Tartalomjegyzék

[1. Bevezetés 3](#_Toc184167918)

[1.1 A kutatás célja 4](#_Toc184167919)

[1.2 A kutatás kérdései 4](#_Toc184167920)

[2. Hazai közösségi autómegosztás 5](#_Toc184167921)

[2.1 Helyzeti előnyök és kihívások 5](#_Toc184167922)

[2.2 Gazdasági szerepe 6](#_Toc184167923)

[2.3 Környezetvédelmi hatása 7](#_Toc184167924)

[2.4 Szolgáltatók bemutatása 7](#_Toc184167925)

[2.4.1 GreenGo 8](#_Toc184167926)

[2.4.2 MOL Limo 8](#_Toc184167927)

[2.4.3 wigo 9](#_Toc184167928)

[3. A kutatás módszertana és módszerei 10](#_Toc184167929)

[3.1 Adatgyűjtés és feldolgozás 10](#_Toc184167930)

[3.2 Adatbővítési technikák 11](#_Toc184167931)

[3.3 Elemzési modellek 11](#_Toc184167932)

[3.4 Értékelési módszerek 12](#_Toc184167933)

[4. Szakirodalom 13](#_Toc184167934)

[4.1 Mély tanulás 13](#_Toc184167935)

[4.1.1 PyTorch 14](#_Toc184167936)

[4.2 Neurális hálózatok 15](#_Toc184167937)

[4.2.1 LSTM 16](#_Toc184167938)

[4.3 Természetes nyelvfeldolgozás 16](#_Toc184167939)

[4.3.1 Téma modellezés 17](#_Toc184167940)

[4.4 Nagy nyelvi modellek 18](#_Toc184167941)

[4.4.1 BERT 19](#_Toc184167942)

[4.5 Szentiment analízis 20](#_Toc184167943)

[4.5.1 Adatbővítés 21](#_Toc184167944)

[4.5.2 Teljesítménymutatók 21](#_Toc184167945)

[4.6 RAG modell 23](#_Toc184167946)

[4.6.1 LangChain 24](#_Toc184167947)

[5. Esettanulmány 24](#_Toc184167948)

[5.1 Adatok ismertetése 24](#_Toc184167949)

[5.2 Téma modellezés 26](#_Toc184167950)

[5.1.1 Kibővített adathalmazok 30](#_Toc184167951)

[5.2 Klasszifikáló modellek 31](#_Toc184167952)

[5.2.1 DistilBERT 31](#_Toc184167953)

[5.2.2 XLM-RoBERTa 33](#_Toc184167954)

[5.2.3 LSTM 34](#_Toc184167955)

[5.2.4 huBERT 36](#_Toc184167956)

[5.3 RAG modell 41](#_Toc184167957)

[5.3.1 Klasszifikáló modell 41](#_Toc184167958)

[5.3.2 Hibrid modell 42](#_Toc184167959)

[6. Összefoglalás 43](#_Toc184167960)

[6.1 Értékelés és következtetések 43](#_Toc184167961)

[6.2 Javaslatok 44](#_Toc184167962)

[7. Irodalomjegyzék 46](#_Toc184167963)

# 1. Bevezetés

Manapság egyre növekvő népszerűségnek örvend a megosztáson alapuló gazdaság elve, nem csak világszerte, de hazánkban is. Különböző vállalatok dolgoznak ki újabb és újabb megoldásokat, melyekkel csökkenthetik adott szolgáltatások és erőforrások kihasználatlanságát, ezzel széles körben kedvezve a társadalomnak. Nem kivétel ezalól a közlekedés sem, hiszen, amennyiben nem szeretnénk a klasszikus tömegközlekedés formájához ragaszkodni, számtalan más módon van még lehetőségünk a fővárosban utazni. Legyen szó kerékpárról, rollerről vagy akár autóról, hiszen ezen járművek bármelyikét könnyen bérbe tudjuk venni a nap huszonnégy órájában az utcákon. Ezek közül kiemelném a carsharing, vagyis a közösségi autómegosztás szolgáltatását, mely egyre nagyobb felhasználói réteggel bír, annak ellenére, hogy jelenleg csak Budapest területén és közvetlen vonzáskörzetében érhető el.

Ez nem meglepő, hiszen egy kényelmes, ám mégis környezetbarát alternatívát nyújtanak a magántulajdonú személygépjárművek egyre növekvő fenntartási és üzemeltetési költségeivel, a város növekvő forgalmával, továbbá a parkolási nehézségekkel szemben.

Saját ismeretségi körömben is, többek között a fentebb említett előnyök miatt egyre többen választják tudatosan fő közlekedési eszközükként a három, legnagyobb körben elérhető közösségi autómegosztó szolgáltatók flottájának használatát, azaz a GreenGo, a MOL Limo vagy a wigo autóit. Személy szerint, én is szívesebben bérlek autót a belvárosi közlekedéshez, ahelyett, hogy a saját járművemmel vennék részt a forgalmi dugókban, vagy szükséges lenne alaposan megfontolnom az adott úticélom parkolási lehetőségeit.

Tovább növeli az említett márkák sikereit, hogy nem csak magánszemélyeknek, hanem a vállalati ügyfelek részére is kínálnak személyre szabott csomagajánlatokat. A költségmegtakarításon felül nem csak az üzleti utak lebonyolítását egyszerűsítheti, hanem akár a munkavállalók közlekedési lehetőségeit is bővítheti.

A közösségi autómegosztás, mint szolgáltatás önmagában nem képes felszámolni a magántulajdonú autók tartását, ugyanis az egybefüggő bérlés időtartama legfeljebb harminc napig terjedhet, bizonyos feltételeknek eleget téve, így egy adott felhasználói réteg csupán kiegészítő megoldásként tekint rá. Azok számára, akik bármilyen okból nem engedhetik meg maguknak a saját jármű fenntartását, kiváló megoldást jelenthet, ugyanakkor sokak számára lehetőséget ad arra is, hogy olyan járműveket próbáljanak ki, amelyeket alapvető körülmények között nem lenne lehetőségük. Ebből kifolyólag a felhasználói elvárások és igények jelentős mértékben eltérnek egymástól, különösen a különböző csomagok, árazás, flották és zónák tekintetében.

## 1.1 A kutatás célja

Szakdolgozatom célja ezen három közösségi autómegosztó cég szolgáltatásainak valós, felhasználói vélemények és értékelések alapján történő összehasonlítása. A célom eléréséhez olyan kutatási módszerek, technikák és mesterséges intelligencia, mélytanulási algoritmusok szolgálnak, melyek lehetővé teszik a nagy mennyiségű felhasználói visszajelzésekből levont következtetések alapos és átfogó elemzését.

A kutatás során a szolgáltatások közötti különbségek, erősségek és fejlesztendő területek azonosításához szentiment elemzést végeztem. Ehhez nagy nyelvi modellek mélytanuló, szöveg elemző algoritmusait alkalmaztam. Ezek közül kiemelném a magyar nyelvre specializált huBERT, valamint egyéb, többnyelvű, BERT-alapú modellek tanítását, finom-hangolását. Továbbá, egy úgy nevezett retrieval-augmented generation (RAG) modellt, avagy keresésalapú szövegkiegészítő algoritmust integráltam, mely lehetővé tette a releváns információk hatékonyabb bővítését, feldolgozását és a tényleges eredmények kinyerését. Az elemzési modellek kialakítása során törekedtem arra, hogy az algoritmusok minél hatékonyabb és precízebb módon legyenek képesek vizsgálni a felhasználói értékelések változatos mintáit. Mindezek megvalósításához elengedhetetlen feltételnek bizonyult a megfelelő minőségű és mennyiségű adat strukturált előállítása, automatizált módon.

## 1.2 A kutatás kérdései

A kutatásom során az alábbi kérdésekre keresem a választ:

1. Vannak-e gyakran előforduló panaszok vagy kiemelkedő elégedettségi tényezők az egyes szolgáltatóknál?
2. Mennyire változatosak az egyes autómegosztókra adott értékelések?
3. Mi jellemzi általánosan az egyes alkalmazások értékeléseit?
4. Alkalmazható-e olyan nyelvi modell a felhasználók értékelésein, amely legalább 80% pontossággal képes klasszifikálni új szöveges adatokat?
5. Feltétlenül hatékonyabban működik-e egy specifikusan magyar nyelvre feltanított és optimalizált modell, egy többnyelvű modellel szemben?

Hipotézisem alapjául szolgál az a megállapítás, hogy létezik egy olyan autómegosztó, amely összességében kiemelkedik a másik kettővel szemben.

# 2. Hazai közösségi autómegosztás

## 2.1 Helyzeti előnyök és kihívások

A magánemberek autózási szokásaiban határozott átalakulás látszódik, mely szorosan összefügg a környezettudatossággal és a saját autó fenntartásával járó magas, kötelező kiadásokkal, költségekkel. Ezzel párhuzamosan a vállalati flották menedzselése is olyan átalakulást mutat, ami jelentősen kedvező hatással bír a közösségi autózás fellendüléséhez. A cégek beszámolása alapján 2023-ban, az előző évi adatokhoz képest, a magán- és üzleti célú felhasználók száma közel 25-30 százalékkal, illetve 65 százalékkal gyarapodott. A szolgáltatók, tehát a GreenGo 2016-os, a MOL Limo 2018-os és a wigo (korábban DriveNow) 2019-es indulása óta több, mint 70 millió kilométert vezettek összesen a felhasználók a bérléseik során, 2024. február 15. dátummal bezárólag. (jovomobilatasa.hu-1, 2024.)

A carsharing terjedését elősegíti az is, hogy a szolgáltatók különböző kedvezményes konstrukciókkal kedveznek a fiatalabb közösség számára. A GreenGo 25 év alattiaknak biztosít egy egyedülálló díjcsomagot, míg a wigonál diáki jogviszony ellenében van lehetőség alacsonyabb utazási díjjal élni. Mint azt korábban említettem, a vállalatoknak mindhárom szolgáltatónál van lehetőségük egyedi csomagokat kötniük, amelyek rendkívül költséghatékony megoldásokat kínálnak. Lehetőségük van támogatniuk dolgozóik mobilitását, akár adó- és járulékmentesen bővíteni a Cafeteria elemeit a szolgáltatással, miközben hozzáférhetnek a flottakezelési előnyökhöz, kiemelt ügyfélszolgálati elérhetőséggel és jelentős adminisztráció nélkül.

A fővárosi parkolás egyre súlyosbodó problémáit enyhítheti a közösségi autók használata, hiszen csökkenti a parkolóhelyek iránti keresletet. A fenntarthatóság jegyében elektromos és hibrid autók gazdagítják a flottákat, amelyekkel hosszabb bérlés esetén díjmentes a parkolás, az egyébként díjköteles zónákban, illetve többnyire kisebb méretűek is, így ezen szempontból is könnyebb a parkolás velük. A hagyományos, belső-égésű járművek nagy számban érhetőek el automataváltós kivitelben, így a kevésbé tapasztalt sofőrök számára is vonzó megoldást jelent, főleg a csúcsidőben történő közlekedés esetén. Valamint, hosszabb utazásra is kényelmes alternatíva lehet, nem csak belföldön az akár harminc napig tartó bérléssel, hanem külföldi út esetén is, ugyanis a Liszt Ferenc repülőtéren a szolgáltatók autóival érkezőknek dedikált parkolórészt tartanak fent.

Azonban kihívást jelent, hogy az emberek sokszor elmulasztják az általános szerződési feltételek betartását, amely a járművek szabálytalan használatát és tisztán tartásának elmaradását eredményezi. Márkaspecifikus fórumokon, de akár az applikációk értékelései között gyakran számolnak be az ügyfelek olyan kellemetlenségekről, melyek kedvezőtlenül hatnak az adott szolgáltató megítélésére. Szintén hátrányosan hatnak a cégekre az olyan esetek, ahol nem csak a felhasználó szembesül visszaéléssel, hanem a forgalom más résztvevője is, a KRESZ be nem tartása vagy felelőtlen vezetési magatartás esetén. Mindezek mellett, a bérlési zónák kibővítése a várt pozitív mellett negatív vonzattal is jár, ugyanis a határok növelésével párhuzamosan nem növekszik kielégítő arányban a flották állománya, ezáltal veszít a szolgáltatás a kényelméből azzal, hogy a délutáni csúcsidőt követően, adott területeken nincsen szabadon foglalható jármű.

## 2.2 Gazdasági szerepe

Az autómegosztás szolgáltatása több szempontból is meghatározó szerepet játszik a gazdaságban. Sikerének köszönhetően egyre több területen teret nyer a megosztott használaton alapuló gazdaság a hagyományos tulajdonlás helyett. A kedvezőtlen gazdasági környezet, energiaválság és infláció okán átalakulóban vannak a közlekedési szokások, az emberek törekednek az olcsóbb és fenntarthatóbb megoldásokra. Egyes felmérések alapján a saját tulajdonú járművek egy év 96 százalékban átlagosan csak parkolnak, ezzel hozzájárulva az egyre nehezebb parkolási helyzethez. Továbbá, az autó méretének függvényében évi 7.500 – 13.000 kilométer megtétele esetén már nem éri meg a fenntartása sem. Szakértők szerint a megosztott használatú járművek képesek legalább hat, de akár tizenkettő saját tulajdonban lévő autót kiváltani. Ez mind az említett parkolási nehézségeken enyhít, a forgalom mértékén, valamint a károsanyag kibocsátás szintjén képes látványos csökkenést elérni. (jovomobilitasa.hu-2, 2023.)

Jelentős költségmegtakarításra lehet vele számítania azoknak, akik nem használják autóikat nap, mint nap, hiszen a bérlés díja használat alapú és alapvetően magába foglalja a szervizelés, üzemanyag, autópályamatrica, biztosítás és gépjárműadó költségeit. A GreenGo kalkulációi alapján egy tizennégy, illetve négyéves autó költségei, alkalmi felhasználás esetén havonta nagyjából 77.000 és 135.000 Forint közé esik, míg a közösségi autó díja 50.000 Forint körül mozog. Napi rendszerességű felhasználás során a két említett saját autó költségei 95.000 és 150.000 Forintra növekednek, ezzel párhuzamosan a bérlés legalább 92.000 Forintot jelentene. (hvg.hu, 2023.)

