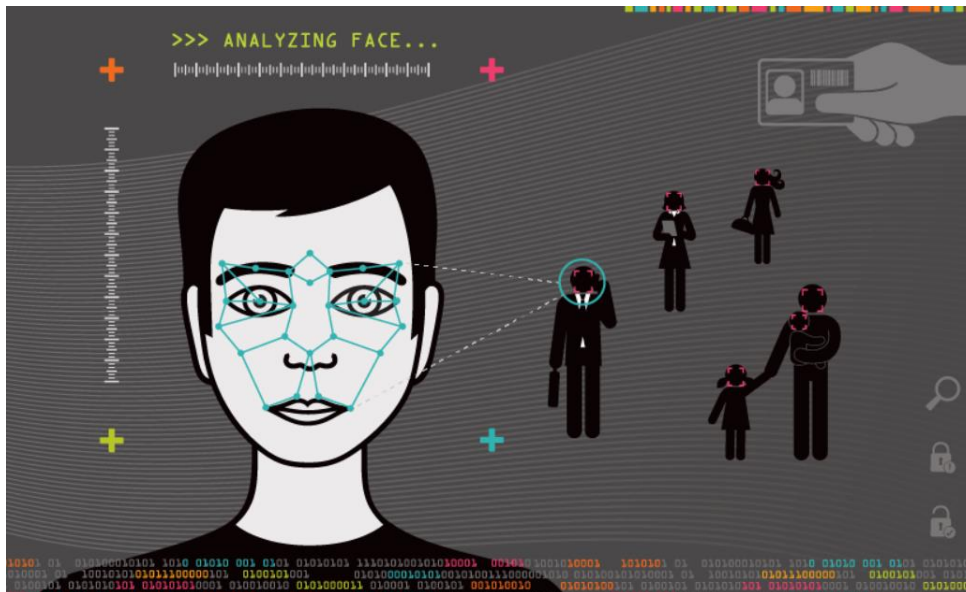


alapul, és ezek az adatok alapjául szolgálnak az arcnak a tanult mintákkal való összehasonlítására.

- 2) Textúra-alapú rendszerek: Az arc bőrének textúráját és mintázatait elemzik, figyelembe véve a finom részleteket, mint például a pórusokat, ráncokat és egyéb bőrszerkezeti jellemzőket. Ezek a rendszerek gyakran használnak lokális bináris minták (LBP) vagy Gabor-szűrők alkalmazásával kinyert jellemzőket, amelyek az arc textúrájának egyedi mintázatát kódolják. Az ilyen megközelítések különösen hasznosak lehetnek olyan helyzetekben, ahol az arc részleges elforgatása vagy elmozdulása történik, mivel a bőrminták stabilabbak lehetnek a geometriai jellemzőkhöz képest.
- 3) Statisztikai alapú rendszerek: Ezek a rendszerek statisztikai módszereket alkalmaznak, mint például az eigenfaces-t, hogy matematikai modelleket alkossanak az arcok jellemzőiről. Az arcokat jellemző adatokat – például a szemek közötti távolságot – statisztikai eloszlások segítségével elemzik. Az eigenfaces technika lényege, hogy a képeket főkomponens-analízissel (PCA) dekomponálja, így a magas dimenziójú képadatokat egy alacsonyabb dimenziós térben ábrázolja, amely az arcok megkülönböztetéséhez szükséges legfontosabb információkat tartalmazza. Ezáltal az arcfelismerés feladata egy vektorokkal végzett összehasonlításra egyszerűsödik.
- 4) Mélytanulás alapú rendszerek: A mélytanuláson alapuló rendszerek, különösen a konvolúciós neurális hálózatok (CNN), közvetlenül a képadatokból tanulják meg az arcok felismerését, anélkül hogy előre meghatározott jellemzők azonosítására lenne szükség. Ezek a rendszerek nagy mennyiségű adat feldolgozására képesek, és összetett mintázatok felismerésére is alkalmasak, például az arc különböző vonásainak, arckifejezéseinek és a megvilágítási különbségeknek a kezelésére.

### **5.2.1 Arcdetektálás**

Az arcdetektálás egy olyan technológiai folyamat, amelynek célja az emberi arcok automatikus azonosítása képeken, videókon vagy élő kamerafelvételeken. Ez a folyamat nem tévesztendő össze az arcfelismeréssel, mivel itt a rendszer kizárólag az arcok jelenlétét és helyzetét határozza meg, anélkül hogy az egyes arcokat konkrét személyekhez kapcsolná.



5.1 ábra - Arcdetektálás

Az arcdetektáló szoftverek mesterséges intelligenciát és gépi tanulási algoritmusokat, valamint statisztikai elemzést és képfeldolgozást használnak annak érdekében, hogy az emberi arcokat felismerjék nagyobb képeken, és elkülönítsék őket az arcokon kívüli objektumoktól, mint például tájak, épületek vagy más emberi testrészek. Az arcdetektálás megkezdése előtt az elemzett média előfeldolgozása történhet a minőség javítása érdekében, valamint a felismerést zavaró objektumok eltávolítása.

Az arcdetektáló algoritmusok jellemzően az emberi szemek keresésével kezdődnek, mivel ezek az egyik legegyszerűbben felismerhető jellemzők. Ezt követően próbálják azonosítani az arc egyéb jellemzőit, mint például a szemöldököt, a száját, az orrot, és az orrlyukakat. Miután az algoritmus úgy ítéli meg, hogy arcot talált, további teszteket végez annak megerősítésére, hogy valóban arcot észlelt.

