**Tartalomjegyzék**

[1 Bevezető 2](#_Toc166336067)

[2 Irodalomkutatás 3](#_Toc166336068)

[2.1 Alapfogalmak és definíciók 3](#_Toc166336069)

[2.2 Hagyományos túraidő-becslési módszerek 3](#_Toc166336070)

[2.2.1 Naismith szabálya 3](#_Toc166336071)

[2.2.2 Tobler függvény 3](#_Toc166336072)

[2.2.3 Book Time formula 4](#_Toc166336073)

[2.2.4 Munter módszer 4](#_Toc166336074)

[2.3 Mesterséges intelligencia alkalmazása a túraidő becslésben 5](#_Toc166336075)

[2.3.1 Mesterséges intelligenciát alkalmazó túra applikációk 5](#_Toc166336076)

[2.3.2 Mesterséges intelligencia algoritmusok a menetidő meghatározásához 5](#_Toc166336077)

[2.3.3 Random Forest algoritmus 6](#_Toc166336078)

[2.3.4 XGBoost algoritmus 7](#_Toc166336079)

[2.4 Adatforrások és adatgyűjtés 7](#_Toc166336080)

[2.5 Összefoglalás az irodalomkutatásból 7](#_Toc166336081)

[3 Rendszerterv 8](#_Toc166336082)

[3.1 Bevezetés a rendszertervbe 8](#_Toc166336083)

[3.2 Alkalmazás architektúra 8](#_Toc166336084)

[3.2.1 Frontend technológiák 8](#_Toc166336085)

[3.2.2 Backend technológiák 8](#_Toc166336086)

[3.2.3 Framework és könyvtárak választása 8](#_Toc166336087)

[3.2.4 API-k és integrációk 8](#_Toc166336088)

[3.2.5 Cloud szolgáltatások és hosting 8](#_Toc166336089)

[3.3 Felhasználók kezelése 8](#_Toc166336090)

[3.4 Fejlesztési környezet 8](#_Toc166336091)

[3.5 Biztonság és teljesítmény 8](#_Toc166336092)

[3.6 Tesztelési stratégia 8](#_Toc166336093)

[3.7 Implementációs terv 8](#_Toc166336094)

[3.8 Dokumentáció és támogatás 9](#_Toc166336095)

[4 Bibliography 10](#_Toc166336096)

# Bevezető

A túrázás a szabadidős tevékenységek terén kiemelkedő népszerűségnek örvend. Azonban a túrázással kapcsolatos egyik legnagyobb kihívás a tervezés: hogyan becsüljük meg az útvonalakon való haladás idejét pontosan? Ez például a tömegközlekedéssel való hazajutás tervezése szempontjából is kritikus lehet. Az időtartam következtetésének hagyományos módszerei – mint például a Naismith-szabály (2.2) vagy a Book Time formula – gyakran nem veszik figyelembe elegendően a túrázó egyéni képességeit, illetve az útvonal specifikus jellemzőit, mint a szintkülönbség és annak sorrendje, valamint azt sem, hogy egyénileg, vagy csoportban halad az illető.

Jelen szakdolgozat célja egy innovatív megoldás bemutatása, amely mesterséges intelligencia (MI) modell segítségével következtetést ad a turista útvonalak teljesítési idejére. Ez az alkalmazás a felhasználó fizikai kondícióját és az útvonal jellegzetességeit egyaránt figyelembe veszi, lehetővé téve ezzel egy személyre szabottabb és pontosabb időbecslést.

A modern technológia alkalmazása a túra tervezésében nem csak hogy pontosabbá teszi az időtartamok becslését, hanem a túrázók biztonságát is növelheti, mivel reálisabb képet kapnak az útvonal nehézségéről és időigényéről. Emellett, az adat vezérelt megközelítés új perspektívákat nyit meg a túraútvonalak optimalizálásában és az egészségügyi hatások monitorozásában is. A szakdolgozat az AI technológiák és az aktív életmód ötvözésének úttörő munkáját mutatja be, lépést tartva a technológiai fejlődéssel és az egyre növekvő felhasználói igényekkel. Ezáltal nem csak a jelenlegi, de a jövő túrázói számára is értékes eszközt kínál, megerősítve az MI létjogosultságát és hasznosságát a mindennapi élet számos területén.