Az iparág továbbá a lokális munkaerőpiac számára új álláslehetőségeket teremt, hiszen szükség van a flották karbantartására, a járművek tisztítására és szervizelésére, ügyfélszolgálat fenntartására és egyéb logisztikai műveletek ellátására, nem beszélve az applikációk és termékek fejlesztéséről, ami további digitális innovációt vonz maga után. A több napra szóló bérlési konstrukciók pedig pozitívan hatnak a belföldi turizmusra. Hosszú hétvégéken, valamint nyáron kedvező alternatívát nyújt a kirándulni vágyóknak. (greenguide.hu, 2021.)

## 2.3 Környezetvédelmi hatása

2023. évi adatok szerint közel 1.500 autó áll rendelkezésre a bérlőknek, melyek 40 százaléka tisztán elektromos hajtású. Ennek nagy hányadát a GreenGo flottája teszi ki, ugyanis az indulásuk óta kizárólag elektromos autók alkotják. Ezáltal 2022-ben több, mint 1.300 tonna extra szén-dioxid terheléstől mentesítették a légkört. Azonban, Európában a közösségi autózással évente akár tizenkét tonnával csökkenhet az átlagos a szén-dioxid emisszió mértéke, a meg nem vásárolt járművek, a kevesebb vezetett kilométerek és az alacsonyabb kibocsátásból adódóan. (jovomobilitasa.hu-2, 2023.)

A környezettudatos működés mellett a szolgáltatók társadalmi felelősségvállalási szerepük is jelentős. A GreenGo az autói felületét gyakran használja hasznos ügyek népszerűsítésére, például védett madarak védelmére, a karbon lábnyom csökkentésére, hazai természetes vizek tisztaságának fenntartására vagy fák ültetésére. Utóbbi kampányuk közel 7.000 fa elültetését finanszírozta a főváros különböző pontjain. (greengo.com-1, 2024.)

A márkák igyekeznek a fiatalabbakat motiválni a fenntartható jövő érdekében, így különböző kedvezményeket nyújtanak a diákoknak és vagy 25 éven alatti korosztálynak. A reggeli és délutáni csúcsidőben további kedvezménnyel ösztönzi a felhasználókat a wigo, ezzel tovább csökkentve az esetleges károsanyag kibocsátás mértékét. (wigomobility.com, 2024.)

## 2.4 Szolgáltatók bemutatása

Az elemzés könnyebb értelmezéséhez szükségesnek tartom röviden bemutatni a három vizsgált carsharing szolgáltatót. Mindhárom márka szolgáltatásai applikáción keresztül érhetőek el, mind Android és iOS rendszereken. Ezen bemutató és összehasonlítás alapjául szolgálnak olyan általános tulajdonságok, mint az egyes flották méretei, elérhető hajtástípusok, díjcsomagok, magánfelhasználók regisztrációs feltételei, valamint a vállalatok számára kínált megoldások. Továbbá, kitérek olyan egyedi tulajdonságokra is, amelyek kiemelkedő szerepet játszhatnak a cégek versenyképességében és a felhasználók megítélésében, mint a különböző fenntarthatósági törekvések, kedvezményprogramok és ajándékozási lehetőségek.

### 2.4.1 GreenGo

A GreenGo 2016-ban indult, elsőként behozva a közösségi autómegosztás iparát Magyarországra. Mai napig az egyetlen olyan szolgáltató, amely flottája 100 százalékban tisztán elektromos hajtású. Mérete már elérte az 500 autót. Céltudatosan törekednek a zöld és fenntartható megoldásokra, a saját töltőhálózatuk megújuló energiaforrásokat használ, valamint a járművek takarítása is környezetkímélő módon történik,

Négy különböző díjcsomag közül van lehetősége választania a felhasználóknak, melyek havidíja a díjmentes és 4.990 Forint között helyezkedik el. Az alap szolgáltatáson felül létezik a ,,Comfort” is, mely a háromórástól egészen a húsz napos bérlési intervallumra kínál lehetőséget. A foglalható járművek típusainak listája ennél a szolgáltatónál a legrövidebb a három közül. Magányszemélyeken kívül, kis- és nagyvállalatoknak is biztosítanak üzleti megoldásokat, akár cégautó használatra, akár béren kívüli juttatásként vagy dolgozói kedvezményprogramra.

A bérlés applikáción keresztül lehetséges, mely elérhető mind Android, mind iOS készülékekre. A honlapon elérhető zónatérképen aktualizáltan láthatóak a foglalható autók. Amennyiben a felhasználó betöltötte a tizennyolcadik életévét, valamint rendelkezik érvényes B kategóriás jogosítvánnyal, pár perc alatt regisztrálhat és igénybe veheti a szolgáltatást. (greengo.com-2, 2024.)

### 2.4.2 MOL Limo

A Magyar- Olaj és Gázipari Nyrt. 2018-ban indította el MOL Limo néven a saját közösségi autómegosztó szolgáltatását, flottája mára 450 autóval rendelkezik, melynek egyharmada elektromos hajtású. Célkitűzésük, hogy a jövőben csak elektromos autók alkossák a flottát, illetve minél szélesebb körben elérhető legyen a közösségi autózás szolgáltatása. (molgroup.info, 2024.)

Három csomagtípus érhető el, a nulla Forintos havidíjtól egészen a 7.990 Forintig, mely értéke teljes mértékben leutazható. Továbbá, az előfizetők közösségi kerékpár, azaz MOL Bubi havibérletre is jogosultak egy fenntarthatósági promóció keretein belül. Az alap szolgáltatással legfeljebb három nap egybefüggő bérlésre van lehetőség, ám a „LIMO FIX” szolgáltatással ez egészen harminc napig kiegészíthető. A foglalható típusok száma összesen tizenhárom, biztosítva megfelelő járművet minden alkalomhoz, valamint a teljesen elektromos és hibrid hajtású autókon túl benzinesek is helyet kaptak a flottában, nem csak automata, hanem manuális váltóval is.

Vállalati ügyfelek, pontosabban egyéni vállalkozók, kisvállalkozások, közép- és nagyvállalatok számára egyedi szerződéskötés elérhető, céges flotta vagy lízingelt autók alternatívájaként, akár kedvezményprogrammal és juttatási rendszer kiegészítőjeként.

A bérlés ugyancsak applikáción keresztül érhető el, Android és iOS rendszereken, azonban a weboldalukon is meg lehet tekinteni az elérhető autók térképét. A regisztráció feltételei azonban szigorúbbak a GreenGo-nál, ugyanis a felhasználónak be kell töltenie a huszadik életévét és legalább egy éve szükséges rendelkeznie érvényes, B kategóriás jogosítvánnyal. Nem csak felhasználók részére érhető el az ajándékutalvány rendelése, 5.000 – 25.000 Forint értékek között, személyre szabható ajándékkártya témával, mely aktiválástól számított egy évig használható fel. (mollimo.hu, 2024.)

### 2.4.3 wigo

2019. tavaszán DriveNow néven elindult a harmadik carsharing szolgáltató is Budapest területén, a német Daimler AG és BMW Group együttesen üzemeltetett nemzetközi SHARE NOW autómegosztó szolgáltatás részeként. (press.bmwgroup.hu, 2019.) Azonban, 2020-ban az egységesítés érdekében ShareNow név alatt üzemelt tovább a cég. Ezzel elérhetővé tették a külföldi regisztrációval rendelkező felhasználók részére a budapesti autók elérését, valamint a magyar felhasználók számára a tizenöt európai nagyváros járműinek használatát. (Simon, 2020.) Az első átnevezést követően 2023. októberében újabb változást vezettek be, a ShareNow helyét átvette a wigo, ami ezúttal már az AutoWallis csoport saját mobilitási márkáját képviseli. (vg.hu, 2023.) A változás a közel 100.000 fős felhasználótábort csupán annyiban érintette, hogy a régi alkalmazás helyett egy újat kellett telepíteniük, nem volt szükség újbóli regisztrációra. (hvg.hu-2, 2023.)

Két fő szolgáltatás típust kínál, a ,,basic” díjmenteset, illetve az ,,extra” 1.690 Forint havidíjút, amelyből 850 Forint levezethető. Mindkét szolgáltatásban három díjcsoportra oszlanak besorolás alapján a bérelhető autók. A „city” megnevezésű típusok a legkedvezőbbek, ide kisebb méretű autók tartoznak. Ezt követi az „active” kategória, amely két, közepes méretű típust, majd a ,,special” a különlegesebb, prémium márkásnak számító járműveket kínál. Az 500 autóból álló flottában a MOL Limo-hoz hasonlóan szintén megtalálhatóak tisztán elektromos hajtású, valamint benzines automata- és manuális váltóval felszerelt gépjárművek. A bérlés legfeljebb három napig tarthat, ezen időtartam túllépése esetén újraindul a bérlés. Ajándékutalványt a wigo is biztosít megvételre, a 60.000 Forintos utalvány 66.000 vezetési kreditnek felel meg, ami aktivációtól számított kilencven napig használható fel. A két konkurens céghez hasonlóan a wigo is kínál üzleti megoldásokat kis- és nagyvállalkozásoknak, melybe a munkavállalók szintén bevonhatóak kedvezményes utazás biztosításaként.

A regisztráció feltétele a tizennyolcadik életév betöltése, valamint legalább egy éve érvényes B kategóriás jogosítvány megléte. Az applikáció mindkét okostelefon rendszeren elérhető, valamint a weboldalon elérhető zónatérképen valós időben megtekinthetőek a bérelhető járművek. (wigomobility.com-2, 2024.)

# 3. A kutatás módszertana és módszerei

Az elemzésem során különböző modern, mesterséges intelligenciával támogatott módszerek sorát alkalmaztam. Ezen fejezetben ki szeretnék térni minden kutatási lépésben használt módszertan jellemzésére, az adatgyűjtéstől egészen az optimalizált modellek végső felhasználásáig. Valamint, ezen eljárások teljesítményének méréséhez alkalmazott metrikákat említem meg.

## 3.1 Adatgyűjtés és feldolgozás

A kutatás eredményének konzisztenciájának fenntartása és pontosságának biztosítása érdekében kulcsfontosságú volt egy strukturált adatgyűjtési és feldolgozási módszertan kialakítása. Ennek eleget téve, a megfelelő mennyiségű adat előállítása automatizált web scraping eljárással valósult meg. Automatizáltsága pedig a strukturált, értelmezhető adattábla formájának megteremtésére is kiterjedt. Az elemzés következő szakasza előtt szükségesnek bizonyult az adatok további tisztítása, mint a duplikált elemek törlése, valamint az egyszerűbb kezelhetőség érdekében, a kutatás szempontjából irreleváns oszlopok eltávolítása, melyek inkonzisztens metaadatokkal szolgáltak.

A feldolgozás során olyan további számítógépes eszközöket és módszereket alkalmaztam, amelyek a nyers szöveget strukturált formává alakították, alkalmassá téve azt további vizsgálatra. Annak érdekében, hogy a klasszifikáló modellek megfelelően képesek legyenek kezelni a szöveges adatokat, tokenizálást alkalmaztam, ezáltal elősegítve a modellek hatékonyságának növelését a tanulási szakaszokon.

A minták azonosításához és előforduló témák elemzéséhez a tokenizálás mellett, a szöveges adatokat megtisztítottam a fölösleges karakterektől, mint a különböző írásjelek, számok és emotikonok, valamint az önmagukban jelentést nem hordozó, azaz stop szavaktól. A numerikus ábrázolás érdekében az így kapott állományt TF-IDF (Term Frequency-Inverse Document Frequency) vektorizációval transzformáltam.

A felsorolt technikák és módszerek önálló, valamint együttes alkalmazásával tehát, a kutatás során használt szövegelemző modellek képesek voltak hatékonyan feldolgozni az adatokat a különböző folyamataik során.

## 3.2 Adatbővítési technikák

A kezdeti adatgyűjtéssel elért mennyiség nem bizonyult megfelelő alapnak a modellek feltanítására, így különböző adatbővítési technikákkal igyekeztem az állomány osztályai között keletkezett kiegyensúlyozatlanságokat javítani. Az egyik ilyen módszer a SMOTE (Synthetic Minority Over-sampling Technique), mely szintetikus minták létrehozásával növeli az alulreprezentált osztályok méretét, a legnagyobb osztály méretével kiegyenlítve, kisebb variancia értékekkel.

Az osztályok közti meghatározó, méretbeli különbségek kiegyenlítésére szinonimák generálásával is próbálkoztam, GPT-4 modell használatával, amely képes volt magyar nyelvű, értelmes, de mégis eltérő mintákat létrehozni, amiket egyszerűen hozzáfűzhettem az adatállományomhoz. Ez a módszer a tanítómondatok jelentésének megváltoztatása nélkül biztosít nagyobb változatosságot a tanító algoritmus számára. Hasonló eljárásként a back-translation technikáját is alkalmaztam, amely során a teljes adatállományt először angol nyelvre fordította egy erre fejlesztett modell, majd pedig vissza, magyar nyelvre. Ez ugyancsak az eredeti szöveges adatok jelentésének torzítása nélkül volt képes elégséges mennyiséggel növelni az adataimat. Végül az így keletkezett, augmentált adatokkal bővített állományokat külön-külön teszteltem a szövegelemző modelleken, melyek egy része javuló teljesítményt eredményezett.

## 3.3 Elemzési modellek

A kutatásom során, a szövegelemzésre az egyik legmodernebb nyelvi modell variációi közül több megoldást is alkalmaztam, azaz a BERT (Bidirectional Encoder Representations from Transformers) transzformer alapú modell DistilBERT, huBERT, RoBERTa változatait. Valamint, egy LSTM (Long Short-Term Memory) neurális hálózatot szintén kipróbáltam. Az említett modellek különböző mélytanulási megközelítéseket ötvöznek a szentiment-analízis hatékony végrehajtása érdekében, a PyTorch keretrendszerének támogatásával. A felsorolt BERT-alapú modellek a nyelvi feldolgozás terén kiváló teljesítményt nyújtanak a szövegkörnyezetek elemzésével, míg az LSTM a sorozatadatok kezelésében és a szövegek időbeli összefüggéseinek felismerésében hatékony. A huBERT modell, amely kifejezetten magyar nyelvre van optimalizálva, különösen fontos szerepet töltött be, hiszen a célom a lehető legpontosabb szentiment-elemzés elvégzése volt, javarészt magyar nyelvű felhasználói visszajelzéseken.