A nagy pontosság biztosítása érdekében az algoritmusokat nagy adatbázisokon tanítják, amelyek több százezer pozitív és negatív képet tartalmaznak. A tanulás javítja az algoritmusok képességét abban, hogy meghatározzák, van-e arc a képen, és pontosan hol helyezkednek el azok határai.

Az arcdetektálásra számos szoftver létezik. Ezek a szoftverek folyamatosan fejlődtek, miközben egyre fejlettebb technológiákat integráltak a pontosabb eredmények és jobb teljesítmény elérése érdekében.

Az egyik első sikeres módszer a Viola-Jones algoritmus volt, amely egy modellt tanít arra, hogy miként lehet megkülönböztetni az arcokat az egyéb objektumoktól. Bár ez a megközelítés még mindig széles körben használatos valós idejű arcfelismerési

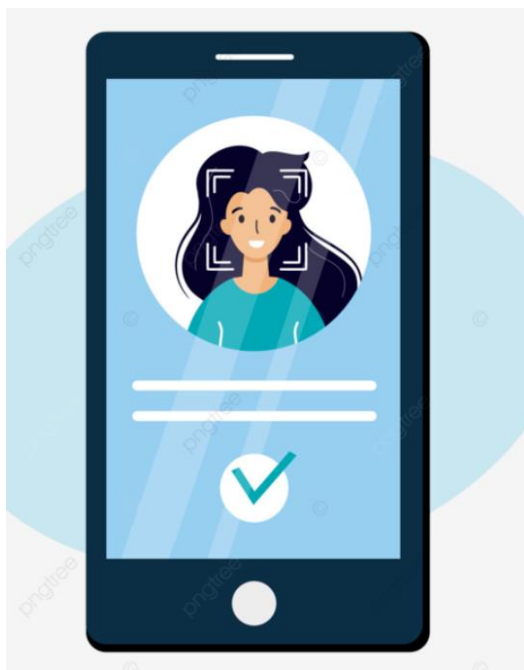
alkalmazásokban, problémákat okoz, ha az arcok takarva vannak, vagy ha nem megfelelő irányban vannak elhelyezve.

A mesterséges intelligencia (AI) technológiák fejlődésével az arcdetektálás képességei is jelentősen javultak. Manapság ezen rendszerek gépi tanulást és mélytanulást alkalmaznak az arcok felismerésére. Emellett konvolúciós neurális hálózatokat (CNN) is alkalmaznak, amelyek a képi adatokat elemzik, és lineáris algebrát használva azonosítják a képen lévő mintázatokat és jellemzőket.

A CNN-alapú arcdetektálás egyik megoldása a Regionális CNN (R-CNN), amely lokalizálja és osztályozza az objektumokat a képeken, majd javaslatokat generál a potenciális arcvonásokra. Ezek a javaslatok a kép olyan területeire összpontosítanak, amelyek hasonlítanak más területekhez, például a szemek pixelált területeihez. Ha a szemek régiója egyezik egy másik szem régióval, az algoritmus felismeri a találatot. A legújabb arcdetektáló megközelítések közé tartozik a Fast R-CNN és a Faster R-CNN is.

Az R-CNN és annak továbbfejlesztett változatai egyik legnagyobb problémája, hogy bonyolulttá válhatnak, és hajlamosak túltanulásra, azaz zajos adatokat illesztenek be ahelyett, hogy a kívánt arcvonásokat tanulnák meg. A CNN-alapú megközelítések másik problémája, hogy gyakran szűk keresztmetszetekbe ütköznek, mivel a rendszer kétszer fut végig a CNN-en: először a régiók javaslatainak generálására, majd az objektumok detektálására.

A Single Shot Detector (SSD) módszer segít orvosolni ezt a problémát, mivel csak egyszer fut át a hálózaton a képen lévő objektumok felismeréséhez. Ennek eredményeként az SSD gyorsabb, mint az R-CNN, de hátránya, hogy nehezebben képes kis arcokat vagy távolról lévő arcokat felismerni.



*5.2 ábra - Arcdetektálás okostelefonoknál*

Az okostelefonok és táblagépek gyakran használják az arcdetektálást a kamerák autofókusz funkciójának aktiválására fényképezéskor és videózáskor. Emellett számos mobil eszköz helyettesíti a jelszavakat vagy PIN-kódokat arcdetektálással. Például az Apple iPhone képes arcdetektálást alkalmazni a telefon feloldásához, ha egy engedélyezett felhasználó próbál hozzáférni.

### **5.2.2 Arcazonosítás**

Az arcazonosítás egy olyan technológiai folyamat, amely lehetővé teszi az emberek azonosítását az arcvonásaik alapján. Míg az arcdetektálás csak az arcok észlelésére korlátozódik egy képen, az arcazonosítás az arcfelismerést is magában foglalja, vagyis a rendszer azonosítja az arcot egy előre rögzített adatbázisban szereplő személyek közül.

Az arcazonosítás számos területen hasznos, például biztonsági rendszerekben, banki alkalmazásokban, mobiltelefonokon és a bűnüldözésben, mivel segíthet a jogosulatlan belépések megelőzésében vagy a személyek azonosításában nagy tömegben. Az arcazonosító rendszerek gépi tanulást és mélytanulást alkalmaznak, hogy folyamatosan javítsák felismerési képességeiket, figyelembe véve az egyedi arcvonásokat, például az orr, a szemek, a száj és egyéb jellemzők elhelyezkedését és távolságát.