# Irodalomkutatás

## ****Alapfogalmak és definíciók****

/ Túrázás, túraidő, mesterséges intelligencia alapjai

## ****Hagyományos túraidő-becslési módszerek****

A túrázás menetidejének kiszámítására számos módszer létezik, amelyeket a világ különböző részein alkalmaznak. Ezek a módszerek különböző tényezőket vesznek figyelembe, mint például a távolságot, az emelkedőket, és a túrázók fizikai állapotát. Az alábbiakban bemutatom a leggyakoribb és legismertebb módszereket:

### Naismith szabálya

A Naismith-szabályt arra használják, hogy becsüljék a gyaloglási időt egyenetlen terepen. A szabályt William Naismith skót hegymászó dolgozta ki 1892-ben, ami 5 km/óra sebességet feltételez sík terepen, minden 300 méter emelkedésre plusz fél órát hozzáadva. A szabályt később finomították különböző empirikus tesztek alapján. Langmuir 1984-ben javított a szabályon, 10 percet levonva minden 300 méteres lejtőért 5 és 12 fok között, és hozzáadva 10 percet minden 300 méteres lejtőért, ha az nagyobb mint 12 fok. [1] [2]

(egyenlet 1)

A Naismith szabály azonban nem vesz figyelembe több olyan tényezőt, mint a terep nehézsége, a túrázó fizikai állapota vagy az időjárási körülmények, amik jelentősen befolyásolhatják a túra időtartamát. Ezért gyakran szükséges módosításokat alkalmazni vagy modernizált változatokat használni, amelyek jobban megfelelnek a valós körülményeknek.

Ezeket a korlátokat felismerve különböző túrázási szakértők és szervezetek fejlesztettek ki kiegészítő szabályokat vagy teljesen új modelleket, amelyek részletesebben veszik figyelembe a különféle változókat, mint például a Langmuir-féle módosítás, amely az emelkedés mértékével arányosan növeli az időtartamot.

### Tobler függvény

A Tobler gyaloglási függvény, amelyet Waldo Tobler dolgozott ki, exponenciális modellt használ a gyaloglási sebesség becslésére a terep lejtése alapján. A függvény a legnagyobb gyaloglási sebességet enyhe lejtőnél (-5%-os gradiens) jósolja meg, és ezeket a paramétereket használja: gyaloglási sebesség v, maximális sebesség α, és együtthatók β1​ és β2​, amelyek a lejtő változása függvényében a sebesség változásait számolják.   
(1. ábra) Ez a modell különösen hatékony a változatos topográfiai körülmények közötti mozgás modellezésére. [2] [3]

A black background with a black square

Description automatically generated with medium confidence

1. ábra: Tobler függvénye. [4]

### Book Time formula

A "Book Time" képlet egy egyszerűsített módszer a túrák teljesítéséhez szükséges idő becslésére, amelyet gyakran használnak túratervezőkben. Általában minden megtett mérföldre 30 percet, és minden 1000 láb magasságnövekedésre további 30 percet számol. Ez a képlet hasonló a Naismith-szabályhoz, de valamivel kisebb tempót alkalmaz, ami különösen hasznos lehet hegyvidéki terepen, ahol a meredekebb és nehezebb utak miatt lassabb haladás várható. [2] [5]

Továbbá, a módszer elismeri, hogy a környezeti, logisztikai és személyes tényezők jelentősen befolyásolhatják a túrák időtartamát, ami azt jelenti, hogy a valós idő eltérhet a terepen.

### Munter módszer

A Munter módszer, amelyet Werner Munter svájci túravezető fejlesztett ki, egy gyakran használt eszköz a hegyekben történő utazási idő becslésére, figyelembe véve az utazott távolságot, a szintkülönbséget és az utazási módot. Ez a módszer különösen hasznos a hegyi túrázáshoz, síeléshez és hegymászáshoz. [6]

Az időt úgy számítja, hogy egységeket vesz figyelembe, ahol minden megtett kilométer vagy minden 100 méter szintemelkedés egy-egy egységet jelent. Az összes egységet egy együtthatóval osztják el, amely változik, attól függően, hogy milyen tevékenységet végeznek (pl. gyaloglás, síelés, hegymászás) és hogy fölfelé, vagy lefelé haladnak.

A Munter módszer egy egyszerű képletet alkalmaz, ahol az időt úgy számolja ki, hogy hozzáadja az utazott távolságot a szintemelkedéshez, osztva százzal, majd ezt az összeget elosztja egy meghatározott ráta értékkel. Lásd: ((egyenlet 2)

(egyenlet 2)

Például: A gyalogos felfelé haladása esetén a *Ráta*-t általában 4-es értéken állapítják meg, sík vagy lefelé haladásnál gyalog 6-os értéket használnak, síelésnél lefelé pedig 10-es értéket.