A hagyományos mélytanulási modellek alkalmazása mellett, a kutatás során egyéb megközelítésű metódusokat is implementáltam, például a polaritás értékelés módszerét, amely az egyes szövegek érzelmi irányultságát (pozitív, semleges vagy negatív) határozza meg. Továbbá az adathalmazon egy témamodell-alapú elemzést is végeztem, az LDA (Latent Dirichlet Allocation) modellezési technikát alkalmazva. Ez a módszer segített azonosítani a különböző szöveges adatokban rejlő témákat, amelyek alapvető struktúrákat tükrözhetnek.

Elengedhetetlen lépésnek bizonyult a mélytanuló algoritmusok tanulási fázisainak olyan finomhangolása, hogy a klasszifikáló modellek minél jobban alkalmazkodjanak az egyes osztályok sajátosságaihoz, így elérve a lehető legoptimálisabb teljesítményt és pontosságot. Ennek érdekében a tanulási hiperparamétereket oly módon határoztam meg, hogy azok a legjobb eredményeket produkálják a felhasználói visszajelzések szentimentjének meghatározásában.

A kutatás végén a legoptimálisabb eredményeket elérő klasszifikáló modellt továbbfejlesztettem egy Langchain alapú rendszerrel, amely képes integrálni a szövegelemzési folyamatot egy generatív modellel. Ezáltal egy olyan Retrival-Augmented Generation (RAG) modellt hoztam létre, amely képes nemcsak a szentiment-elemzés elvégzésére, hanem a felhasználói értékelések, valamint a gyakran ismételt kérdések dokumentumaiból történő releváns információk kinyerésére is. A rendszer a kinyert információkat összegzi, és a prediktált szentiment értékkel együtt egy generált válasz formájában adja vissza az eredményt.

Ez az integrált megoldás nem csupán a szövegek érzelmi irányultságának meghatározására korlátozódik, hanem képes a felhasználói visszajelzések és egyéb, a közösségi autómegosztó szolgáltatókról elérhető dokumentumok egyesítésével, gyors és pontos választ adni a felmerülő kérdésekre. A kutatás során tehát a legmodernebb nyelvi modellek és algoritmusok alkalmazásával egy olyan komplex és hatékony rendszert sikerült létrehoznom, amely képes megbízhatóan elemezni és feldolgozni a felhasználói értékeléseket, ezzel hozzájárulva a szolgáltatások és a felhasználói élmény jobb megértéséhez.

## 3.4 Értékelési módszerek

A klasszifikáló modellek teljesítményének mérésére több metrika is szolgált, melyek együttese egyértelműen reprezentálta a modellek jóságát és az eredmények pontosságát. A pontosság (accuracy) alapvető mutatóként mindegyik modell kiértekélésénél szerepelt, amely a helyesen klasszifikált adatok arányát mérte az összes adaton elosztva. Emellett a precizitás (precision), visszahívás (recall) és az ezekből származtatott F1-érték mutatóit mind alkalmaztam az elemzések során. Utóbbi metrikák a modellek teljesítményét részletesebben jellemzik, különösen egyes osztályokra vetítve. A vizualizációs kiértékeléshez konfúziós mátrixokat használtam, továbbá ROC-görbék (Receiver Operating Characteristic curve) segítségével ábrázoltam a modellek diszkriminációs képességét, különösen az osztályozási küszöbök változtatása során, és az AUC-értékek (Area Under Curve) számításával összehasonlítottam a modellek általános teljesítményét. Néhány esetben kevésbé megszokott metrikákat is alkalmaztam, mint például a Hamming-score mutatót, amely a XOR művelet alapján számolja ki a hiba mértékét.

Az iterációk során a modellek beépített értékeléseit külön diagramokon ábrázoltam az evaluate modul segítségével, amely lehetővé tette az egyes metrikák részletes vizsgálatát. A validáció során egyedi függvényt implementáltam, amely a modellek kimeneteit és a valódi címkéket hasonlította össze, így az iterációk közötti különbségek jól nyomon követhetők voltak.

A felsorolt értékelési módszerek eredményei alapján alakítottam és finom-hangoltam a modelleket mindaddig, amíg a lehető legoptimálisabb teljesítményt érték el a rendelkezésre álló adatállományon.

# 4. Szakirodalom

## 4.1 Mély tanulás

Az átfogó elemzés végrehajtásához elengedhetetlen volt a mesterséges intelligenciához köthető eszközök alkalmazása. Pontosabban, különböző mély tanuláson alapuló modellek használata. A számítógépek számára ezen megközelítés, a mély tanulás teszi lehetővé a tapasztalati úton történő tanulást, emberi beavatkozással történő explicit tudás átadása nélkül. Ezzel a megoldással a gépek tulajdonképpen az adott problémát egy olyan hierarchián keresztül ismerhetik meg, ahol minden fogalom az egyszerűbb fogalmakhoz való viszonya által kerül meghatározásra. A fogalmak hierarchiája alapján, tehát egyszerűbb fogalmakból épít fel bonyolultakat, amelynek az összetételét, ha vizualizálnánk, egy rengeteg rétegből álló, mély grafikont kapnánk. A mesterséges intelligencia használatának ezen megközelítésének neve, mint mély tanulás, ezen okból ered.(Bengio et al., 2015)

A gépi tanulás során, a világról valós tudás ismeretével olyan szubjektívnak tűnő döntéseket hozhat az adott algoritmus vagy modell, amelyet a nyers adatokból kinyert és felismert minták tesznek lehetővé. Ezen modellek teljesítménye erős összefüggésben van a felhasználni kívánt adatok reprezentációjával, melynek problémájára megoldást jelenthet a mély tanulás implementálása. Ennek egy klasszikus példája a többrétegű perceptron, amely a bemeneti adatokat egy matematikai függvényként térképezi fel a kimenetre, ahol minden réteg az adatok egy újabb reprezentációját adja. Ezáltal lehetővé teszi az adatok megfelelő reprezentációjának elsajátítását, rétegről rétegre feldolgozva adatokat. Minél mélyebb egy hálózat felépítése, annál nagyobb feldolgozási kapacitással rendelkezik, hiszen képes több utasítás szekvenciális végrehajtására, ahol az egyes rétegek eredményei visszacsatolhatók a következő lépésekhez.(Bengio et al., 2015)

A felügyelt tanulás egy olyan gépi tanulási megközelítés, ahol a feldolgozandó adathalmaz jellemzőihez társítható egy címke vagy célérték is. Ez a kifejezés tehát abból ered, hogy a célértékeket egy olyan tanár biztosítja az algoritmus számára, mely pontosan megmutatja, miként cselekedjen. A klasszifikáció például klasszikus felügyelt tanulási eljárásnak számít. (Bengio et al., 2015)

Generalizációnak nevezik a gépi tanulás azon képességét, amikor a modell nem csak a tanításához használt adatokon teljesít megfelelően, hanem új, korábban nem látott adatokon is. A generalizációs hiba pedig az új bemeneteken várható hiba mértéke, amit egy olyan teszt adathalmaz alapján számol ki az algoritmus, amely tartalma eltér a tanító halmazétól. A független és azonos eloszlás feltételezése mellett, azaz, hogy a tanító és validációs halmazok ugyanazon valószínűségi eloszlásból származnak, lehetőség nyílik a tanító és teszt hiba kapcsolatának számítására. A modell abban az esetben válik alultanult állapotúvá, ha nem képes a tanító halmazon elég alacsony hibát elérni. Túlillesztés, avagy túltanult állapot pedig akkor fordulhat elő, ha a tanító és teszt hibák közötti különbség értéke túl nagy. Egy gépi tanulási algoritmus akkor teljesít a legjobban, ha kapacitása megfelel a feladat valódi összetettségének és a rendelkezésre álló tanító adatok mennyiségének.(Bengio et al., 2015)

### 4.1.1 PyTorch

A PyTorch egy nyílt forráskóddal rendelkező keretrendszer, amit a Facebook csapata kifejezetten mély tanulási algoritmusok kezelésére fejlesztett ki, python környezetben. Két nagyobb neurális hálózat könyvtár koncepcióit vették hozzá alapul, a Lua alapú Torch és a japán Chainer ötvözete, ezzel széles körben elterjedt, rugalmas és dinamikus hálózatépítést tettek lehetővé. (Pointer, 2019)

Különböző adattípusokat támogat, kép- és hang alapú, valamint szöveges adatokat. Alapkoncepciója a tenzorokkal való dolgozás, amelyek speciális többdimenziós tömbök. Ezek lehetnek skalár, vektor vagy mátrix típusúak akár. Az adatfeldolgozás egységesen data setekkel és data loaderekkel történik, utóbbi végzi el a hálózatba történő adat átadását.(Pointer, 2019) Leggyakrabban használt aktivációs függvénye, többcímkés osztályozási hálózatok esetén az úgynevezett softmax, avagy normalizált exponenciális függvény, amely az értékeket nulla és egy közötti tartományba alakítja át, egy egészet összegezve, ezzel biztosítva az eredmény pontos valószínűségi eloszlását. (Goyal et al., 2018)

## 4.2 Neurális hálózatok

A mesterséges neurális hálózatokat az emlősök agyának működésének mintájára fejlesztették ki, lehetővé téve az emberi képességek elsajátítását a számítógépek számára, megfigyelési adatokon tanulva. Többek között képes a kép- és beszédfelismerésre, továbbá a természetes nyelvfeldolgozásra. Alapelve olyan egyszerű elemek gyűjteménye, mint a mesterséges neuronok, azaz perceptronok összessége. Ezek számos bináris bemenettel és egyetlen bináris kimeneti értékkel rendelkeznek. A neuronok rétegekbe csoportosulnak, amelyeken belül és között kapcsolatok állnak elő, ezzel létrehozva a hálózatot. A komplexitásuk függ a neuronok és azok szomszédos kapcsolatainak számától. (Goyal et al., 2018)

A neurális hálózatok bármennyi rétegből és neuronból állhatnak, azonban a leggyakrabban alkalmazott struktúrát három réteg alkotja, a bemeneti, rejtett és kimeneti rétegek. A bemeneti neuronok passzívak, nem módosítják az adatokat, egyetlen értéket küldenek tovább többszörözve a kimenet felé. Ezzel szemben, a rejtett és a kimeneti réteg csomópontjai aktívak, tehát módosítanak az adaton. Az adatmanipulációs képessége a súlyok optimális kiválasztásán és meghatározásától függ, amely a tanulási algoritmus feladata, statisztikai példák alapján. Ezen tanítási példák minősége és érvényessége kimondottan fontos, hiszen a teljesítmény kedvezőtlen romlását eredményezheti, ha túl nagy különbségek szerepelnek a tanító és validációs adathalmaz mintáiban. (Steven W. Smith, 1999)

Azonban, a hagyományos neurális hálózatok nem képesek szekvenciális adatfeldolgozásra, tehát nem alkalmasak összefüggések feltárására korábbi és jövőbeli események között. Ennek megoldására jött létre a visszacsatolt neurális hálózat (RNN), mely folyamatában minden lépésénél meghív egy ismétlődő függvényt, amely az időbeli adatok halmazának modellezésére szolgál. Ezáltal képes nagy mennyiségű korábbi információ eltárolására és hatékony továbbadására. Így a predikció során figyelembe veszi a memória tényezőjét, a jelenlegi és korábbi bemenetek alapján, biztosítva a rejtett rétegek közti kapcsolatok létrejövetelét. (Goyal et al., 2018)Hátránya azonban az, hogy amennyiben a bemeneti rés nagy, nem képes megtanulni a bemeneti adatok releváns információit, limitált a hosszú távú függőségek kezelése. (Yu et al., 2019)

A hálózatok tanítása során, ahogyan az adatok áthaladnak a hálózat rétegein, a veszteségfüggvény segítségével meghatározzák a predikció és a tényleges címke közötti különbséget, majd ezt az információt felhasználva frissítik a súlyokat, a veszteségfüggvény minimalizálása érdekében. A hálózat frissítéseinek végrehajtásához használatos az optimalizáló függvény, amely a klasszifikációs problémák esetén leggyakrabban alkalmazott, az Adam. (Pointer, 2019)

### 4.2.1 LSTM

Az egyik legelterjedtebb visszacsatolt neurális hálózat a Long Short-Term Memory (LSTM) típusa, amely képes változó hosszússágú szekvenciák feldolgozására, ezzel kiküszöbölve az RNN korlátait. Ezt a cellaszerkezetbe integrált kapufüggvények segítségével teszi lehetővé. Típusait tekintve, létezik LSTM felejtő kapuval és anélkül, illetve peephole kapcsolattal, utóbbi a cella állapotába biztosít közvetlen betekintést a kapuk részére. Legtöbb esetben a felejtő kapuval ellátott változat kerül alkalmazásra.(Yu et al., 2019) Ezen kapuk szabályozzák az új tartalom megjegyzésének mértékét, a korábbi törlését, valamint az aktuális tartalom hozzáférhetőségét. (Cheng et al., 2016)A hosszú távú függőségek kezelésének alapelve egy lineáris hurok jelenléte, amelyen keresztül a gradiensek hosszabb időn keresztül áramolhatnak. (Bengio et al., 2015)

A hálózat kapacitásának és mélységének növelésére az LSTM rétegek egymásra helyezése a legegyszerűbb módszer, ezzel egy többrétegű LSTM hálózatot létrehozva, amely az egyik legalapvetőbb szerkezeti forma. (Yu et al., 2019) Ezen fajta hálózat sajátosságai okán, kifejezetten jól alkalmazható chatbotok, szekvenciális minta felismerés, videó- és audió klasszifikációs feladatok, valamint szentiment analízis esetén. (Goyal et al., 2018)

## 4.3 Természetes nyelvfeldolgozás

A természetes nyelvfeldolgozás (NLP), legegyszerűbb értelemben a számítógépek vagy rendszerek azon képessége, amellyel az emberi nyelvet megértik és feldolgozzák, ugyanolyan módon, ahogy azt az emberek is tennék. Ezen képesség elsajátítását azonban nehezítik olyan, az emberi párbeszéd során kimondatlan információk, mint a mimikák és gesztusok, vagy a kétértelmű szavak és kifejezések. Ennek ellenére mégis számos alkalmazási területtel rendelkezik az NLP, mint a szövegösszefoglalás vagy szövegcímkézés, nevesített entitás felismerés, chatbotok és beszédfelismerés feladatai. Ebből kiindulva két egységre osztható a nyelvfeldolgozás, a természetes nyelv megértésére (NLU), továbbá generálására (NLG). (Goyal et al., 2018) Előbbi az emberi nyelv strukturálatlan formájának az algoritmus számára érthető kóddá alakítását segíti. (L. Ashok Kumar & D. Karthika Renuka, 2023)