### 5.2.3 Arcverifikáció

Az arcverifikáció egy biometrikus azonosítási folyamat, amelynek célja annak ellenőrzése, hogy egy adott személy valóban ő-e, az arca alapján. A folyamat során az arckép, amelyet egy személy benyújt, összehasonlításra kerül egy előzetesen rögzített referenciaképpel, például egy adatbázisban tárolt képpel. Az arcverifikáció tehát nem az egyes személyek azonosítására szolgál, hanem annak megerősítésére, hogy az arc valóban egy előzőleg regisztrált személyhez tartozik.

Ezt a technológiát számos területen alkalmazzák, például biztonsági rendszerekben, mobiltelefonoknál, banki alkalmazásoknál, illetve olyan platformokon, amelyek a személyes adatokat védik, és lehetővé teszik a felhasználói hozzáférés engedélyezését vagy elutasítását. Az arcverifikációs rendszerek gyakran gépi tanulást és mélytanulást használnak, hogy javítsák a felismerés pontosságát és sebességét.



5.3 ábra - Arcverifikáció mobiltelefonon

Az arcverifikációs rendszerek folyamatosan fejlődnek, hogy még pontosabbak és megbízhatóbbak legyenek. A modern technológiák, mint a mélytanulás és a konvolúciós neurális hálózatok (CNN), lehetővé teszik a rendszer számára, hogy az arcok jellemzőit még finomabb részletekben, például a szemek közötti távolság, az orr és a száj elhelyezkedése alapján elemezze. Ezenkívül az arcverifikáció képes alkalmazkodni az arc különböző elhelyezkedéseihez, szögeihez, fényviszonyaihoz és az esetleges arckifejezésekhez is. A megfelelő környezetben és megfelelő minőségű

képeken az arcverifikáció egy rendkívül gyors és hatékony módszerré válik a személyazonosításra, mivel nem szükséges fizikai eszközöket, például kártyákat vagy jelszavakat használni.

A technológia egyre inkább elterjedt, mivel az emberek egyre inkább keresnek kényelmesebb és biztonságosabb módokat a digitális eszközökhöz való hozzáféréshez. A bankok, pénzügyi szolgáltatók és mobiltelefon-gyártók is egyre inkább integrálják az arcverifikációs megoldásokat az ügyfelek védelme és a tranzakciók biztonságának növelése érdekében. Azonban, ahogy minden biometrikus azonosítási technológia, úgy az arcverifikáció is etikai és adatvédelmi kérdéseket vet fel. Az adatkezelés átláthatósága, a személyes adatok védelme és a felhasználói beleegyezés biztosítása kulcsfontosságú tényezők, amelyek meghatározzák a jövőbeni alkalmazások biztonságát és elfogadottságát.

#### 5.2.4 Használt módszerek

Az arcfelismeréses hitelesítés megvalósításához az alkalmazásban a local\_auth csomagot használtam, amely az eszköz operációs rendszerének beépített biometrikus hitelesítési megoldásait veszi igénybe, mint az iOS-en a Face ID-t, Androidon pedig az elérhető arcfelismerő rendszert. A csomag támogatja az eszközön történő helyi autentikációt, így lehetőséget biztosít arcfelismerés, ujjlenyomat, PIN-kód, jelszó vagy minta használatára a felhasználó azonosításához.

```
Future<UserModel> authenticateUser(UserModel userModel) async {
    bool isAuthorized = false;
    try {
        errorMessages = [];
        isAuthorized = await localAuthentication.authenticate(
            localizedReason: faceIdAuthentication,
        );
    } catch (e) {
        if (e.toString().isNotEmpty) {
            errorMessages = [e.toString()];
        } else {
            errorMessages = [standardErrorMessage];
        }
    }
    if (isAuthorized) {
        isUserAuthorized = true;
        String? userId = userLoginSingleton.getRememberedUserId();
        String? userJson = userLoginSingleton.getRememberedUserData();
        String? token = userLoginSingleton.getRememberedUserToken();
        if (userJson != null && userId != null && token != null) {
            UserDTO userDTO = UserDTO.fromJson(jsonDecode(userJson), userId);
            userModel = UserModel.fromDTO(userDTO);
        }
    } else {
        isUserAuthorized = false;
    }
    return userModel;
}
```

5.4 ábra - Az alkalmazásban használt arcfelismerés autentikáció

A Face ID az Apple által fejlesztett korszerű arcfelismerő technológia, amelyet az iPhone és iPad Pro készülékeken alkalmaznak a biztonságos és felhasználóbarát azonosítás érdekében. A rendszer központi eleme a TrueDepth kamera, amely több összetevőből áll: egy infravörös pontprojektor több mint 30 000 láthatatlan pontot vetít az arcra, egy infravörös kamera rögzíti ezek elhelyezkedését, és egy infravörös megvilágító biztosítja a megfelelő fényviszonyokat. Ezek az adatok egy háromdimenziós térképet alkotnak az arcról, amelyet a készülék biztonságos alrendszere, a Secure Enclave dolgoz fel és hasonlít össze a korábban tárolt mintával. A Face ID működése során a rendszer neurális hálózatokat alkalmaz az arcfelismeréshez. Az infravörös képek és pontminták alapján egy matematikai modellt hoz létre az arcról, amelyet összevet a regisztrált adatokkal. Ez a megközelítés lehetővé teszi a rendszer számára, hogy alkalmazkodjon az arc megjelenésének változásaihoz.

