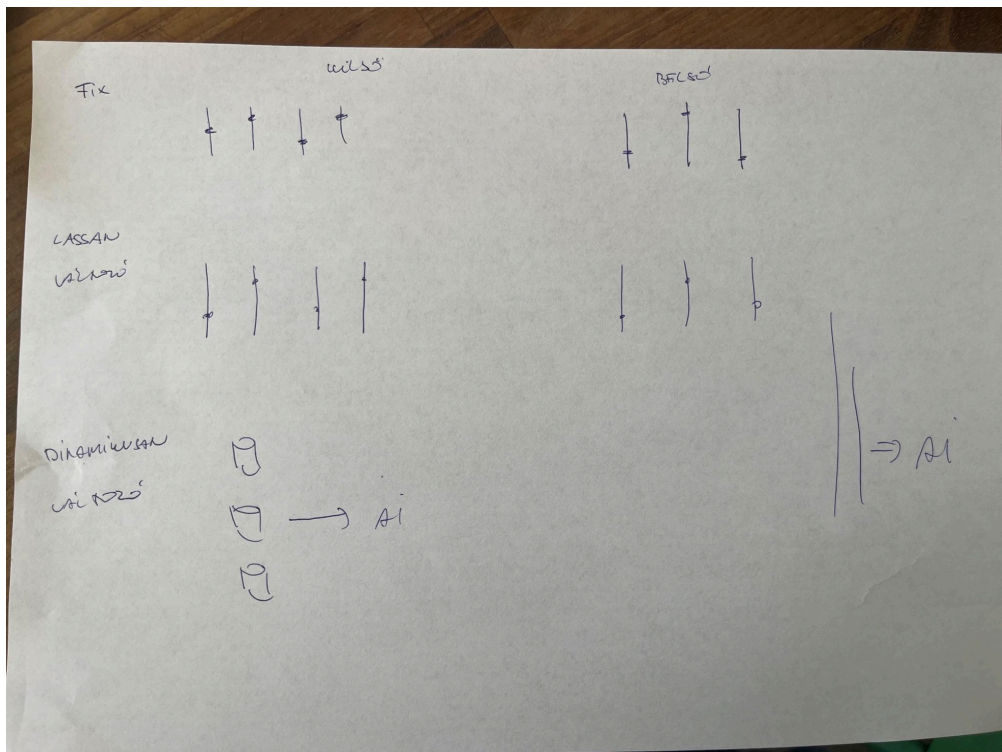
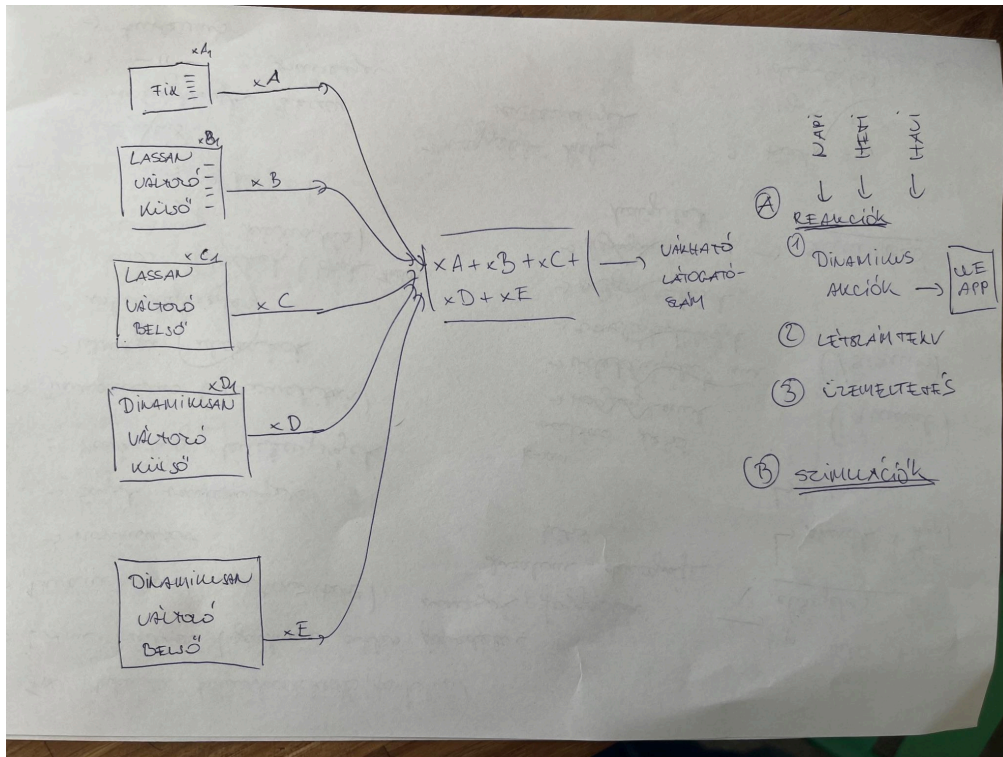