Ennek megvalósítása érdekében különböző eljárások szolgálnak a bemeneti adatok vektorizált formára alakítására. A vektortér modellek által az N-dimenziós térben ábrázolt szavak lehetővé teszik a hasonló szövegek csoportosítását az új vektortérben. A szóbeágyazási modellek hatékonyan kezelik ezt, azáltal, hogy minden szó környezetét is figyelembe veszik, a szövegben rejlő információkat is felhasználják a modell tanításához. Ezáltal sűrű vektorokká alakulnak a szóbeágyazások, amelyek egy folytonos vektortérben reprezentálják a szavakat. A beágyazás tehát a szó koordinátáit jelenti a vektortérben. (Goyal et al., 2018) A kapott vektornak értéket a TF-IDF eljárással biztosítanak a legtöbb esetben, hiszen az adott szó előfordulásának és az inverz dokumentum gyakoriságai alapján rendel súlyt hozzá. (Wehnert et al., 2021)

### 4.3.1 Téma modellezés

A témamodellezés az egyik leghatékonyabb technika az NLP és azon belül a szöveges adatbányászat terület, látens adatok felfedezésére és szöveges, sokszor rendezetlen dokumentumok közötti kapcsolatok feltárását teszi lehetővé. Az egyik legnépszerűbb módszere a Látens Dirichlet-allokáció (LDA), egy olyan valószínűségi modell, amely koncepciója a látens témák véletlenszerű keverékeként való ábrázolása, egy adott témát alkotó szavak eloszlásának jellemzésével. Tehát, egy nem felügyelt generatív valószínűségi módszer, amely feltételezi, hogy minden dokumentum valószínűségi eloszlással jellemezhető a látens témák felett, valamint a dokumentumokban található témák eloszlása közös Dirichlet-eloszlással rendelkezik. Ezen eljárás különösen alkalmas diszkrét adatok bemutatására, és hatékony megközelítést kínál az óriási adathalmazok, korpuszok rejtett mintáinak, szemantikájának azonosítását és a témák számszerűsítését. Továbbá, lehetővé teszi a dokumentumok szavaiból történő szókészletek képzését, amelyek ezt követően rejtett témák felfedezésére alkalmazhatóak. (Jelodar et al., 2019)

LDA modulokat kínáló megoldások közül a Gensim könyvtárát említeném, amelyen belül hatékonyan lehetséges az LDA modellek egy tréningkorpuszon való becslése, valamint a témák eloszlásának következtetése új, korábban nem látott szöveges adatokon. (Goyal et al., 2018)

## 4.4 Nagy nyelvi modellek

A nagy nyelvi modellek (LLM) képesek szöveg feldolgozásra és generálására, értelmes és koherens kommunikációra, továbbá különféle egyéb nyelvi feladatok hatékony elvégzésére. Az LLM-ek emellett olyan képességekkel is rendelkeznek, mint a logikai következtetés, tervezés, döntéshozatal és a kontextus-alapú tanulás. Ezen tulajdonságok a modellek hatalmas méretéből származnak, még abban az esetben is, ha az előzetes tanítás nem kifejezetten ezen képességek elsajátítására irányult. Ezek a modellek emberi szintű teljesítményt érnek el sokféle feladatban, azonban mindez magasabb erőforrás igénnyel, lassú tanítási sebességgel, valamint jelentős működési költségekkel jár. (Naveed et al., 2023)

Az NLP, avagy LLM modellek fő adat gyűjteményeit korpusznak nevezik, amelyeket vagy írott szövegek alkotnak, vagy rögzített beszéd átiratok. (Khurana et al., 2023) A modell tanítását megelőzi az adatállomány tokenizálása, amely folyamat során a szöveget kisebb egységekre, tokenekre bontanak. A tokenek lehetnek karakterek, szóelemek, szimbólumok vagy szavak, a tokenizálási folyamat típusától függően. Ezen eljárások biztosítják a modellek optimális bemenetének formátumát, valamint a szöveg hatékony feldolgozását. (Naveed et al., 2023)

Elsősorban a karakter-alapú és szó-alapú tokenizálás az elterjedtebb. A szó-alapú esetén a szöveget értelemszerűen a szavak mentén felbontják, és minden szót egy egész számhoz rendelnek. Ez a módszer megrövidítheti a tanítási szakasz időtartamát és csökkentheti a modell bonyolultságát, azáltal, hogy a modell elkerüli a karakterekből történő tanulást. Ez a megközelítés különösen hasznos lehet, ha a szövegben szereplő szavak jól definiáltak és többször előfordulnak, azonban nagyobb szókészlet esetén a modell mérete és összetettsége jelentősen megnövekedhet. (Lewis Tunstall et al., 2022)

A figyelem mechanizmusa súlyokat, avagy figyelmi értéket rendel a bemeneti tokenekhez fontosság szerint, így a modell nagyobb hangsúlyt fektethet a releváns tokenekre. (Naveed et al., 2023)

A Hugging Face Transformers könyvtárja előre betanított, úgynevezett transzformer modelleket kínál, illetve különböző API-kat a modellek további tanításához, finomhangolásához, testreszabásához. Ezen modelleket maszkolt nyelvi modellezés (MLM) útján tanították, vagyis véletlenszerűen tokeneket vagy azok sorozatát elfedték, majd a múlt- és jövőbeli kontextus alapján jeleztették előre a modellel a maszkolt tokenek értékét. Továbbá, az első szakaszban a modelleket nagy korpuszon tanították a bemenetek alapján a következő tokenek előrejelzésében, ez az alapvető pretraining. Ezt követően a modell adapterrétegekkel kerül finomhangolásra konkrét feladatokra. Az LLM-ek képességeinek fejlesztésében ez az eljárás játszik alapvető szerepet. Ezen transzformer modelleknél a figyelmi mechanizmus lekérdezés, kulcs és érték mátrixokat párosít a bemeneti szekvenciákhoz, ahol a figyelem értékét a lekérdezés és a kulcs szorzata képezi. (Naveed et al., 2023) Az önfigyelem, vagy selfattention során a súlyok ugyanazon blokkból kerülnek kiszámításra minden rejtett állapotra. Előnye abban rejlik, hogy a meghatározott token beágyazások helyett a teljes bemenetet átlagát kiszámítja minden beágyazott tokenre, azaz lehetővé teszi a párhuzamos számítást. (Lewis Tunstall et al., 2022)

Modellarchitektúrákat tekintve, az LLM-ek felépítése állhat csak kódoló (encoder) vagy dekódoló (decoder), vagy mindkét blokk együtteséből (encoder-decoder). Az architektúrák különböző feladatokra specializáltak, a csak kódoló modellek, mint a BERT modellcsalád, főként NLU feladatokra alkalmasak. A dekódoló modellek, például GPT modellek pedig az NLG feladatokra optimálisak. A mindkét blokkot tartalmazó architektúrákat jellemzően szekvenciák közötti modellezésre használják, például fordítási feladatokra, ahol a bemeneti és a kimeneti adatok is szekvenciák. (Naveed et al., 2023)

### 4.4.1 BERT

A Bi-directional Encoder Representations from Transformers (BERT) egy olyan előre betanított modell, amelynek korpuszát a BookCorpus és az angol Wikipédia címkézetlen szövegei alkotja. A modell különböző NLP feladatokhoz finomhangolható, mint a szentiment elemzés, szövegklasszifikáció, mondatbeágyazás vagy a szövegbeli kétértelműség értelmezése. Korábbi nyelvi modellek általában csak egy irányban vizsgálták a szöveget, ami elsősorban szöveg generálásához bizonyult alkalmasnak. Ezzel szemben a BERT modell egyszerre mindkét irányban elemzi a szöveget, ami mélyebb nyelvi megértést tesz lehetővé. Kontextusfüggő beágyazást biztosít minden szövegben szereplő szóhoz, valamint legfeljebb 512 tokent tartalmazó szekvenciák kezelésére képes, ezért a hosszabb szövegek kisebb tokenekre bontása elengedhetetlen. (Khurana et al., 2023) A modell a fentebb említett transzformer architektúra kódoló részét használja, valamint a maszkolt nyelvi modellezés és a következő mondat előrejelzésének (NSP) eljárását. A szövegszekvenciái olyan tokenbeágyazások, amelyekben minden token egy rögzített dimenziójú vektorhoz van rendelve, pontosan 768 dimenziós vektorhoz.(Lewis Tunstall et al., 2022)

Mivel a BERT alap modellje is meglehetősen nagyobb méretű és ezzel arányos az erőforrás igénye is, a tudáshígítás módszerével készült egy kompaktabb változata, a DistilBERT. Ez az eljárás különösen elterjedt az LLM-ek tömörítésének alkalmazásában, a praktikus és könnyebb felhasználhatóság jegyében. A tudáshígítás során egy kisebb diák modell arra van képezve, hogy utánozzon egy lassabb, nagyobb, de jobb teljesítményű tanár viselkedését. Felügyelt feladatoknál, például finomhangolás során, a fő elképzelés az, hogy a tanuláshoz szükséges valódi címkék mellé lágy valószínűségi eloszlásokat nyújt a tanár modellje. Ezek a kiegészítő információk segítik a diák modellt a tanulásban, hogy utánozza ezeket a valószínűségeket, olyan információ kivonásával, amely nem érhető el kizárólag a címkékből. A tudáshígítást előtanítás során is alkalmazható egy általános célú diákmodell létrehozására, amely később specifikus feladatokra finomhangolható. Ilyen esetben a tanár egy előre tanított nyelvi modell, például a BERT, amely az általa elsajátított maszkolt nyelvi modellezési tudást adja át a diák modellnek.(Lewis Tunstall et al., 2022)

A BERT modellt tovább optimalizálva és finomhangolva egyéb, elterjedt felhasználói körben alkalmazott modellek jöttek létre. Ilyen a RoBERTa, amely hosszabb ideig és terjedelmesebb tanító adat kötegeken tanult, valamint az NSP eljárásának alkalmazása nélkül, ezzel egy jelentősen jobb teljesítményű modellt eredményezve. Ezen modellt egy lépéssel továbbfejlesztették az XLM többnyelvűségével, a tanító adatok ismételt növelésével, létrehozva az XLM-RoBERTa modellt. (Lewis Tunstall et al., 2022)

## 4.5 Szentiment analízis

Az említett NLU feladatok közül a klasszifikációt szeretném kiemelni, azon belül is a szöveges adatok osztályozását, hiszen ez az egyik leggyakoribb alkalmazása az NLP megoldásoknak. A többosztályos klasszifikáció típusú feladatban az algoritmusnak azt kell meghatároznia, hogy az adott bemenet melyik osztályba tartozik, egy becslési függvény alkalmazásával. (Bengio et al., 2015) Továbbá, rengeteg gyakorlati alkalmazásban megtalálható, mint a levelezőrendszerek spam szűrése, vagy a szentiment analízis is, amely során az adott szöveg polaritását határozza meg egy modell. (Lewis Tunstall et al., 2022)

A szentiment analízis során általában szubjektív elemeket vizsgálnak, melyek lehetnek csupán szavak, kifejezések vagy egész mondatok, akár teljes dokumentumok. Azonban, az érzelem elsősorban kisebb nyelvi egységekben rejlik. Az érzelem polaritása a szöveg egy saját jellemzője. Ez többnyire két részre osztható, pozitív és negatív értékekre, de skálaként is felfogható. Osztályozása során, egy véleménytartalmú szövegrész alapján a vélemény két ellentétes érzelmi polaritás egyikébe kerül vagy a két szélsőség közti folytonos skálán vesz fel értéket. Továbbá, az érzelem polaritása és annak erőssége különböző értékeket jelent. Az ellentétes érzelmek különböző keverékeinek megkülönböztetésére a polaritásosztályozás többlépcsős skálát használ, ilyennek tekinthető például a különböző fórumokon megfogalmazott vélemény csillagjainak száma. Ezesetben többosztályos szövegosztályozásról problémáról van szó. A téma alapú többosztályos klasszifikációval ellentétben, ahol a szókincs osztályonként teljesen különbözik, vagy csak alacsony mértékben fedik egymást, a pozitív, semleges és negatív osztályok szókincsei nagyon hasonló tartalommal bírhatnak, csak néhány kritikus szó eltérésével. A tagadások, amelyeket a szövegelemzések nagy részében gyakran figyelmen kívül hagynak, mint jelentéktelen elemeket, az érzelemvizsgálatban fontos szerepet kapnak, hiszen képesek egy eredetileg pozitív kifejezést negatívvá alakítani, ahogyan fordítva is. (Mejova, 2009)

### 4.5.1 Adatbővítés

Szöveg és érzelem osztályozási feladatok során elengedhetetlen az osztályok eloszlásának előzetes vizsgálata, hiszen egy kiugróan magas egyedszámmal rendelkező osztályt tartalmazó adathalmaz esetén más tanítói eljárás alkalmazandó, mint egy egyenlő eloszlású halmaz esetében. Amennyiben kiegyensúlyozatlanság tapasztalható az osztályok arányai között, különböző adatbővítési eljárások alkalmazhatók, amennyiben nem lehetséges a további adatgyűjtés, például a kisebb osztályok véletlenszerű túlmintavételezése vagy ennek ellenkezője a nagyobb osztályok esetében. (Blagus & Lusa, 2013)

Túlmintavételezési eljárásként a Synthetic Minority Over-Sampling Technique (SMOTE) módszerét szeretném részletezni. Alkalmazásával szintetikus mintákat hoz létre a kisebbségi osztályok részére, a teljesen vagy majdnem kiegyensúlyozott osztályok biztosításának céljával. A szintetikus minták a legkisebb osztály két hasonló egyedének lineáris kombinációi, amelyek véletlenszerű kiválasztás útján kerülnek felhasználásra, az adott minta szabadon meghatározott számú legközelebbi kisebbségi szomszédja közül. A SMOTE minták várható értéke megegyezik az eredeti kisebbségi osztály globális várható értékével, kisebb varianciával. Így az osztályspecifikus átlagértékeken és globális varianciákon alapuló klasszifikáló modellekre nincs szignifikáns hatása, ellenben az osztályspecifikus varianciákat használókkal, a torzított becslés miatt. Valamint, a minták erősen pozitív korrelációval bírnak a kisebbségi osztály mintái iránt. (Blagus & Lusa, 2013)