Az Androidos arcfelismeréses hitelesítés a BiometricPrompt API használatával történik, és a működése attól függ, hogy az adott eszköz milyen hardverrel és szoftverrel rendelkezik. Az arcfelismerést két fő kategóriába sorolhatjuk: 2D-alapú (kameraalapú) és 3D-alapú (mélységérzékelős) megoldásokra.

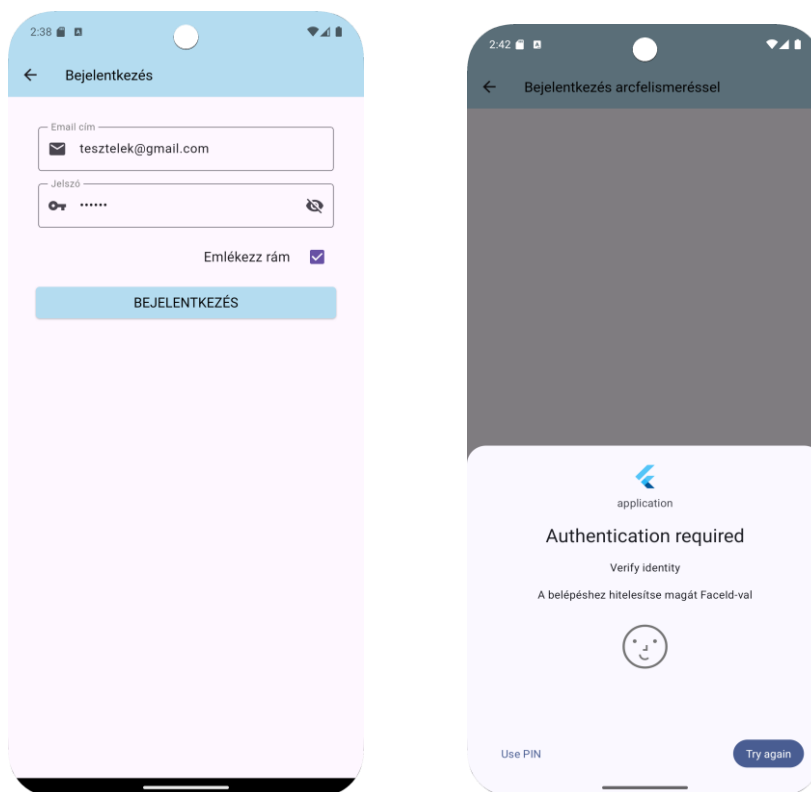
- 2D-alapú arcfelismerés hagyományos kamerát használ az arc kétdimenziós képének rögzítésére. Ez gyors, de kevésbé biztonságos, és könnyebben megtéveszthető például egy fényképpel. A 2D-s arcfelismeréses modellek gyakran egyszerűbb képfeldolgozási eljárásokat alkalmaznak, mint például a Haar Cascade Classifier vagy Eigenfaces.
- 3D-alapú arcfelismerés speciális szenzorokat, például infravörös kamerát és mélységérzékelőt alkalmaz, hogy az arc térbeli szerkezetét is figyelembe vegye, ezáltal nagyobb biztonságot nyújt. A 3D-s arcfelismerési megoldások általában mélytanulási modelleket, például konvolúciós neurális hálózatokat (CNN) használnak. Az ilyen típusú rendszerek például a Google Pixel 4-en alkalmazott 3D mélységérzékelést és infra kamerát, amelyek egy neurális hálózat segítségével elemzik az arc jellemzőit és azok térbeli elhelyezkedését.

Ezek a 3D-s megoldások fejlettebbek és kevésbé érzékenyek a manipulációkra, mint a 2D-s modellek, mivel az arc háromdimenziós struktúráját is figyelembe veszik. A mélytanulási modellek (például CNN) lehetővé teszik a pontosabb arcazonosítást a bonyolultabb, több jellemzőt figyelembe vevő algoritmusokkal. Ezáltal az Androidos arcfelismerés rendszerei a készülék típusától és az alkalmazott hardver-támogatástól

függően különböző modelleket használhatnak, mindkét kategóriában alkalmazva a megfelelő biometrikus technológiai megoldásokat a biztonságos és gyors autentikáció érdekében.

A mobilalkalmazás multiplatform kialakításának köszönhetően mind iOS, mind Android rendszereken működik, és képes kihasználni a platformok beépített arcfelismerő (biometrikus) hitelesítési megoldásait. Ennek köszönhetően a hitelesítés gyors, kényelmes és biztonságos, így a felhasználók könnyedén kihasználhatják a natív biometrikus azonosítás előnyeit a bejelentkezéskor.

Az alkalmazásban az arcfelismeréses belépés úgy működik, hogy amennyiben a felhasználó a bejelentkezés során aktiválja az „Emlékezz rám” funkciót, az alkalmazás biztosítja, hogy legközelebb, amikor újra megnyitja azt, ne kelljen ismét bejelentkeznie. Ha az alkalmazás bezárása után nem jelentkezik ki, a következő alkalommal az arcfelismerés segítségével automatikusan hozzáférhet az alkalmazáshoz.



5.5 ábra - Az alkalmazás bejelentkezés arcfelismeréssel felülete

Ez a megoldás lehetővé teszi, hogy a felhasználók gyorsan és egyszerűen hozzáférjenek az alkalmazáshoz, miközben biztosítja a kényelmes és biztonságos autentikációt. A biometrikus azonosítás segítségével az alkalmazás automatikusan felismeri a felhasználót, így a belépés minden alkalommal zökkenőmentes és gyors.