A tervezéshez a GuidePace [6] - mobilalkalmazás is használja a Munter módszert más rendszerekkel együtt, hogy segítse a felhasználókat az időbecslések tervezésében és kiszámításában különböző tereptípusokra. Ez az alkalmazás különösen az útikalauzok és komoly „backcountry” rajongók számára készült, akiknek pontos időbecsléseket kell készíteniük útjaikhoz.

## ****Mesterséges intelligencia alkalmazása a túraidő becslésben****

### Mesterséges intelligenciát alkalmazó túra applikációk

Az AllTrails kiemelkedik átfogó funkcióival, amelyek megfelelnek a hétvégi kirándulóknak és a tapasztalt hegymászóknak is. Mesterséges intelligencia felhasználásával az AllTrails valós idejű adatok és felhasználói visszajelzések alapján állítja be a túrák időbecsléseit, biztosítva, hogy a javaslatok mindig az aktuális útviszonyokhoz és az egyéni tempóhoz igazodjanak. Az alkalmazás képessége, hogy integrálja a személyes fittségi szintet és a túrázási preferenciákat az útvonaltervező algoritmusaiba, lehetővé teszi személyre szabott útiterv kialakítását, amely kihívást jelent, de mégis megvalósítható. További információk az AllTrails weboldalán találhatók [7].

A Komoot részletes útvonaltervezést kínál, fordulóról fordulóra történő hang alapú navigációval, amely különösen hasznos az ismeretlen helyszíneken. Az alkalmazás mesterséges intelligencia alkalmazásával javítja az útvonaltervezést, figyelembe véve olyan tényezőket, mint a terep nehézsége, az időjárási viszonyok és a túrázó korábbi kirándulásai, annak érdekében, hogy a legalkalmasabb utakat javasolja. Ez a technológia biztosítja, hogy a túrázók újra tervezést valósíthassanak meg, offline állapotban is, így megbízható társ a vadonban történő kalandokhoz. További információk a Komoot weboldalán érhetők el [8].

A PeakVisor egy másik innovatív alkalmazás, amely mesterséges intelligenciát ötvöz kiterjesztett valósággal, hogy valós időben tudja azonosítani a hegyeket a túrázók számára. A kamera vizuális bemenetének elemzésével a PeakVisor képes megjeleníteni az éppen látható hegyek nevét és információit, javítva ezzel a túrázás informatív részét. Az AI vezérelte felület az információkat a felhasználó helyzetéhez és nézőpontjához igazítja, így személyre szabott és interaktív kültéri élményt nyújt. További információk a weboldalon találhatók [9].

### Mesterséges intelligencia algoritmusok a menetidő meghatározásához

A túraútvonalak időtartamának becslésében kiemelkedő szerepet tudnak játszani a mesterséges intelligencia algoritmusai, különösen az összetett tanítási módszerek, beleértve a Random Forest (2.3.3) és az XGBoost-ot. Ezek a technikák azért hatékonyak, mert több gépi tanulási modell következtetéseit összesítik, növelve ezzel a pontosságot és megbízhatóságot. Ezenkívül neurális hálózatokat is alkalmazhatunk mély tanulási képességeik miatt, amelyek hatékonyan kezelik a bonyolult adatgyűjteményeket és az adatváltozásokat, amelyek a különböző terepviszonyokkal és körülményekkel járnak.

### Random Forest algoritmus

A Random Forest algoritmus egy összetett gépi tanulási technika, amelyet osztályozási és regressziós feladatokra egyaránt használnak. Ez a módszer a döntési fák koncepciójára épül, amelyek egyszerű modellek, melyek elágazási módszereket alkalmaznak a döntések meghozatalához a bemeneti jellemzők alapján. A Random Forest javítja a döntési fa teljesítményét úgy, hogy egy "erdőt" hoz létre fákból, melyeket a képzési adatok és jellemzők véletlenszerűen választott részhalmazai alapján generálnak, majd átlagolják előrejelzéseiket a pontosság növelése és a túltanulás ellenőrzése érdekében [10].

Lényegében a Random Forest az egyes, egymással nem szorosan összefüggő döntési fák előnyét használja fel egy általánosíthatóbb és robusztusabb modell létrehozásához. Az erdő minden egyes fája „szavaz” egy adott előrejelzésre, és a végső kimenetet a többségi szavazat határozza meg osztályozási esetben, vagy az átlag regressziós esetben. Ez a módszer hatékonyan csökkenti a túltanulás kockázatát, amely az egyes döntési fákra jellemző, mivel több előrejelzés átlagolásával kiküszöböli a torzításokat (2. ábra).