### 4.5.2 Teljesítménymutatók

Különböző osztályozási modellek vagy gépi tanulási technikák értékelése és összehasonlítása esetén elengedhetetlen a teljesítménymutatók használata. Számos olyan metrika létezik, amelyek a többosztályos osztályozók képességeinek kiértékelésére szolgál. Ezek a mutatók a fejlesztési folyamat különböző szakaszaiban hasznosak lehetnek, például két különböző modell teljesítményének összehasonlításánál vagy ugyanazon modell viselkedésének elemzésénél különböző paraméterek hangolása esetén. (Grandini et al., 2020)

A legegyszerűbb és leghatékonyabb metrikák közé tartozik például a konfúziós mátrix, egy olyan kereszttábla, amelyen a valódi osztályok előfordulásainak számát arányolja a prediktált értékekhez. A mátrix főátlóján helyezkednek el a helyesen osztályozott elemek, amely hőtérképes vizuális reprezentációján a legmélyebb színeket viselik. A precizitás a valós pozitív elemek számát osztja el az összes pozitívként prediktált egyeddel, ezzel a modell pozitív osztályozási biztosságát jelenti. A visszahívás vagy recall értéke a valós pozitív emelek és az összes pozitív elem arányából származik, ezzel a modell előrejelzési képességét méri. A pontosság, azaz accuracy, összességében a legnépszerűbb teljesítménymérő szám a többosztályos klasszifikációs modelleknél, az igaz pozitív és negatív elemek összegét osztja el az összes elem összegével, tehát a modell előrejelzésének helyességét méri. Azonban, kiegyensúlyozatlan eloszlású osztályokon való alkalmazása esetén elfedheti a kisebb mintával rendelkező osztályok súlyos osztályozási hibáit a domináns osztályokéval szemben, ezzel kedvezőtlen teljesítményi értéket eredményezve. Az F1-Score a precision és recall mutatók súlyozott átlaga, értéke nulla és egy között minél magasabb, annál jobb a modell teljesítménye, egyesével minden osztályon. A keresztentrópia függvénye két eloszlásfüggvény hasonlóságát méri, a valódi és az előrejelzett osztályok közötti különbség kiértékelésével, figyelembe véve az egység specifikus jellemzőit. Gyorsan és kis erőforrás igénnyel kiszámítható, így rendkívül széles körben elterjedt a használata költségfüggvényként, annak ellenére, hogy csak a valódi osztály valószínűségét veszi figyelembe, a maradék valószínűségek eloszlásának figyelembevétele nélkül. (Grandini et al., 2020)

A konfúziós mátrixhoz hasonlóan, szintén grafikusan demonstrálja a modell teljesítményét a Receiver Operating Characteristic (ROC) görbe, a teszt adatok értékének egy küszöbértékkel való összehasonlításának alapján. Ha a prediktált, kimeneti érték meghaladja az említett küszöböt, jelen van a célállapot. A ROC görbe ábrázolja a pozitívként azonosított célállapotok százalékát, szemben a tévesen pozitívként azonosított nem célállapotok százalékával, különböző küszöbértékek mentén, vagyis, a görbe minden pontjával a valós és hamis pozitív teljesítmény közti kompromisszumot jelképezi. Minél hatékonyabb az osztályozás és egyúttal a modell teljesítménye, a görbe annál inkább hajlik a grafikon bal felső sarkának irányába. A küszöbértéket képviselő átló, a véletlenszerű osztályozás mértéke 0.5, a tökéletesség pedig az egy egészet jelenti. Ez a mérőszám a görbe alatti terület (AUC) értéke. (Steven W. Smith, 1999)

## 4.6 RAG modell

Ugyan az LLM-ek képesek nagy mértékű, tényszerű ismeretek tárolására, valamint finomhangolás útján kiemelkedő teljesítményt elérni a különböző NLP feladatokon, a tudásuk elérése és pontos manipulációja erősen korlátozott. Emellett, kihívást jelent a döntéseik eredetének nyomon követése, a hallucináció, valamint a világról szerzett tudás frissítése is. Ezen problémákra megoldást nyújtanak a paraméteres-, illetve nem-paraméteres memória kombinációjával ellátott, Retrieval Augmented Generation (RAG) modellek. (Lewis et al., 2020) A RAG tulajdonképpen az LLM-eket külső tudásbázisból történő, dokumentumrészek visszakeresésével egészíti ki, ezzel elérhetővé téve a legfrissebb tudást a nyelvi modell számára. Ezt az integrált megoldást az indexelés, visszakeresés és generálás lépései alkotják. (Gao et al., 2023)

Az indexelés során a felhasználni kívánt, különböző formátumú dokumentumok egységes, szöveges formátumra alakítja, majd kisebb egységekre, azaz chunkokra bontja szét. Összetételüket és méretüket tekintve, lehetnek tokenek de akár komplett dokumentumok is. Ebből kiindulva, a nagyobb chunkok több információval rendelkeznek, magas zaj mellett, míg a kisebbek, a zaj minimális szinten tartása ellenére nem biztos, hogy tartalmaznak releváns információt. Ezen egységeket egy beágyazási modell segítségével vektorrá alakítja, melyek egy speciális, vektor tároló adatbázisban kerülnek eltárolásra a visszakeresések biztosítására. (Gao et al., 2023) A Facebook AI Similarity Search (FAISS) könyvtára az egyik legnépszerűbb megoldás a vektor-tárolók és keresők közül, a legközelebbi szomszéd keresés elvére alapozva, különféle indexelési módszerekkel. Az indexekkel eltárolhatók, és fokozatosan növelhetőek a vektorkészletek. Lekérdezés során a vektorizált kérdés vagy keresési feltétel alapján, az Euklideszi távolságot alkalmazva adja vissza az adathalmazban található legközelebbi vektort. (Douze et al., 2024) A generáció folyamatában a lekérdezett és kiválasztott szöveg, valamint dokumentumelem alapján létrejön egy prompt, amely mentén fogalmaz meg egy választ a modell. Az LLM-ek sajátosságaiból kiindulva, sajnos a hallucináció esete nem zárható ki teljesen, mert a generáció során nem mindig veszi figyelembe a modell a visszakeresett elemet. (Gao et al., 2023)

### 4.6.1 LangChain

A LangChain egy olyan moduláris keretrendszer, amely az LLM-eket hasznosító alkalmazások fejlesztésére szolgál, például egy integrált RAG modell létesítésére, vagy chatbotok építésére. Különböző komponensekkel, testreszabható, konkrét felhasználási esetekhez igazított folyamatokkal, azaz láncokkal biztosítja a különböző alkalmazások kompatibilitását. Főbb komponensei közé tartoznak a promptok, amelyek az LLM bemeneti adatait alkotják, kérdésekkel, példákkal és specifikus utasításokkal. Alapvető építőelemei a láncok, amelyek az adott nyelvi modellt és az utasítási halmazát, promptját kombinálva biztosít lehetőséget különböző NLU és NLG feladatok elvégzésére. Az egyszerű láncok egy bemenettel és kimenettel rendelkeznek, különlegességük abban rejlik, hogy több lánc összekapcsolása esetén az egyik láncszem bemenetét a korábbi kimenete alkotja. A keretrendszer népszerűségéhez hozzájárul az adatfeldolgozási funkcionalitásainak sokszínűsége is, egyszerűen biztosítja a dokumentumok betöltését a leggyakrabban használt formátumokból, a modellek hatékonyabb feldolgozását elősegíti dokumentum átalakítókkal, szövegbeágyazási modellekkel. A beágyazott adtaok tárolására saját vektortárolók is felhasználhatók és az itt végzett keresésre szintén biztosít eszközöket, retrieverek formájában. (Topsakal & Akinci, 2023)

# 5. Esettanulmány

## 5.1 Adatok ismertetése

A kutatásomat az adatgyűjtés lépésével indítottam. A három közösségi autómegosztó szolgáltató applikációira adott értékeléseket töltöttem le a Google Play és AppStore áruházakból, erre a célra fejlesztett python könyvtárak segítségével (google\_play\_scraper, app\_store\_scraper). Igyekeztem minden rendelkezésre álló adatot összegyűjteni és később felhasználni, így nem csak a magyar, de az angol nyelven írt értékelésekre is szűrtem. Mivel a wigo az arculatváltást követően új applikáció letöltésére szorgalmazta felhasználóit, így a korábbi, ShareNow alkalmazás értékeléseit is igyekeztem hozzáadni az adathalmazomhoz, ügyelve rá, hogy csak Magyarországon adott értékeléseket vegyen figyelembe a scraper. Mindkét felhasznált könyvtár automatikusan pandas dataframe formátumban adta vissza a kinyert adatokat, minden meta adatnak külön oszlopot létrehozva. Az oszlopok számát ritkítottam, ugyanis az elemzésembe nem vettem bele, hogy az adott értékelésre a fejlesztő válaszolt-e, milyen dátummal keletkezett az értékelés, milyen verziószámú applikációt használt, más felhasználó hasznosnak találta-e stb. A két forrásból kinyert, összesen nyolc darab, kisebb táblát egybefűztem és csv, valamint pkl formátumban letöltöttem a lokális tárhelyemre, hogy az elemzett modellekben könnyedén hasznosíthassam. Összesen 1660 sor adatot gyűjtöttem össze ezzel a módszerrel, az értékelések és applikációk közti eloszlást pedig az alábbi diagram szemlélteti.

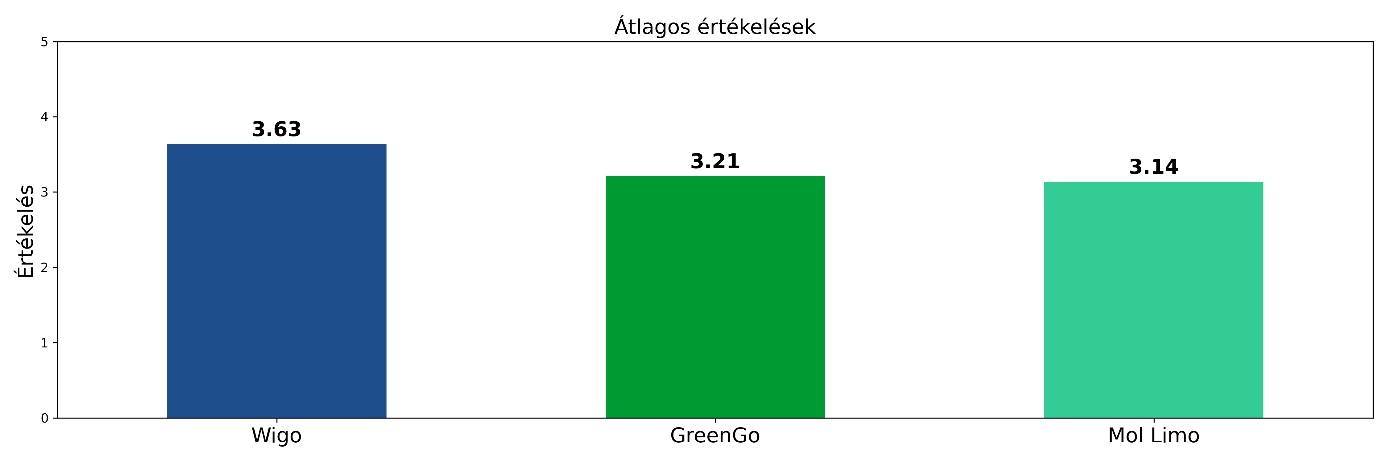
A képen diagram látható

Automatikusan generált leírás

1.ábra: A szolgáltatók értékeléseinek eloszlása. Forrás: saját szerkesztés.

Az ábra alapján egyértelműen megállapítható, hogy a felhasználók a szélsőségek felé hajlanak, a két, három és négy csillagos értékelések száma sok esetben az egy és öt csillagos értékelések számának a harmada. Az osztályok közötti egyensúly ilyen mértékű hiánya gondot okoz a modellek tanítási szakaszában, így szükséges volt a halmaz későbbi augmentációjára.

A mélytanuló modellek implementálást megelőzően, az adatokon egy előzetes elemzést végeztem, a felhasználói minták általánosabb felfedésére. Ezen okból, az átlagos értékelések alakulása az alábbi ábrán látható. Ennek eredménye fordított arányban reprezentálja az értékelések mennyiségét, ugyanis az összességében legkevesebb értékeléssel a wigo rendelkezik, ezt követi a GreenGo, majd a Mol Limo. Mivel nem tapasztalható nagy variancia, ez arra enged következtetni, hogy talán a későbbi modellek alapján sem lesz megállapítható számottevő különbség a szolgáltatók minősége között.



2. ábra: A három szolgáltató értékeléseinek átlaga. Forrás: saját szerkesztés.

## 5.2 Téma modellezés

Annak érdekében, hogy mélyebb betekintést nyerjek a felhasználói értékelések mintázataiba és tartalmába, különböző adatvizualizációs technikákat alkalmaztam. A szöveges adatállomány elemzését az értékelések hosszának vizsgálatával kezdtem, ahol látványos különbségek mutatkoztak meg a vizsgált szolgáltatók között.

A képen szöveg, képernyőkép, diagram, Diagram látható

Automatikusan generált leírás

3. ábra: Az értékelések karaktereinek számának eloszlása. Forrás: saját szerkesztés.

A mellékelt diagramon továbbá olyan egyszerű statisztikai mutatók is megjelennek, mint a három applikációra adott értékelésekben előforduló szavak számának átlaga, mediánja és szórása. Ezen mutatók tudatában még szemléletesebb az ábra, az alkalmazások közötti eltérések és hasonlóságok közt. Ez a jelenség eredhet az eltérő ügyfélélmények jellegéből, valamint a felhasználói rétegek különböző kommunikációs stílusából. Az átlagos, 25 karakterszámnál hosszabb vélemények tükrözhetnek nagyobb elégedettséget, vagy éppen súlyosabb panaszokat, azaz erősebb érzelmeket.