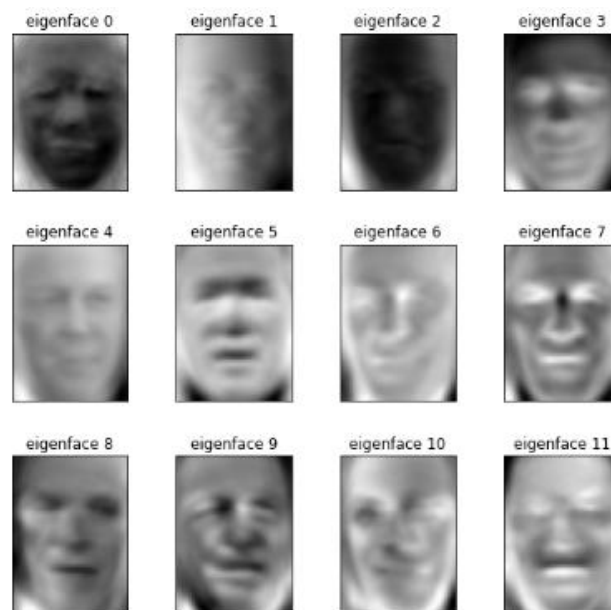
Ezáltal a felhasználóknak nem kell minden alkalommal újra megadniuk a jelszavukat, hiszen a Face ID vagy arcfelismerés gondoskodik a hitelesítésről.

### 5.3 Arcfelismerési technikák

#### 5.3.1 Klasszikus képfeldolgozási módszerek

A klasszikus képfeldolgozási módszerek az arcfelismerés kezdeti fejlesztési szakaszainak alapját jelentették. Ezek a technikák a kép különböző jellemzőinek manuális kinyerésére és feldolgozására összpontosítanak. Néhány elterjedten alkalmazott módszer:

- **Eigenfaces:** Egy arcfelismerési technika, amely az arcképek matematikai reprezentációján alapul. A módszer az arcképek közös jellemzőit (azaz az "eigenfaces"-eket) használja az arcok azonosításához. Az alapötlet az, hogy minden arc egy egyedülálló kombinációját adja a fő komponenseknek, amelyeket a főkomponens-analízis (PCA) segítségével nyernek ki.



*Eigenfaces arcaazonosítás*

Ez a módszer hatékony, de érzékeny lehet a változó környezeti tényezőkre, például világítási körülményekre és arckifejezésekre.

- **Fisherfaces:** egy fejlettebb arcfelismerési technika, amely a lineáris diszkriminancia analízis (LDA) elvén alapul, és az Eigenfaces módszer továbbfejlesztéseként jött létre. Míg az Eigenfaces a főkomponens-analízist (PCA) használja a képek dimenziócsökkentésére, a Fisherfaces célja, hogy az

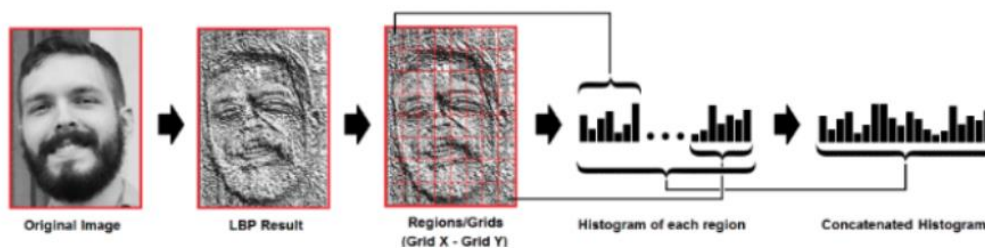
egyes arcok közötti különbségeket és az egyes osztályok közötti elválasztást maximalizálja, miközben minimalizálja a különbségeket egy-egy osztályon belül.



*Fisherfaces arcazonosítás*

Ez a technika különösen hasznos lehet olyan alkalmazásokban, ahol az arcfelismerés nagy pontosságot igényel, és ahol az arcok közötti különbségek nemcsak a tisztán geometriai jellemzők alapján jelentkeznek, hanem a vizuális és statisztikai tulajdonságok alapján is.

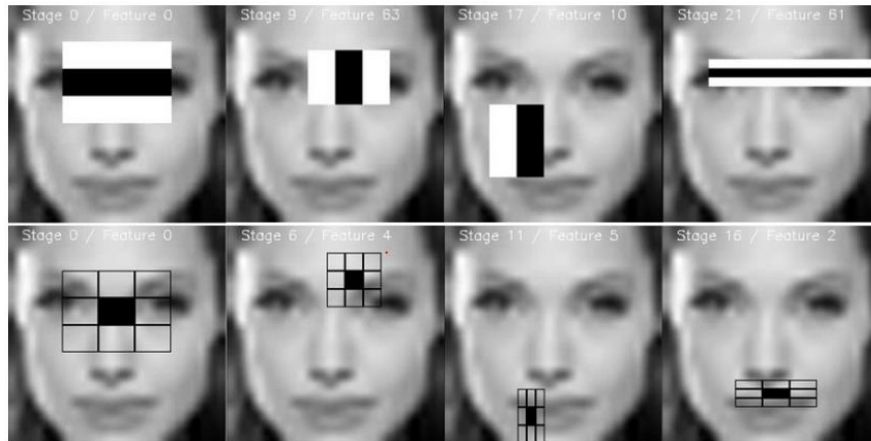
- LBPH Algoritmus: Egy elterjedt és hatékony módszer az arcfelismerés terén, amely a képfeldolgozási és mintázatelemzési technikákra épít.