Az algoritmus hatékonysága abból adódik, hogy képes nagy adathalmazokat kezelni magas dimenzionalitással, és nem érzékeny a lényegtelen jellemzők bevonására, így sokoldalú és hatékony eszköz a gépi tanulási módszerek között.A diagram of a group of people

Description automatically generated

2. ábra: A Random Forest algoritmus működése.

### XGBoost algoritmus

Az XGBoost [11] egy rendkívül hatékony továbbfejlesztett változata a gradiens fa alapú algoritmusoknak, amely széles körben használt a gépi tanulási feladatoknál, különösen nagy vagy ritka adathalmazok esetén. Ez az algoritmus több fejlett funkcióval finomítja az alap folyamatokat:

Szabályozott Tanulási Cél: Az XGBoost szabályozási kifejezéseket vezet be a célfüggvénybe a túltanulás kontrollálása érdekében, ami gyakori probléma a standard erősítési eljárásoknál. Ez a szabályozás nemcsak a modell teljesítményének javítására szolgál, hanem egyszerűsíti a kapott modellt is, gyorsabbá és hatékonyabbá téve azt.

Gradiens Fa Erősítés: Az XGBoost gradiens erősítési keretrendszereket használ az egymást követő fák felépítésére, ahol minden új fa korrigálja az előzőleg betanított fák által elkövetett hibákat. A modell additív módon képződik, vagyis az új fák egyesével adódnak hozzá, és a modellben már meglévő fák nem módosulnak.

Szórványosság tudatosság: Az algoritmus úgy van kialakítva, hogy kezelje a valós adatokban előforduló különböző adatritkasági mintákat. Ezáltal olyan fastruktúrát épít, amely figyelembe veszi a hiányzó adatokat, a nullás bejegyzéseket és az úgynevezett „one-hot encoded” [12] jellemzőket, optimalizálva az osztópontok megtalálását és alkalmazását a tanítás során.

Rendszertervezés: Az XGBoost rugalmas és erőteljes rendszertervezéssel büszkélkedhet, amely támogatja mind a sűrű, mind a ritka adatokat. A rendszer egyetlen gépen futtatható, vagy több gépen is skálázható egy megosztott környezetben, kihasználva a nagyon nagy adathalmazok hatékony kezelésének képességét. Összességében az XGBoost skálázhatósága, teljesítménye és a különböző adattípusok kezelésének képessége miatt népszerű választás a különböző területeken dolgozó adattudománnyal foglalkozók körében, amikor összetett prediktív modellezési kihívásokkal kell megbirkózniuk.

### Recurrent Neural Networks (RNN)

Az időszaki neurális hálózatok (RNN-ek) olyan neurális hálózatok osztálya, amelyek sorozatos adatok feldolgozására lettek tervezve. [13] Különösen hatékonyak, mivel fenntartanak egyfajta memóriát azáltal, hogy kimenetüket bemenetként használják a következő lépéshez, ami létfontosságú azoknál a feladatoknál, ahol a sorozat korábbi részeiből származó kontextus releváns a későbbi részek feldolgozásához.

### Long Short-Term Memory (LSTM)

A hosszú rövid távú memóriahálózatok, (LSTM-ek) egy fejlettebb típusú RNN (2.3.5), amelyet az elhalványuló gradiens probléma kezelésére fejlesztettek ki, ami a standard RNN-ekben előfordulhat. Ez a probléma megnehezíti az RNN számára, hogy megtanuljon és megőrizzen mintákat hosszú intervallumokkal rendelkező adatokban. Az LSTM-ek ezt úgy oldják meg, hogy beépítenek úgynevezett kapukat: bemeneti, elfelejtési és kimeneti kapukat. Ezek a kapuk határozzák meg, hogy milyen információkat kell megtartani vagy eldobni a hálózaton keresztül, lehetővé téve az LSTM-ek számára, hogy megtanulják az adatok hosszú távú függőségeit [13].

A diagram of a machine

Description automatically generated

3. ábra:Az RNN és az LSTM architektúrák működése. [14]

Mindkét architektúra értékes a sorozatos adatok feldolgozásának és az idősoros adatokon alapuló előrejelzések készítésének képessége miatt, így alkalmasak olyan alkalmazásokra, mint a beszédfelismerés, nyelvmodellezés és még összetett feladatok, mint például a túraútvonalak időtartamának előrejelzése, ahol az időbeli minták kulcsfontosságúak.