Egyszerű, de szemléletes módszerként a szófelhők vizualizálását alkalmaztam, az angol és magyar nyelvű stop szavak eltávolítását követően. Egyéb, a kontextusban különösebb jelentést nem nyújtó, de sűrűn előforduló szavak mellőzését is elvégeztem, mint az autó vagy app szavak. Ez a megközelítés már alaposabb betekintést nyújtott az értékelések tartalmi mintáinak felismeréséhez, valamint a felhasználók fókuszpontjainak azonosításához. A mellékelt ábrán főleg pozitív töltetű szavak jelennek meg, ami nem véletlen, hiszen a három alkalmazáshoz tartozó értékelések legnagyobb hányadát az öt csillagos teszi ki. Mivel a második legnagyobb hányadot az egy csillagos értékelések képzik, negatív töltetű kifejezések is megjelennek a halmazokban. Ezeken túl, az alkalmazások elnevezései is megfigyelhetők, mint visszatérő motívum, ahogy általánosabb, a szolgáltatási egységekre vonatkozó szavak is fellelhetőek, mint az ügyfélszolgálat vagy az ingyenes regisztráció.

A képen szöveg, Betűtípus, kézírás, tervezés látható

Automatikusan generált leírás

4. ábra: A szolgáltatók értékeléseiben leggyakrabban előforduló szavak. Forrás: saját szerkesztés.

Az adatok további ismertetéséhez szövegalapú klaszterezési vizualizációt is alkalmaztam. A t-SNE alapú dimenziócsökkentés lehetővé tette a vélemények klaszterekre bontott vizuális csoportosítását. Ez a módszer segített azonosítani a közös mintázatok metszetét, ezzel párhuzamosan szemléltetve a szélsőséges előfordulásokat is. A scikit-learn könyvtár TfidfVectorizer moduljának segítségével ismételten megtisztítottam az adatokat a stop szavaktól és feldolgoztam a klaszterezés implementálása érdekében. Az ábrán jól látszik, hogy mindhárom alkalmazás értékelései közel hasonló, ha nem teljesen azonosítható klaszterbe sorolható. Kifejezetten nagyobb elkülönülések nem figyelhetők meg, ez azt sugallja, hogy az összesen öt osztályba tartozó értékelések szentimentjei közt nem azonosíthatóak nagy eltérések, jelen van egy bizonyos mértékű átfedés. A nagyobb távolságok utalhatnak eltérő véleménystruktúrákra vagy témákra, ezek a kiugró, ritkábban előforduló adatok.

A képen képernyőkép, diagram, Diagram látható

Automatikusan generált leírás

5. ábra: Dimenziócsökkentett ábrázolás a véleményekről. Forrás: saját szerkesztés.

A teljeskörű elemzésbe továbbá beletartozott egy téma alapú modellezés, Látens Dirichlet Allokáció (LDA) eszközével, azzal a céllal, hogy strukturáltan feltárjam a felhasználói visszajelzések főbb témáit. Az előforduló értékelések halmazában négy, egyértelműen elkülöníthető téma került azonosításra. Ezen domináns témák, alkalmazások közti eloszlásának vizualizációját az alábbi kördiagram szemlélteti. Jól látható, hogy nagyjából egységesen volt képes felosztani a modell az értékelések mintáit, szolgáltatókként minden téma 22 és legfeljebb 28 százalékban van jelen, hasonló eloszlást mutatva.

A képen diagram, Betűtípus látható

Automatikusan generált leírás

1. ábra: Az azonosított témák százalékos megoszlása. Forrás: saját szerkesztés.

A téma elemző modell eredményeit PyLDAvis interaktív vizualizációval szemléltettem, amely segített mélyebben megérteni a témák tartalmát és az értékelések sokszínűségét. Ez az integrált elemzési technika lehetőséget teremtett arra, hogy az értékelések tartalmi sokrétűségét átfogó módon mutassam be. Az ábra kezdeti állapotán jól látható az azonosított négy fő téma, amelyek egymástól elkülönülve, uniók alkotása nélkül helyezkednek el. A körök egymástól való távolsága jelzi a témák közti különbségek mértékét, ezalapján a második, illetve negyedik témák tartalma hasonlít a leginkább. Az ábrán továbbá felsorolva látható a 30 leggyakrabban előforduló kifejezések, kulcsszavak listája, gyakoriságukkal együtt jelölve. A diagram további vizsgálata alapján kiderül, a felhasználók milyen aspektusok és motivációk alapján fogalmazzák meg értékeléseiket. Az első témában promóciókra és regisztrációra vonatkozó kulcsszavak szerepelnek, míg a második téma a szolgáltatásra és a használhatóságra fókuszál. A harmadik téma elemei alapján az általános ügyfélélmények azonosíthatóak, az utolsó, negyedik témában pedig maguk a márkák közvetlenül jelennek meg.

A képen szöveg, képernyőkép, diagram, kör látható

Automatikusan generált leírás

7. ábra: LDA modellezés alapján a témák távolsága. Forrás: saját szerkesztés.

### 5.1.1 Kibővített adathalmazok

A tanuló algoritmusok, képességeikből eredően az eredeti, kiegyensúlyozatlan állományon kevésbé értek el kielégítő teljesítményt. Az eredmények javítása érdekében többféle módon kibővített adatállományokkal kísérleteztem, ezeket ismertetném a következőkben.

A legkézenfekvőbb módszer az osztályok olyan jellegű kiegyenlítése volt, ahol a mintavételt manuálisan a legkisebb osztályhoz mérten, legfeljebb 200 egyedre módosítottam. Hasonló, de mégis ellenkező elven a SMOTE technikát alkalmazva, automatikusan feldúsít minden osztályt, a legtöbb egyeddel rendelkező osztállyal megegyező mennyiségre. Ennek megvalósításához ismételten a TF-IDF vektorizáló modullal numerikus formára alakítottam az adatokat, majd a t-SNE klaszterezés eredménye és az alacsony egyedszámú osztályok alapján kisebb, pontosan három darabra állítottam a legközelebbi szomszédok paraméterét. Így a szintetikus adatok automatizált előállításával minden osztály, egységesen 704 darab egyeddel rendelkezett.

További változatok létrehozására nagy nyelvi modellek által támogatott eljárásokat alkalmaztam. Az egyik egy manuálisabb megközelítést jelent, azaz egy GPT-4 modellel generáltattam az alul reprezentált osztályok egyedei alapján szinonimákat, amelyeket hozzáfűztem az eredeti adathalmazhoz. A másik megoldás pedig a visszafordítás (back translation) módszere volt, melyhez a Helsinki-NLP által fejlesztett két, nyelvpárokon működő, fordításra optimalizált modelljeit alkalmaztam. Az algoritmus a teljes, eredeti adathalmazt először angolra, majd visszafordította magyar nyelvűre. A hatékony generálás érdekében, tíz mintás kötegekkel dolgozta fel az adatokat, melyek tokenizálást követően kerültek feldolgozásra. Az így kapott egyedekkel kibővítettem az állományomat, amit további kétféle módon teszteltem a modellekkel, a duplikált sorok törlésével, illetve azok megőrzésével.

A képen képernyőkép, szöveg, diagram, Diagram látható

Automatikusan generált leírás

8. ábra: A három felhasznált adathalmaz osztály eloszlásai. Forrás: saját szerkesztés.

## 5.2 Klasszifikáló modellek

A szentiment analízishez BERT alapú nyelvi modellek különböző variációit futtattam, beleértve kompaktabb, többnyelvre feltanított, illetve kimondottan a magyar nyelvre optimalizált modelleket, a lehető legoptimálisabb eredmény elérésének céljával. A transzformer alapú modellek mellett egy hagyományosabb szövegfeldolgozó eljárást is kipróbáltam, az LSTM neurális hálót, amely inkább viszonyítási alapként szolgált a két megközelítés között. Valamennyi algoritmus az augmentált halmazokon is alkalmaztam, így egyes modellek teljesítményéről és képességeiről részletesebb képet alkothattam. Az alábbiakban az elemzéshez felhasznált modelleket, azok paraméterezését, valamint eredményeik összehasonlítását szeretném bemutatni.

### 5.2.1 DistilBERT

A kutatásomban először a DistilBERT modelljére esett a választásom, több nyelven való optimalizáltsága és kompakt mérete okán. Utóbbi a 66 milliós paraméterezettségéből fakad, ugyanis az alap BERT modell 110 millió paraméterrel rendelkezik. Ez a különböző folyamatok futtatási idejében jelentős, még CUDA használata esetén is. A modellt összesen kétféle metrika együttessel futtattam, négyféle adathalmazzal.

A tanítást először az eredeti adathalmaz felosztásával kezdtem, az sklearn train\_test\_split függvényének segítségével külön tanító és validációs halmazokra, 80 és 20 százalékos megoszlásban. Ezt követően a nagyobb méretű adathalmazt a tesztelés szándékával tovább bontottam, 99 százalékában továbbra is tanító adathalmazra, a fennmaradó egy százalékban pedig a tesztelésre. A következő lépés az előállt állományok tokenizálása, valamint kötegelése a tanítási szakaszt megelőzően, ezzel biztosítva a megfelelő adatfeldolgozást. A tanítás során a modell egy olyan finomhangoláson esik át, amellyel képes klasszifikálni a megadott címkék, tehát a csillagok száma alapján a felhasználói értékeléseket. Optimalizációs eszközként az egyik legelterjedtebb, kimondottan osztályozási feladatokra szánt optimizer egyik típusát használtam, az AdamW-t, ami a modell túltanulását igyekszik megakadályozni. Tanulási sebességként 0.00005 értéket határoztam meg, ami ugyan valamivel lassabb, de stabilabb tanulást ígér. Mivel többcímkés osztályozási feladatról van szó, ezért a veszteségek számítására a kategorikus keresztentrópia költségfüggvényét alkalmaztam. Ezen modell összesen három teljes iteráción keresztül tanult, közepesen jó teljesítményt elérve. A validációs pontosság értéke 0.62, a veszteségé összességében 1.18, az F1-score pedig 0.56 értéket ért el. A kapott eredmények egyértelműen az adatállomány kiegyensúlyozatlanságából ered, ugyanis az öt lehetséges osztályból csupán hármat volt képes prediktálni. Ugyanezen paraméterekkel, az osztályonként, egységesen 200 egyedet tartalmazó halmazon romlott a teljesítménye a modellnek. A pontosság így 0.38 értékre csökkent, ezzel párhuzamosan a veszteség 1.47 értékre növekedett, az F1-score pedig a pontossághoz hasonlóan, 0.31 értéket eredményezett. Az eredeti modell eredményétől szintén kedvezőtlenebbül teljesített a GPT-4 által augmentált állomány, bár nagyobb méretéből adódóan a második variációnál optimálisabb értékeket mutatott. A hiperparaméterek közül egyedül a tanulási sebességen módosítottam még lassabbra, 1e-5, azaz 0.00001 értéket állítottam be. A pontosság 0.5, a veszteség 1.27 értéket kapott, az F1-score pedig 0.46 értéket eredményezett. A legutóbbi paraméterekkel megegyezően a legnagyobb, visszafordítással előállt állományt is feltanítottam, ami minimálisan optimálisabb teljesítményt nyújtott, mint az eredeti. A korábbi három prediktált osztály helyett négyet volt képes visszaadni, 0.66 pontossággal, 0.91 veszteséggel és 0.64 F1 értékkel.

A DistilBERT nyelvi modellt egy kissé másfajta megközelítésben is kipróbáltam, a korábbitól eltérő metrikák alkalmazásával. Ezen algoritmusokban a modell címke-egyezési pontosságát (Hamming Score) és a hibaarányát (Hamming Loss) is vizsgáltam. Míg az első megközelítésnél használt metrikák inkább az egyedi minták teljes helyességét vették figyelembe, a Hamming-alapú metrikák fókuszában a predikciók részleges egyezéseinek és hibáinak értékelése áll. Az adatfeldolgozás hasonlóan történik, költségfüggvényemként egyszerű keresztentrópiát alkalmaztam, Adam optimizert és 1e-05 értékű tanulási sebességet. Négyes kötegekben először egyetlen iteráción keresztül folyt a tanítás, majd ciklikus, iteratív módon igyekeztem a legoptimálisabb teljesítményt nyújtó hiperparamétereket megtalálni. Összesen háromféle adathalmazon, az eredetin, a legfeljebb 200 egyedszámra kiegyensúlyozotton, illetve a GPT-4 által bővítetten futtattam a modellt. Az eredetin, a kezdeti paraméterekkel értem el a legjobb eredményt, ahol az egyezés értéke 0.7, a hiba pedig 0.84 volt. A legkevesebb adattal rendelkező modell szintén kedvezőtlenebb eredményt ért el, 0.49 egyezéssel és 0.74 hibaaránnyal. Az említett állományok közül pedig a legnagyobbon értem el ezen modell talán legkedvezőbb teljesítményét, 0.75 egyezési rátával és 0.76 hibával.

### 5.2.2 XLM-RoBERTa

A következő nyelvi modell variáns, amelyt kipróbáltam az XLM-RoBERTa Base volt. Méretét tekintve lényegesen nagyobb, mint a kompakt DistilBERT, ugyanis 270 millió paraméterrel rendelkezik, ezzel közepes méretű modellnek nevezhető. Összesen kétféle finomhangolt változatát tanítottam fel, az eredeti, illetve a visszafordítással bővített adathalmazokon.

Első megközelítésként egy olyan változatát használtam a modellnek, ami Twitter bejegyzéseken került finomhangolásra és háromféle címkére, avagy polaritásra képes osztályozni; negatív, neutrális és pozitív szentimentekre. Az adatfeldolgozás természetesen ezesetben is a tokenizálás folyamatával zajlott, majd a polaritások meghatározásához softmax függvénnyel biztosítottam az értékek valószínűségekhez történő rendelését. Mivel eredetileg öt osztályos problémával dolgoztam, az eredmények kinyeréséhez a három polaritási érték alapján egy prediktált osztályt fűztem az adatokhoz. Ennek megvalósításához egy olyan függvényt definiáltam, amely a negatív érzelmi töltetű értékeket 0.6 felett az egy csillagos címkéhez rendelte, alatta pedig a ketteshez. A neutrális értékek esetében a hármas osztályt határoztam meg, továbbá, a pozitív predikciókat szintén a 0.6 küszöb alapján soroltam ötös vagy négyes osztályokba. Így a súlyozott F1 értéke 0.43-ot eredményezett az eredeti adathalmazon, 0.39 értékű pontossággal. Következő megközelítésben egy magyar nyelvre finomhangolt változatát használtam a modellnek, ami ugyancsak három osztály predikciójára alkalmas, mellette magabiztossági értékeket visszaadva. Nyolcas kötegekben tanítottam fel, minden más paraméterét alapértelmezett beállításokkal hagyva. A kapott értékeket a korábbi elv alapján átalakítottam, hogy megfeleltethessem a korábbi öt osztállyal, 0.54 értékű pontosságot elérve. Az F1 súlyozott átlaga ugyancsak 0.54 értéket eredményezett, valamint az alábbi konfúziós mátrix alapú hőtérkép alapján jól látható az osztályok eloszlása.