*LBPH algoritmus reprezentálása*

Az LBPH algoritmus célja, hogy az arcfelvételeken az arc jellemzőit helyi szinten rögzítse, majd azokat összegzi egy hisztogram formájában, ami segíti az arcok felismerését és azonosítását. Az algoritmus egyszerűsége és hatékonysága miatt széles körben használják az arcfelismerési alkalmazásokban.

- Haar-cascade: egy gyors arcfelismerési módszer, amely a fényerősségkülönbségeken alapuló Haar jellemzők alkalmazásával dolgozik. Az algoritmus Adaboost tanulással javítja a felismerés hatékonyságát, és osztályozó kaskádokat alkalmaz a képen lévő arcok gyors azonosítására.



*Haar-cascade osztályozó*

Az előnyei közé tartozik a gyors működés és az egyszerű implementálhatóság, különösen az OpenCV használatával. Hátránya, hogy érzékeny a fényviszonyokra és a részleges arceltakarásra, ami csökkentheti a felismerés pontosságát.

### 5.3.2 Modern megközelítések

Az arcfelismerési technológiák az utóbbi években jelentős fejlődésen mentek keresztül, amelyet a mélytanulás terén elért áttörések és a nagy mennyiségű, címkézett adatbázisok elérhetősége segített elő. A legmodernebb modellek új határokat szabtak az arcfelismerés pontosságában, megbízhatóságában és hatékonyságában. Ilyen arcfelismerési technológiák a következők:

- FaceNet: Egy mélytanuláson alapuló arcfelismerési modell, amelyet a Google Research fejlesztett ki. A hálózat az embeddings megközelítést alkalmazza, amelyben az arcokat egy magas dimenziós térben reprezentálja, így lehetővé téve a hasonló arcok közötti távolság minimalizálását. A triplet loss függvény segítségével a modell hatékonyan tanulja meg az arcok közötti különbségeket, növelve ezzel az azonosítás és a hitelesítés pontosságát. A FaceNet-et széles körben alkalmazzák biometrikus azonosításban, képfeliratozásban és egyéb arcfelismerési feladatokban.
- DeepFace: Egy arcfelismerő modell, amelyet a Facebook fejlesztett ki, és 2014-ben mutattak be. Ez a modell mély konvolúciós neurális hálózatot (CNN) használ az arcok pontos azonosítására és osztályozására. A DeepFace egyik legfontosabb újítása az volt, hogy a képeket egy egységes 3D-s modell segítségével igazította, így csökkentve a különböző szögekből készült arcok közötti eltéréseket. Az algoritmus jelentősen javította az arcfelismerés pontosságát, megközelítve az emberi teljesítményt.

- VGGFace: egy arcfelismerő modell, amelyet az Oxfordi Egyetem Visual Geometry Group (VGG) kutatócsoportja fejlesztett ki. A modell egy mély konvolúciós neurális hálózatot (CNN) használ, amelyet nagyméretű arcképadatbázison tanítottak be. A VGGFace architektúrája hasonló a VGG hálózatokhoz, amelyeket általános képfeldolgozási feladatokra fejlesztettek ki, de kifejezetten az arcfelismeréshez optimalizálták. A rendszer képes nagy pontossággal azonosítani és összehasonlítani arcokat, miközben robusztus a különböző fényviszonyokkal és nézőpontokkal szemben.
- DeepId: egy arcfelismerési modell, amelyet a Chinese University of Hong Kong kutatói fejlesztettek ki. Ez volt az egyik első olyan mélytanulási alapú megoldás, amely a hagyományos képfeldolgozási módszerekhez képest jelentős előrelépést hozott az arcfelismerés pontosságában. A DeepID egy konvolúciós neurális hálózat (CNN) segítségével tanulja meg az arcképek reprezentációját, és képes a különböző személyek arcképét diszkrét, nagy dimenziós jellemzőtérbe leképezni. A modell egyik kulcsfontosságú újítása, hogy az arcokat osztályozási feladaton keresztül tanítják be, majd az így kapott jellemzők felhasználhatók azonosításra és összehasonlításra.