## ****Adatforrások és adatgyűjtés****

//Területi és időjárás adatok

//Nyilvános adatbázisok és API-k használata

## ****Összefoglalás az irodalomkutatásból****

//Főbb tanulságok és hiányosságok

//Kutatási rés kérdések

# Rendszerterv

## Bevezetés a rendszertervbe

// A fejlesztendő alkalmazás célja és funkcionális követelményei

// Várható felhasználói bázis és esetleges terhelési elvárások

## Alkalmazás architektúra

### Frontend technológiák

// Lehetőségek: React, Angular, Vue.js

// Felhasználói interfész dizájn és reszponzivitás

### Backend technológiák

// Szerveroldali programozási nyelvek: Node.js, Python (Django, Flask), Ruby on Rails

// Adatbázis-kezelés: SQL (PostgreSQL, MySQL), NoSQL (MongoDB)

### Framework és könyvtárak választása

// Keretrendszerek és fejlesztési környezetek alkalmazása

### API-k és integrációk

// Külső szolgáltatásokkal való integráció (pl. időjárás API-k, térképi szolgáltatások)

### Cloud szolgáltatások és hosting

// Platform: AWS, Google Cloud, Azure

// Szolgáltatások: számítási kapacitás, adattárolás, hitelesítés

## Felhasználók kezelése

// hitelesítés és autorizáció

// Felhasználói fiókok kezelése, jogosultsági szintek

// Adatvédelmi elvárások és GDPR megfelelés

## Fejlesztési környezet

// Kódverzió-kezelés (Git)

// Fejlesztési és tesztelési szerverek konfigurálása

// CI/CD folyamatok implementálása (pl. Jenkins, Travis CI)

## Biztonság és teljesítmény

// Biztonsági protokollok és gyakorlatok (HTTPS, JWT)

// Teljesítményoptimalizálás és skálázási stratégiák

## Tesztelési stratégia

// Automatizált tesztek (egység-, integrációs-, terhelési tesztek)

// Manuális felhasználói tesztek tervezése

## Implementációs terv

// Fázisok és mérföldkövek a fejlesztési ciklusban

// Szerepkörök és felelősségek definiálása

## Dokumentáció és támogatás

// Technikai dokumentáció és felhasználói kézikönyvek

// Karbantartási és támogatási tervek

# Bibliography

|  |  |
| --- | --- |
| [1] | Y. Mingyu, F. van Coillie, M. Liu, R. de Wulf, L. Hens and X. Ou, “BioOne Digital Library,” *A GIS Approach to Estimating Tourists' Off-road Use in a Mountainous Protected Area of Northwest Yunnan, China,* 2014. |
| [2] | "Adventure Nerds," [Online]. Available: https://adventurenerds.com/article/hiking-time-calculator-and-how-to-estimate-hiking-time/. |
| [3] | C. D. Higgins, "Findings," 2021. [Online]. Available: https://findingspress.org/article/28107-hiking-with-tobler-tracking-movement-and-calibrating-a-cost-function-for-personalized-3d-accessibility. |
| [4] | J. van Etten, "ResearchGate," [Online]. Available: https://www.researchgate.net/figure/Toblers-hiking-function\_fig3\_305656946. |
| [5] | "The Hiking Adventure - MASTERING HIKING SPEED: HOW TO CALCULATE YOUR PACE AND TIME," [Online]. Available: https://thehikingadventure.com/hiking-speed-calculator. |
| [6] | C. Burk, "Applied Alpinism," [Online]. Available: https://www.appliedalpinism.com/guide-pace. |
| [7] | "All Trails," [Online]. Available: https://www.alltrails.com. |
| [8] | "Komoot," [Online]. Available: https://www.komoot.com. |
| [9] | "PeakVisor," [Online]. Available: https://peakvisor.com. |
| [10] | G. Louppe, "Cornell University arXiv," 2015. [Online]. Available: https://arxiv.org/abs/1407.7502. |
| [11] | T. Chen and C. Guestrin, "Cornell University - arXiv," 2016. [Online]. Available: https://arxiv.org/abs/1603.02754. |
| [12] | "Codefinity - One-hot encoded," [Online]. Available: https://codefinity.com/courses/v2/a65bbc96-309e-4df9-a790-a1eb8c815a1c/1fce4aa9-710f-4bc9-ad66-16b4b2d30929/a6d33d0d-3057-4a2f-b8df-4ecd00ffd598?utm\_source=google&utm\_medium=cpc&utm\_campaign=21193856569&utm\_content=&utm\_term=&gad\_source=1&gclid=CjwKCAjwrv. |