A képen szöveg, képernyőkép, diagram, Téglalap látható

Automatikusan generált leírás

9. ábra: Konfúziós mátrix a RoBERTa teljesítményén. Forrás: saját szerkesztés.

Ugyanezen két szövegelemző rendszer variációját feltanítottam a visszafordítással augmentált adatállományon szintén. Az első elemzési folyamat teljesítménye ezesetben szignifikánsan romlott, 0.21 értékű pontosságot és 0.21 súlyozott átlagolt F1 értéket eredményezett. Különös módon, a nagyobb és kiegyensúlyozottabb halmazon a második algoritmus ugyanolyan teljesítményt ért el, mint az eredeti adatokon.

### 5.2.3 LSTM

A szentimentanalízist el szerettem volna végezni kissé hagyományosabb módszereket alkalmazva is, így esett a választásom egy LSTM neurális hálózat kiépítésére, a Pytorch LSTM moduljának használatával. A készített architektúrát a már korábban is említett visszafordítással augmentált adathalmazon és az eredetin teszteltem, továbbá, mindkét állományt a SMOTE módszerrel megvalósuló adatbővítéssel szintén megfuttattam.

Az adatfeldolgozás során újból 80-20 százalékos arányban felosztottam az adatokat tanító- és validációs halmazokra, valamint a DistilBERT tokenizálójával alakítottam numerikussá a szöveges egyedeket. A neurális modell felépítését tekintve a bemeneti réteg egyben a beágyazó rétegnek felel meg, ahol meghatározásra kerül a bemeneti tokenizáló szókincsén felül a beágyazások mérete is, ami 256 értéket kapott. Ezt követi maga az LSTM réteg, amely valójában egy kétrétegű hálózat, melynek fő fókusza az olyan szekvenciális kapcsolatok figyelembevétele, mint például mennyire hasonlítanak az adott értékelési címke egyedei. Ezután szerepel egy teljesen kapcsolt réteg, majd a kimeneti réteget megelőző regularizációs, dropout réteg. A rejtett réteg 256 neuronból áll, bízva a modell minél optimálisabb tanulásában. Költségfüggvényként ezúttal is a keresztentrópia függvényét alkalmaztam, Adam optimizert, valamint a tanulási szakasz paramétereiként 2e-5 sebességet, tizenhatos kötegméretet határoztam meg, öt iterációval. Az eredeti adatokon ezzel 0.43 értékű pontosságot értem el, az F1-score súlyozott átlaga pedig 0.3 értéket eredményezett. A több egyeddel rendelkező halmazon minimális javulás volt tapasztalható a megadott paraméterek mellett, 0.48 pontosság és 0.36 súlyozott F1 átlag értékekkel.

Mivel az épített LSTM architektúra nem bizonyult megfelelő elemzési rendszernek a feladathoz, az eredeti adatállományt SMOTE augmentációs módszerrel bővítve is teszteltem, némiben eltérő paraméterek megadásával. Ezen osztálykiegyenlítés alkalmazásához a feldolgozási lépésben vektorizáltam az adatokat, a korábban már ismertetett eljárással. A SMOTE automatikus mintavétellel szintetikus egyedeket generál, a három legközelebbi szomszéd alapján. Így az eredeti halmaz új eloszlása egységesen 704 darab egyedet foglalt magába, osztályonként. A neurális háló felépítésén szintén módosítottam, beágyazás nélkül dolgozza fel a bemeneti tenzorokat, az LSTM rétege csakugyan egyrétegűre csökkent, ezzel egyszerűsítve a modellt, valamint nincsen regularizációs lépés vagy dropout réteg sem. A rejtett réteg az előző modellel megegyezően szintén 256 neuronból áll, illetve az osztályok egységesen egy egész értékű súllyal kerültek definiálásra. A költségfüggvény és az optimizer megegyezik az első architektúráéval, azonban a tanítás sebességének paraméterét 0.001 értékre növeltem, ahogyan a tanítási iterációk számát is megemeltem tízre. Ezek beállításával és meghatározásával, az eddigi legkedvezőbb eredményt értem el, 0.94 pontossággal és súlyozott F1 átlaggal. A mellékelt ábrán jól látható, hogyan alakult a modell teljesítménye a tíz iteráció során, valamint az is megfigyelhető, mely ponton vált túltanulttá a hálózat. A bal oldali, költségeket jelző diagramon a harmadik tanítási lépés végrehajtásánál látható a legnagyobb egyezés a tanító és teszt adatok veszteségeinél, továbbá, a negyedik iterációt követően nagyon lelassul az értékek csökkenése. A pontosságot reprezentáló ábrán csakugyan a negyedik lépést követően mutat lassulást a teljesítmény növekedése, legmagasabb értékét a hatodik iterációnál érte el. Ezáltal hatékonynak bizonyult a hagyományosabb klasszifikálás eszközeként az LSTM jellegű neurális hálózat, ugyan némi fenntartással, hiszen az elkerülhetetlen kiegyensúlyozatlanságok kezelésében nem remekel.

A képen szöveg, diagram, sor, Diagram látható

Automatikusan generált leírás

10. ábra: SMOTE augmentált halmazon elért modell teljesítmény. Forrás: saját szerkesztés.

### 5.2.4 huBERT

Végül, az elemzésem utolsó BERT variációjaként a kimondottam magyar nyelven finomhangolt és előre feltanított huBERT modelljét implementáltam, a további szövegelemző rendszereim mellett, mely közel 110 millió paraméterrel rendelkezik. Ezen változat teljesítményét összesen három adathalmazon vizsgáltam, az eredeti, a GPT-4 nyelvi modell által generált szinonimákkal kibővített, illetve a visszafordítás módszerével augmentált állományokon. Az adatgyűjtemény előkészítése során a már korábban is alkalmazott módon, 80 és 20 százalékban bontottam fel a halmazt tanító és teszt adatokra, az osztályok arányos eloszlásának biztosításával. Az alap BERT modell beépített tokenizálójának segítségével dolgoztam fel megfelelő formátumba a szöveges adataimat és adtam át Pytorch kompatibilis tenzorok formájában a modell tanításához. A tanítás az eredeti halmaz esetén tizenhatos kötegek, öt tanítási iteráció paraméterek beállításával történt, minden egyéb paraméter a BERT alapértelmezett klasszifikáló modelljének beállításait alkalmazta. Ebbe beletartozik a költségfüggvény is, amely csakugyan a keresztentrópia függvény. A modell teljesítménye a következő ábrákon látható. Az egyértelmű túltanulás állapota a harmadik iteráció után következett be, ez a diagramokon szemléletesen kimutatható. Különösen ahogy a két halmaz költségeinek értékei élesen távolodnak egymástól, a validációs költség emelkedésével és egyúttal romlásával. Ekkor a modell pontossága 67 százalékos volt, a validációs költsége pedig a 0.94 értéket vette fel.

A képen szöveg, diagram, sor, Diagram látható

Automatikusan generált leírás

11. ábra: Eredeti adathalmazon mért modell teljesítmény. Forrás: saját szerkesztés.

Természetesen a teljesítmény középszerűségét az adatok kiegyensúlyozatlansága magyarázza, ami még látványosabban megfigyelhető a precision, recall és F1-score mutatók osztályonkénti alakulásának diagramján. Ezen metrikák súlyozottan átlagolt értékei, rendre 0.59, 0,63 és 0.6, amelyek megfelelően tükrözik a fentebb meghatározott teljesítmény mértékét is.

A képen szöveg, diagram, képernyőkép, Diagram látható

Automatikusan generált leírás

12. ábra: Teljesítménymutatók az osztályok szerint, eredeti adathalmazon. Forrás: saját szerkesztés.

A teljesítmény további elemzéséhez ROC-görbe vizualizációt is alkalmaztam, amely eredményeképp az AUC = 0.76. A kék színű, szaggatott vonal a véletlenszerű osztályozás referenciáját jelenti. A zöld színnel jelölt görbe mivel nem keresztezi a kéket, a modell teljesítménye szerencsére meghaladja a véletlenszerű osztályozás szintjét, ami a 0.5 értéket jelenti. A lépcsőzetes növekedése arra utal, hogy bár képes az osztályok megkülönböztetésére, mégsem éri el a lehető legjobb eredményt. Ez ismételten és végérvényesen visszaigazolja a modell ezen változatának közepesen optimális működését, amelyet mindenképpen érdemesnek bizonyult tovább vizsgálnom az említett két másik adathalmazokon.

A képen szöveg, diagram, sor, Diagram látható

Automatikusan generált leírás

13. ábra: ROC görbe az eredeti halmazon. Forrás: saját szerkesztés.

A második, egyszerűbb módon, egyszerű szinonimákkal kiegészített halmazon a modell nem mutatott szignifikáns fejlődést. Az előzőleg öt darab iteráció esetén tapasztalt túltanulás okán, ezúttal háromra definiáltam a tanítási időszak paraméterét. A pontosság értéke minimálisan romlott, összesen három százalékkal, azonban az AUC értéke ugyanilyen arányban javult. Továbbá, a görbe élesebb emelkedés mellett, simább haladást mutat, ezzel részben jobb osztályozási teljesítményre utalva.

A képen szöveg, sor, diagram, Diagram látható

Automatikusan generált leírás

14. ábra: ROC görbe a GPT-4 augmentált halmazon. Forrás: saját szerkesztés.

A legoptimálisabb teljesítményt a harmadik, egyben legnagyobb volumenű adathalmaz tanításával értem el. A paraméterek közül egyedül az iterációk számán módosítottam, pontosan hat darabra. Ezen feltételek mentén a modell 84 százalékos pontosságot produkált, ezzel párhuzamosan a validációs költség 0.6 értéke körül mozgott. Ezen értékek alakulását az alábbi diagramok szemléltetik. A túltanulás állapota ezesetben már a negyedik iterációnál megfigyelhető, ezt alátámasztja a két költség görbe szétnyílása, illetve a pontosság egyszeri csökkenése is. Bár a pontosság az ötödik iterációnál érte el a legmagasabb értékét, a teszt adathalmaz költsége a tanítási költségnek közel négyszerese volt ezen a ponton.

A képen szöveg, diagram, sor, Diagram látható

Automatikusan generált leírás

15. ábra: Visszafordítással augmentált halmazon elért modell teljesítmény. Forrás: saját szerkesztés.

Az eredeti adathalmazon elért eredményhez képest a precision, recall és F1 mutatók értékei a modell teljesítményének növekedésével arányosan javultak. Az osztályok kiegyensúlyozatlan eloszlása továbbra is megfigyelhető a recall és F1-score esetén, de határozottan kevésbé látványos, mint az eredeti halmaz mutatói esetén. Mivel a három csillagos értékelések többnyire semleges szentimentet hordoznak magukban, így a közeli osztályaival magasabb szintű átfedés okán a modell bizonytalansága növekedett. Ez megmagyarázhatja, miért kapott azon osztály esetében magasabb értéket a recall, mint a precision. Az egy csillagos értékelések esetében a szarkasztikus kommentek félrevezetőek lehetnek a modell számára. Továbbá, az öt csillagosra értékelt vélemények során teljesített a legjobban az algoritmus, ugyanakkor ezen osztály rendelkezett a legnagyobb egyedszámmal.

A képen szöveg, képernyőkép, diagram, Diagram látható

Automatikusan generált leírás

16. ábra: Teljesítménymutatók alakulása a visszafordításos adathalmazon. Forrás: saját szerkesztés.

A ROC-görbe csakugyan a lehető legoptimálisabb ívet érte el, ezzel párhuzamosan az AUC pontosan 0.89 értéket eredményezett. A három görbe közül ezen az ábrán a leghangsúlyosabb a helyes osztályozási képesség, a széles ívű növekedéssel. Valamint, ezen modell esetében éri el a görbe leggyorsabban a legmagasabb valós pozitív arány értéket, alacsony hamis pozitív arány tartása mellett.

A képen szöveg, sor, diagram, Diagram látható

Automatikusan generált leírás

17. ábra: ROC görbe a visszafordítással augmentált adathalmazon. Forrás: saját szerkesztés.

## 5.3 RAG modell

Annak érdekében, hogy az elemzésemmel megválaszolhassam a kutatás legfontosabb kérdéseit és igazoljam, vagy éppen megcáfoljam a hipotézisemet, létrehoztam egy RAG modellt. Ennek megvalósítását a LangChain keretrendszere tette lehetővé, valamint a minél pontosabban értelmezhető eredmények eléréséhez három pipeline variánsát modelleztem. Az egyszerű és hatékony feldolgozás okán a legoptimálisabb huBERT modellt integráltam minden esetben. Ezen modell paramétereit használtam fel a tokenizálási, valamint a beágyazási folyamatok során.