## Irodalomjegyzék

- [1] Types of Machine Learning - [https://blog.udemy.com/what-is-machine-learning-and-how-does-it-work/?utm\\_source=adwords&utm\\_medium=udemyads&utm\\_campaign=Search\\_DSA\\_GammaCatchall\\_NonP\\_la.EN\\_cc.ROW-English&campaigntype=Search&portfolio=ROW-English&language=EN&product=Course&test=&audience=DSA&topic=&priority=Gamma&utm\\_content=deal4584&utm\\_term=.ag\\_169801645584\\_.ad\\_700876640599\\_.kw\\_.de.c\\_.dm\\_.pl\\_.ti.ds.a-1456167871416\\_.li\\_9063015\\_.pd\\_.&matchtype=&gad\\_source=1&gclid=CjwKCAiAl4a6BhBqEiwAqvrquujWIaoKJfUuYVh\\_Ja4FMgV0ZcLL00PrkISsINM6IDHi5pstL1J9VxoCLBsQAvD\\_BwE](https://blog.udemy.com/what-is-machine-learning-and-how-does-it-work/?utm_source=adwords&utm_medium=udemyads&utm_campaign=Search_DSA_GammaCatchall_NonP_la.EN_cc.ROW-English&campaigntype=Search&portfolio=ROW-English&language=EN&product=Course&test=&audience=DSA&topic=&priority=Gamma&utm_content=deal4584&utm_term=.ag_169801645584_.ad_700876640599_.kw_.de.c_.dm_.pl_.ti.ds.a-1456167871416_.li_9063015_.pd_.&matchtype=&gad_source=1&gclid=CjwKCAiAl4a6BhBqEiwAqvrquujWIaoKJfUuYVh_Ja4FMgV0ZcLL00PrkISsINM6IDHi5pstL1J9VxoCLBsQAvD_BwE)
- [2] Mi az a gépi tanulás? - [https://www.coursera.org/articles/what-is-machine-learning?trk\\_ref=articles\\_unified\\_description\\_page\\_recs\\_card&utm\\_medium=sem&utm\\_source=gg&utm\\_campaign=B2C\\_EMEA\\_coursera\\_FTCOF\\_career-academy\\_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid=&device=c&keyword=&matchtype=&network=x&device\\_model=&adposition=&creativeid=&hide\\_mobile\\_promo&gad\\_source=1&gclid=CjwKCAiAl4a6BhBqEiwAqvrquulupX1yEnVIGAP0-3RkrstqMc2Eukl-s0cHGac4c9BUBFrBm-IVIZBoCR90QAvD\\_BwE](https://www.coursera.org/articles/what-is-machine-learning?trk_ref=articles_unified_description_page_recs_card&utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA_coursera_FTCOF_career-academy_pmax-multiple-audiences-country-multi-set2&campaignid=20882109092&adgroupid=&device=c&keyword=&matchtype=&network=x&device_model=&adposition=&creativeid=&hide_mobile_promo&gad_source=1&gclid=CjwKCAiAl4a6BhBqEiwAqvrquulupX1yEnVIGAP0-3RkrstqMc2Eukl-s0cHGac4c9BUBFrBm-IVIZBoCR90QAvD_BwE)
- [3] What is a machine learning algorithm? - <https://www.ibm.com/topics/machine-learning-algorithms>
- [4] Gépi tanulási algoritmusok - <https://www.geeksforgeeks.org/machine-learning-algorithms/>
- [5] Ajánlórendszer - <https://www.nvidia.com/en-eu/glossary/recommendation-system/>
- [6] Recommender Systems: Why And How? - <https://towardsdatascience.com/recommender-systems-a-complete-guide-to-machine-learning-models-96d3f94ea748>
- [7] Az ajánlórendszerek típusai - <https://onespire.hu/ajanlorendszerek-a-gyakorlatban/>
- [8] Hibrid rendszerek - <https://medium.com/codex/hybrid-recommender-system-netflix-prize-dataset-e9f6b4a875aa>
- [9] Felügyelet nélküli tanulás - <https://www.geeksforgeeks.org/ml-types-learning-part-2/>
- [10] What Is Deep Learning? - <https://builtin.com/machine-learning/deep-learning>
- [11] Arcdetektálás - <https://www.techtarget.com/searchenterpriseai/definition/face-detection>
- [12] Arcverifikáció - <https://www.sciencedirect.com/topics/computer-science/face-verification>
- [13] Arfelismerési módszerek - <https://medium.com/@khwabkalra1/face-detection-e18784e1c1f9>
- [14] Methods for Face Recognition - <https://medium.com/@khwabkalra1/face-recognition-e45aff329fba>
- [15] What is Mobilenet V2 model - <https://www.analyticsvidhya.com/blog/2023/12/what-is-mobilenetv2/>
- [16] MobileNet V2 modell - <https://www.geeksforgeeks.org/what-is-mobilenet-v2/>
- [17] Face ID - <https://support.apple.com/en-us/102381>
- [18] Apple Face ID - <https://www.stuff.tv/features/apple-face-id-explained/>
- [19] iOS Face ID - <https://www.pocket-lint.com/phones/news/apple/142207-what-is-apple-face-id-and-how-does-it-work/>
- [20] Android face autentucation - <https://source.android.com/docs/security/features/biometric/face-authentication>
- [21] Biometric - <https://source.android.com/docs/security/features/biometric>

```

Future<List<EventDTO>> getRecommendedEventsForUser(String userId) async {
    try {
        final userEventsSnapshot = await _firestore
            .collection('events')
            .where('participants', arrayContains: userId)
            .get();
        final List<EventDTO> userEvents = [];
        for (var doc in userEventsSnapshot.docs) {
            final data = doc.data();
            final event = EventDTO.fromJson(data, doc.id);
            userEvents.add(event);
        }
        final filteredUserEvents = userEvents
            .where((e) => e.embeddingVector != null && e.embeddingVector!.isNotEmpty)
            .toList();
        if (filteredUserEvents.isEmpty) return [];
        List<double> averageVector = List.filled(filteredUserEvents[0].embeddingVector!.length, 0.0);
        for (var event in filteredUserEvents) {
            for (int i = 0; i < event.embeddingVector!.length; i++) {
                averageVector[i] += event.embeddingVector![i];
            }
        }
        averageVector = averageVector.map((val) => val / filteredUserEvents.length).toList();
        final allEventsSnapshot = await _firestore.collection('events').get();
        final List<EventDTO> allEvents = [];
        for (var doc in allEventsSnapshot.docs) {
            final data = doc.data();
            final event = EventDTO.fromJson(data, doc.id);
            allEvents.add(event);
        }

        final List<Map<String, dynamic>> scoredEvents = [];
        for (var event in allEvents) {
            final bool isAlreadyAttended = userEvents.any((e) => e.id == event.id);
            final bool hasValidEmbedding = event.embeddingVector != null &&
                event.embeddingVector!.isNotEmpty &&
                event.embeddingVector!.length == averageVector.length;
            if (!isAlreadyAttended && hasValidEmbedding) {
                final ratingsSnapshot = await _firestore
                    .collection('ratings')
                    .where('eventId', isEqualTo: event.id)
                    .get();
                final ratings = ratingsSnapshot.docs
                    .map((doc) => (doc.data()['rating'] as num?).toDouble())
                    .whereType<double>()
                    .toList();
                if (ratings.isNotEmpty) {
                    final avgRating = ratings.reduce((a, b) => a + b) / ratings.length;

                    if (avgRating > 3.0) {
                        final similarity = cosineSimilarity.cosineSimilarity(
                            averageVector,
                            event.embeddingVector!,
                        );
                        scoredEvents.add({
                            'event': event,
                            'similarity': similarity,
                        });
                    }
                }
            }
        }
    }
}