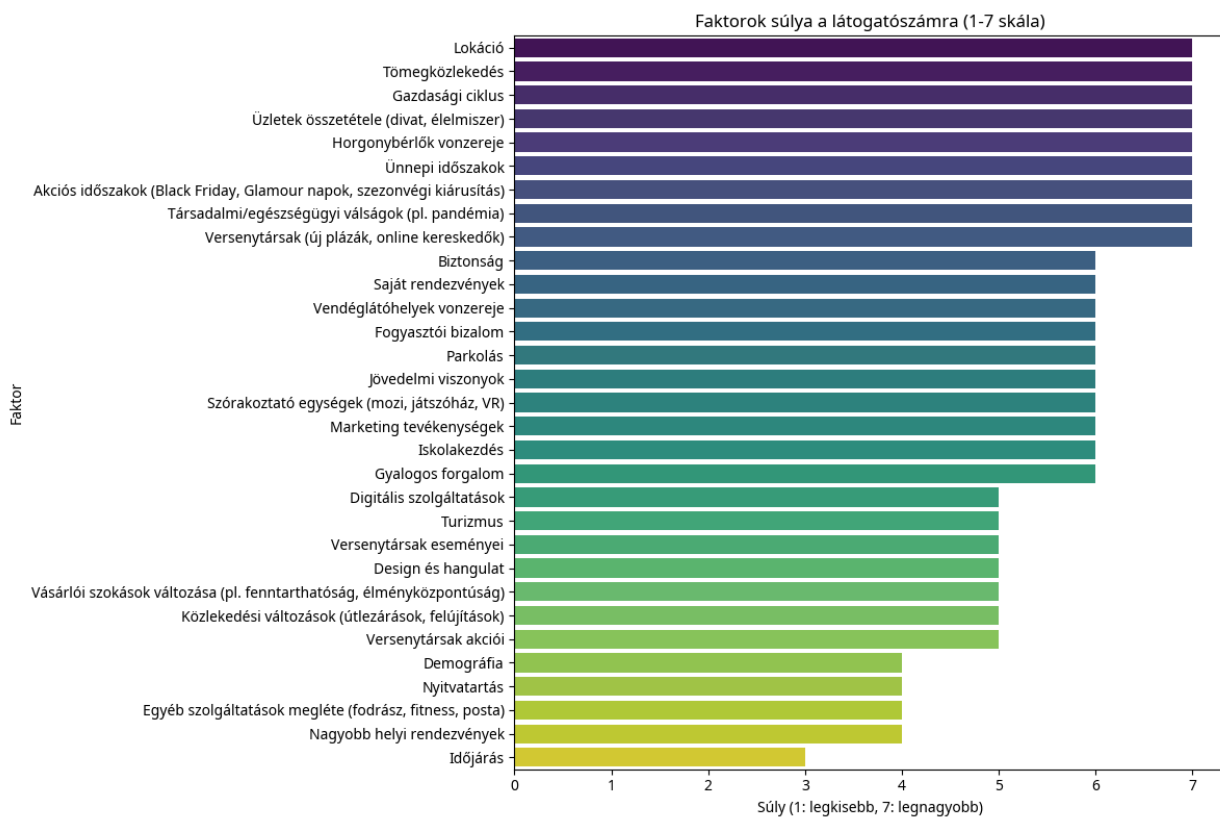