### 5.3.1 Klasszifikáló modell

Az első megközelítés esetében csak egy klasszifikáló modell felépítése volt a célom. Ezt kétféleképpen valósítottam meg, az első példa esetében a korpuszt az említett huBERT modell teszt adathalmazán prediktált értékeinek dataframejével egészítettem ki, valamint olyan pdf formátumú fájlokkal, amelyek a három szolgáltató, manuálisan bemásolt gyakran ismételt kérdéseit és válaszait tartalmazzák. Az adatfeldolgozás szakaszában a pdf dokumentumokat egyszerű szöveggé alakítottam a LangChain saját eszközével, illetve további kisebb, könnyebben kezelhető darabokra bontottam a chunkolás módszerével. Az adattábla minden oszlopát felhasználva, ugyanilyen módon feldolgoztam, majd az így kapott szöveges állományt egybefűztem. Majd, ezt az adathalmazt a FAISS eszközével vektoralapú formában eltároltam. A szöveg klasszifikáló algoritmus paramétereként a bemeneti szöveges adatok méretét legfeljebb 512 tokenre korlátoztam, minden más beállítást a korábban definiált, saját finomhangolt huBERT modellje tartalmaz. Az egyéni sablon definiálását követően az elemzési feladatnak megfelelően formáztam a promptot, valamint az elvárt kimenetet. A lekérdezések során világossá vált, hogy a korpusz nem fog megfelelő információval szolgálni a modellem számára, ugyanis a legtöbb megadott feltételemre, keresésemre szuboptimális válaszokat javasolt. Visszaadott válaszában vegyesen fogalmaz meg irreleváns részleteket a gyakran ismételt kérdések közül, illetve kisebb arányban valóban releváns értékeléseket.

Így módosítottam a modell korpuszát, ezúttal kizárólag az eredeti értékeléséket tartalmazó adathalmazt felhasználva, kizárólag az értékelés, a szöveges vélemény és az applikáció típusának oszlopaival. Ugyanazon keresési kifejezéssel az eredmény koherensebb választ adott, valamint a hármas kategóriába sorolta, amely semlegessége okán akár helyesen is értelmezhető. A panaszos véleményeket kérdő lekérdezések esetén pedig minden alkalommal helyesen, a legnegatívabb értékelés osztályába sorolja.

### 5.3.2 Hibrid modell

A kielégítően működő RAG modellemet ki szerettem volna egészíteni egy generatív eljárással, bízva a még részletesebb válaszok eredményében. Ennek megvalósításához a második klasszifikáló modellt egy hibrid pipelineban implementáltam. Az előzőleg inicializált vektor adatbázist használtam retrieverként, a generatív modellként és annak tokenizálójaként pedig a többnyelvű Google mT5 alap verzióját alkalmaztam, amely körülbelül 580 millió paraméterrel rendelkezik. A hibrid architektúrában először a dokumentumok szűrése valósul meg, biztosítva a dokumentumok egyedi előfordulását, amit a klasszifikáló modell promptja és eredménye követ. Végül a generatív válaszképzési rész következik, egy újabb, összegzésre szorgalmazó prompt meghatározásával és az eredmények paramétereinek megadásával. Ebbe beletartozik a válasz maximális hossza, ami 100 token, öt válaszopció generálása, illetve a válasz kreativitásának paramétere, a temperature 0.5 értékkel, ami kiegyensúlyozottabb válaszokat jelent. Végül megadtam a válaszok elvárt megjelenítési formátumát, hogy külön szerepeljen a dokumentum visszaadott releváns tartalma, az osztályozás eredménye és a generált válasz. A hibrid megközelítést több lekérdezéssel teszteltem, eredményeinek elemzése alapján megállapíthattam, hogy a klasszifikálás megfelelően megy végbe, koherens és releváns értékelések visszaadásával válaszol a keresett kifejezésre vagy kérdésre. A generált válasz esetében azt tapasztam, hogy volna hely a javulásnak, azonban bizonyos mértékig releváns összegzést készít a visszaadott értékelések alapján.

# 6. Összefoglalás

A kutatásom céljaként megfogalmazott, közösségi autómegosztó szolgáltatók összehasonlításához több, mély tanuló, szövegelemző modellt használtam fel. Ezek implementálása olykor kihívásnak bizonyult, azonban úgy gondolom, az elemzésem összességében sikeresnek tekinthető. A bevezetésben felvetett kérdéseket az alábbiakban megválaszolom, részletesen ismertetem a hipotézisvizsgálat eredményét.

## 6.1 Értékelés és következtetések

Az elemzés korai szakaszában világossá vált, hogy az emberek általánosságban kijelenthető, szélsőségek felé hajlása miatt nem lesz egyszerű egy optimális klasszifikáló modell felállítása. A hihetetlenül alulreprezentált osztáályok feljavítása nem kevés időt és erőforrást vett igénybe, a különböző augmentációs eszközök kipróbálásával, azok optimalizálásával és futtatásával. Az adatok minőségével és mennyiségével kapcsolatosan az is kihívást jelentett, hogy a szolgáltatók kevesebb, mint tíz éve vannak jelen Magyarországon, hiszen a GreenGo 2016-ban, a MOL Limo 2018-ban, a wigo pedig, eredeti formájában DriveNow néven 2019-ben indult el. A földrajzi specifikusság által, hogy csak a fővárosban és annak közvetlen vonzáskörzetében érhető el a carsharing szolgáltatás elsősorban, tovább redukálta a potenciális vélemények számát. Mivel a modellek minél több, lehetőleg több ezer nagyságú, közel egyenletes eloszlású állományokon remekelnek igazán, ez hátrányosan hatott az elemzés eredményeire. Továbbá, szintén a földrajzilag kis területen való vizsgálat okán, a magyar nyelvű szöveges adatok tanításán alkalmas nyelvi modellek köre meglehetősen szűk. Mint az ki is derült, a transzformer alapú LLM-ek esetén a legjobb teljesítményt a kimondottan magyar nyelvre finomhangolt huBERT modelljével értem el, az általánosan többnyelvű BERT variánsok kevésbé voltak képesek elsajátítani a magyar nyelv sajátosságait és értékelésekben előforduló mintáit. Ugyanezen korlát az adatbővítő módszerek alkalmazásánál is jelen volt, sok augmentációs eszköz csak angol nyelvű szöveges adatokon értelmezhető.

A kutatás kérdéseire térve, valóban alkalmazható olyan nyelvi modell a felhasználói értékeléseken, amely legalább 80 százalékos pontossággal képes új adatokat osztályozni. Ez a modell az említett huBERT, amelynek accuracy értéke 84 százaléknak felel meg, amelyet a visszafordítás módszerével, duplájára növelt adathalmazon tanítva ért el. Az eredeti adathalmazon vizsgálva, legjobb esetben 66 százalékkal egyezett meg az osztályozási teljesítménye. Ehhez szorosan kapcsolódik az a kijelentés, hogy valóban jobban teljesít egy magyar nyelven optimalizált modell egy többnyelvű változatával szemben. Ezt alátámasztja az XLM-RoBERTa modell legjobb elért teljesítménye, amely az eredeti adathalmazon, általános formájában alig 40 százalékos pontosságot ért el, míg ugyanezen modell magyar nyelven finomhangolt változatának teljesítménye 54 százalékra növekedett. Amennyiben a huBERT teljesítményéhez viszonyítjuk, mindkét esetben szembetűnő az utóbbi modell fölénye.

A további kérdések megválaszolásához a téma modellezés eredményeit figyelembe véve, az egyes autómegosztókra adott értékelések között nem figyelhető meg különösen nagy eltérés. Ezt az egyszerű vizualizációk is alátámaszthatják, mint a szófelhők ábrája, amelyen a három szolgáltató értékelései között hasonló gyakorisággal szerepelnek ugyanazon szavak, mint a jó és szuper, illetve a regisztrációval és használattal szorosan összefüggő kifejezések. A t-SNE alapú grafikonon is megfigyelhető, hogy bár a vélemények osztályonként eltérő tartalommal rendelkeznek, mégsem tartozik különösebben specifikus vélemény egyik szolgáltatóhoz sem. Az LDA alapján pedig ez végképp beigazolódik, hiszen a kördiagramokon jól látható a négy, azonosított témának nagyon hasonló eloszlása az applikációknál. Az egyes appok általános értékeléseinek jellemzéséről szóló kérdést a RAG modell segítségével szeretném megválaszolni. Ezalapján, a „GreenGo vélemény” lekérdezésre határozott, 98 százalékos biztossággal ötös értékelésbe sorolt választ adott, valóban pozitív tartalommal, egyes felhasználók szerint a legjobb szolgáltatásként hivatkozva. A MOL Limora ugyanezen lekérdezés esetén 95 százalékos biztossággal sorolta a legpozitívabb szentimentbe, általános pozitív megjegyzésekkel, ajánlással. A wigo esetén pedig 45 százalékos értékkel szintén öt csillagosnak kategorizálta, egy olyan értékelést is kiemelve, ahol a valóban öt csillagos értékelés ellenére az autó tisztaságáról panaszkodott.

Gyakran előforduló panaszok azonosításához ismételten egy lekérdezést végeztem a modellel, válasza alapján a wigo ügyfélszolgálatával, illetve az applikáció technikai megoldásaival kapcsolatban elégedetlenek az ügyfelek. Kiemelkedő elégedettségre azonban mindhárom szolgáltatóhoz tartozik értékelés, a MOL Limo esetén a könnyű használatot és az autókat dicsérik, a wigo értékelői a kedvező árazást és a lefedettségi területet fogadják örömmel, a GreenGo felhasználói pedig hasznosnak tartják és összességében vannak megelégedve a szolgáltatással.

Hipotézisemként azt fogalmaztam meg, hogy létezik egy olyan autómegosztó szolgáltató, amely összességében kiemelkedik a másik két céggel szemben. Az elemzés teljességét tekintve, ezen hipotézist elutasítom, ugyanis mind az általános statisztikai értékelés, a téma modellezés, a szentiment analízis és RAG modell eredményei alapján, erős hasonlóság figyelhető meg mindhárom szolgáltató felhasználói véleményei között.

## 6.2 Javaslatok

Bár kielégítő teljesítményt értek el az adatokon egyes modellek, természetesen érdemes lenne további fejlesztések implementálása, a pontosabb eredmények elérése érdekében. Ezen célt szolgáló javaslatokat a következőkben részletezem.

Mint az fentebb említésre került, az alkalmazásokra adott értékelések mennyisége önmagukban egy stabil modell tanításához nem elegendő. Továbbá, a wigo, elődjeivel együtt számolva összesen három darab alkalmazással rendelkezik, melyek közül kettő nem exkluzívan Magyarországon volt elérhető, így azok korábbi adatait elenyésző mennyiségben tudtam hasznosítani. Azonban, a szolgáltatók teljeskörűbb összehasonlításához remek alapot szolgált volna a Google Térképen írt vélemények összessége, ami az értékelések számát is növelte volna, várhatóan az eloszlás javításával együtt. Sajnos ezen adatok automatizált összegyűjtése a Google biztonsági szabályozásai miatt nem egy egyszerűen kivitelezhető feladat.

Továbbá, érdekes eredményeket hozhat a további modell kombinációkon folytatott tanítások teljesítménye, amelynek az idő és rendelkezésre álló erőforrások szabnak korlátot. Ehhez hasonlóan, az alkalmazott modellek további tanítási hiperparamétereinek optimalizálása szintén eredményezhet kielégítő működést, akár az általánosan többnyelvű finomhangolt variációk esetén is. Egy hatékony optimalizációs eljárással jelentősen csökkenne az idő és erőforrás igénye ezen megoldási javaslatnak.

A RAG modell tekintetében, érdemes lehet más beágyazó, tokenizáló, retriever modellekkel felépíteni a pipelinet és összehasonlítani azok működését, teljesítményét. Illetve, másik keretrendszerben történő felépítés működése csakugyan eredményezhet optimálisabb válaszokat. Sajnos, a hibrid RAG megoldásában alkalmazott generatív modell nem bizonyult a legmegfelelőbb választásnak, érdemes lenne egyéb modellekkel helyettesíteni, lehetőleg magyar nyelven optimalizálttal. Természetesen, a megadott promptolt utasítások alternatíváival is érdemes volna foglalkozni, elemezni, mennyiben befolyásolja az optimális működést. Továbbá, a korpusz lehetőségei is vizsgálandók, akár az osztályozó aspektusától elragaszkodva, egy teljesen generatív megoldással.

# 7. Irodalomjegyzék

Bengio, Y., Goodfellow, I., & Courville, A. (2015). *Deep Learning*.

Blagus, R., & Lusa, L. (2013). *SMOTE for high-dimensional class-imbalanced data* (Vol. 14). http://www.biomedcentral.com/1471-2105/14/106

Cheng, J., Dong, L., & Lapata, M. (2016). *Long Short-Term Memory-Networks for Machine Reading*.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., & Jégou, H. (2024). *The Faiss library*.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). *Retrieval-Augmented Generation for Large Language Models: A Survey*. https://arxiv.org/abs/2312.10997v5

Goyal, P., Pandey, S., & Jain, K. (2018). Deep Learning for Natural Language Processing. In *Deep Learning for Natural Language Processing*. Apress. https://doi.org/10.1007/978-1-4842-3685-7

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, *78*(11), 15169–15211. https://doi.org/10.1007/s11042-018-6894-4

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

L. Ashok Kumar, & D. Karthika Renuka. (2023). *Deep Learning Approach for Natural Language Processing, Speech, and Computer Vision*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459–9474). Curran Associates, Inc. https://proceedings.neurips.cc/paper\_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

Lewis Tunstall, Leandro von Werra, & Thomas Wolf. (2022). *Natural Language Processing with Transformers*. O’Reilly Media Inc.

Mejova, Y. (2009). *Sentiment Analysis: An Overview Comprehensive Exam Paper*. http://www.pewinternet.org/Reports/2009/15-The-Internet-and-Civic-Engagement.aspx

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2023). *A Comprehensive Overview of Large Language Models*.

Pointer, Ian. (2019). *Programming PyTorch for deep learning : creating and deploying deep learning applications*. O’Reilly Media.

Steven W. Smith. (1999). *The Scientist and Engineer’s Guide to Digital Signal Processing*. California Technical Publishing.

Topsakal, O., & Akinci, T. C. (2023). Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. *International Conference on Applied Engineering and Natural Sciences*, *1*(1), 1050–1056. https://doi.org/10.59287/icaens.1127

Wehnert, S., Sudhi, V., Dureja, S., Kutty, L., Shahania, S., & De Luca, E. W. (2021). Legal norm retrieval with variations of the bert model combined with TF-IDF vectorization. *Proceedings of the 18th International Conference on Artificial Intelligence and Law, ICAIL 2021*, 285–294. https://doi.org/10.1145/3462757.3466104

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. In *Neural Computation* (Vol. 31, Issue 7, pp. 1235–1270). MIT Press Journals. https://doi.org/10.1162/neco\_a\_01199

Ábrák, képek, táblázatok jegyzéke

Mellékletek

[A kutatás Git tárhelye](https://github.com/mlaura227/Carsharing)