```

```

model_name = "NYTK/PULI-GPT-2"
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)

base_events = [
    "Jazz Fesztivál", "Kolbász Fesztivál", "Virágkarnevál", "Színházi előadás", "Bábszínház",
    "Sportverseny", "Kézműves workshop", "Karácsonyi vásár", "Bor fesztivál", "Múzeumi Tárlatvezetés",
    "Operaelőadás", "Stand-up comedy show", "Gyereknapi", "Hagyományőrző fesztivál", "Gasztronómiai fesztivál",
    "Bor kóstoló", "Rock koncert", "Filmfesztivál", "Könyvvásár", "Táncház", "Szilveszteri buli",
    "Képzőművészeti kiállítás", "Virtuális valóság élmény", "Történelmi emlékhely látogatás",
    "Tudományos előadás", "Karácsonyi koncert", "Tavaszi piac", "Kézműves vásár", "Egyetemi nyílt nap",
    "Zenei workshop", "Képzőművészeti workshop", "Légibemutató", "Vízilabda mérkőzés", "Futóverseny",
    "Biciklis túra", "Fesztivál utca", "Fiatal tehetségek koncertje", "Bográcsozás", "Borkóstolás",
    "Gasztronómiai túra", "Hegyi túra", "Légivásár", "Bábszínházi előadás", "Főzőverseny",
    "Közösségi táncst", "Divatbemutató", "Borkóstoló rendezvény", "Reggeli jóga", "Falkaszínház",
    "Zenei fesztivál", "Bajnokság döntő", "Felhőtúra", "Népzenei koncert", "Kertészeti kiállítás",
    "Kertészeti workshop", "Állatvédelmi rendezvény", "Kézműves termék bemutató", "Sportnap",
    "Bálnales", "Indián napok", "Őszköszöntő buli", "Hegyi futás", "Kerékpáros verseny",
    "Úszóverseny", "Képzőművészeti alkotótábor", "Filmklub", "Vadászati kiállítás", "Régészeti túra",
    "Tudományos fesztivál", "Társadalmi kezdeményezés bemutató", "Szent Iván-éji buli",
    "Hegyi túrázó verseny", "Befőzés workshop", "Fotókiállítás", "Fejlesztő workshop",
    "Interaktív színházi előadás", "Autókiállítás", "Képzőművészeti rendezvény",
    "Mesefilm maraton", "Alkotói közösség est", "Fenntartható divat nap", "Tavaszcímű kézműves foglalkozás"
]

prompt_templates = [
    "Írj egy izgalmas leírást erről az eseményről: {}.",
    "Mutasd be részletesen, mi várható a rendezvényen: {}.",
    "Adj vonzó ismertetőt a következő eseményhez: {}.",
    "Meséld el élményszerűen, milyen a(z) {} esemény!"
]

generated_texts = []

for index, event in enumerate(base_events, start=1):
    prompt = random.choice(prompt_templates).format(event)

    inputs = tokenizer(prompt, return_tensors="pt")

    output = model.generate(
        **inputs,
        max_length=250,
        num_return_sequences=1,
        temperature=1.2,
        top_k=50,
        top_p=0.9,
        no_repeat_ngram_size=2
    )

    generated_text = tokenizer.decode(output[0], skip_special_tokens=True)

    cleaned_text = generated_text.replace(prompt, "").strip()

    generated_text = "\n".join([line.strip() for line in cleaned_text.strip().split("\n")])

    generated_texts.append(generated_text)

df = pd.DataFrame({
    "event_name": base_events,
    "description": generated_texts
})

```

## Nyilatkozat

Alulírott, Kun Réka Bianka programtervező informatikus MSc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszékén készítettem, programtervező informatikus MSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem Diplomamunka Repozitóriumában tárolja.

Dátum 2025. május 12.

A handwritten signature in blue ink, reading "Kun Réka Bianka".

Aláírás



### **Köszönetnyilvánítás**

Szeretnék ezúton köszönetet nyilvánítani mindazoknak, akik támogattak abban, hogy ez a szakdolgozat elkészülhessen. Köszönettel tartozom a barátaimnak és családomnak, akik állandóan bíztattak és támogattak, amikor nehéz pillanatokat éltem át a szakdolgozat írása során.

Külön szeretnék köszönetet mondani, Kiss-Vetráb Mercedesnek, aki időt és energiát szánt rám, valamint hasznos tanácsokkal és javaslatokkal látott el. A szakdolgozat készítésének folyamata alatt sokat tanulhattam tőle és nagyra értékelem a szaktudásáért, és tapasztalatáért.

Végezetül, köszönöm mindenkinek, aki hozzájárult ahhoz, hogy ezt a szakdolgozatot elkészíthettem.