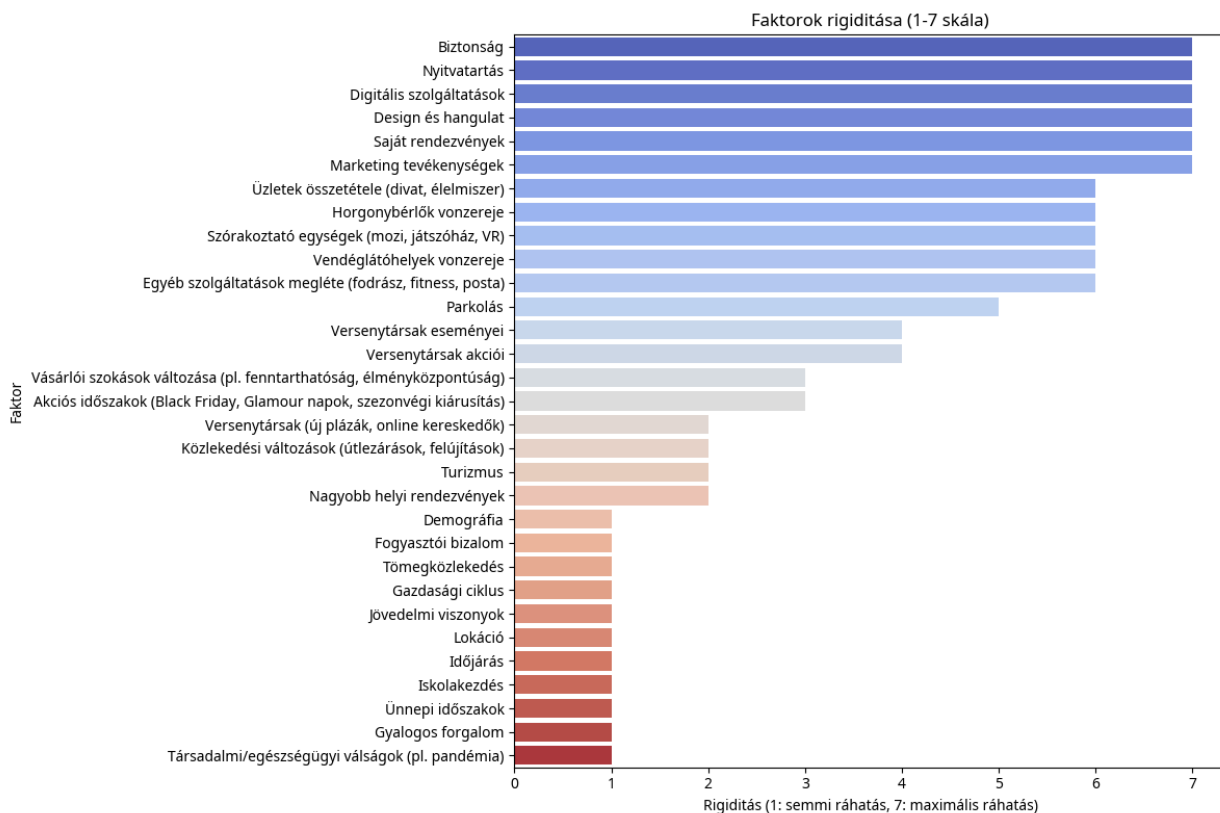
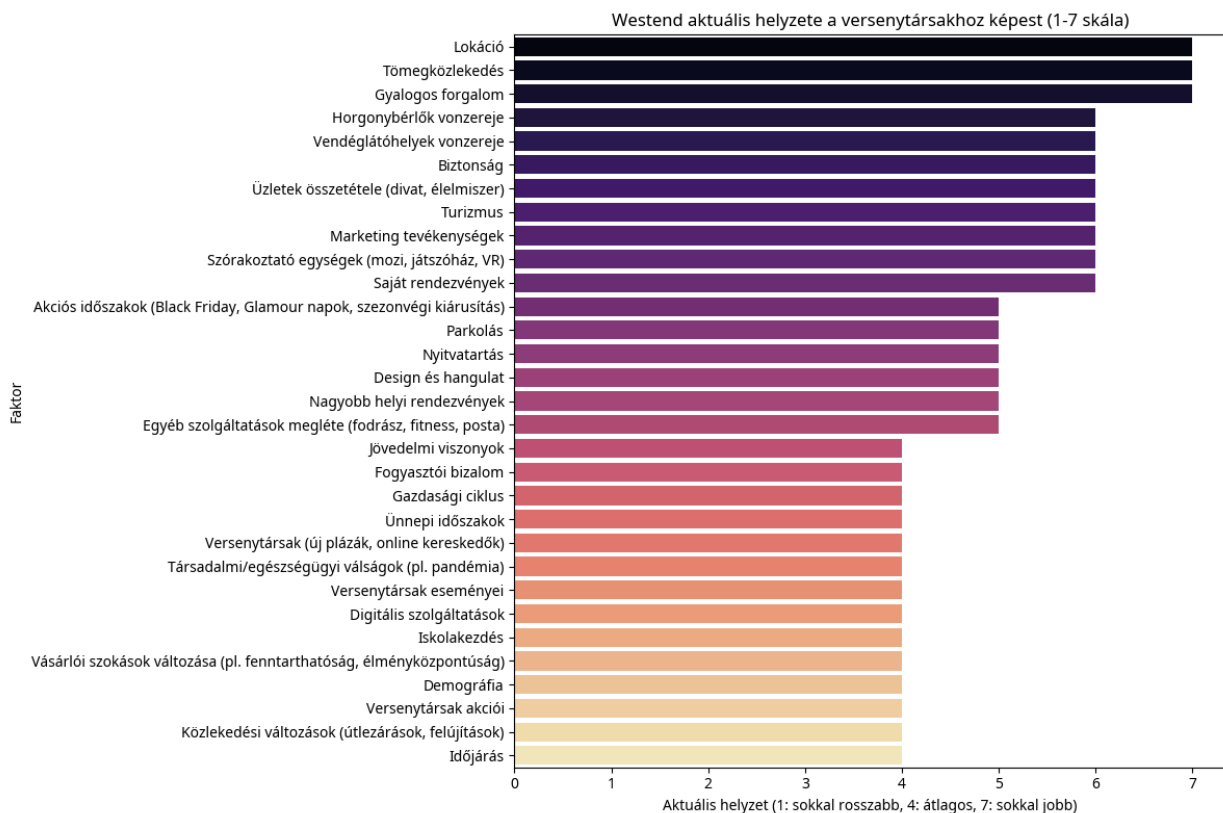


Koncepció	2
Adatmodellek	5
Egylépcsős előrejelző képlet koncepciója	5
Képlet koncepciója (általánosított lineáris modellként szemlélítve):	5
Hogyan működne ez egy AI modellben?	5
Előnyök:	6
Hátrányok:	6
Kétlépcsős előrejelző képlet koncepciója	6
Képlet koncepciója (kétlépcsős megközelítés):	6
Előnyök:	7
Hátrányok:	8
Az előrejelzéshez szükséges adatok és a modellfejlesztés lépései	8
1. Szükséges adatok	8
2. Modellfejlesztés lépései	9
Hackathon	10
Hackathon Célkitűzés: Egyszerűsített Látogatószám Előrejelző Prototípus	10
Fókuszpontok:	10
Miért reális pár óra alatt?	10
Egyszerűsített Modell Koncepció Hackathonra	11
Demonstrációs Terv és Szükséges Eszközök Hackathonra	12
1. Demonstrációs Folyamat (kb. 10-15 perc)	12
2. Szükséges Adatok	13
3. Szükséges Eszközök/Könyvtárak	13
4. Várható Kimenet	13

Koncepció







Shopping mall factors to visitors

File Edit View Insert Format Data Tools Extensions Help

Get professional email like "your-company.com" Plus 2 TB of storage per user, longer video calls, and more with Google Workspace. Try Workspace

100% Calibri 11

	A	B	C	D	E	F	G	H	I
	Dinamika	Faktor jellege	Faktor	Faktorok rigiditása	Faktorok súlya a látogatászáma	Faktorok aktuális helyzete	Adatforrás	Dinamika	
1	Fix	Belső	Lokáció	1	7	7			
2	Fix	Külső	Tömegközlekedés	1	7	7			
3	Fix	Belső	Parkolás	4	5	5			
4	Fix	Külső	Gyalogos forgalom	1	6	7			
5	Fix	Belső	Előző évek látogatászáma	1	7	7	FAKE IT		
6	Lassan változó	Külső	Gazdasági ciklus	1	7	4			
7	Lassan változó	Külső	Jövedelmi viszonyok	1	6	4			
8	Lassan változó	Külső	Fogyasztói bizalom	1	6	4			
9	Lassan változó	Külső	Demográfia	1	4	4			
10	Lassan változó	Külső	E-kereskedelem hatása	1	5	4			
11	Lassan változó	Külső	Versenytársak (új plázák, online kereskedők)	1	7	4			
12	Lassan változó	Külső	Vásárlói szokások változása (pl. fenntarthatóság, élményközpontúság)	3	5	4			
13	Lassan változó	Belső	Horgonybérlek vonzereje	5	7	6			
14	Lassan változó	Belső	Üzletek összetétele (divat, élelmiszer)	5	7	6			
15	Lassan változó	Belső	Vendéglátóhelyek vonzereje	5	6	6			
16	Lassan változó	Belső	Egyéb szolgáltatások megléte (fodrász, fitness, posta)	5	4	5			
17	Lassan változó	Belső	Design és hangulat	5	5	5			
18	Lassan változó	Belső	Biztonság	5	6	6			
19	Lassan változó	Belső	Nyitvatartás	5	4	5			
20	Lassan változó	Belső	Digitális szolgáltatások	7	5	4			
21	Dinamikusan változó	Külső	Időjárás	1	3	4			
22	Dinamikusan változó	Külső	Turizmus	2	5	6			
23	Dinamikusan változó	Külső	Versenytársak akciói	1	5	4			
24	Dinamikusan változó	Külső	Versenytársak eseményei	1	5	4			
25	Dinamikusan változó	Külső	Nagyobb helyi rendezvények	2	4	5			
26	Dinamikusan változó	Külső	Ünnepi időszakok	1	7	4			
27	Dinamikusan változó	Külső							

Convert to table

https://docs.google.com/spreadsheets/d/1QjH_hRVF90rAe-cl9hlyVegHzPPJgU3aePRI_F81MVK/edit?usp=sharing

Adatmodellek

Egylépcsős előrejelző képlet koncepciója

Az egylépcsős (vagy "end-to-end") megközelítés lényege, hogy az összes releváns bemeneti faktort közvetlenül használjuk fel a látogatószám előrejelzésére. Ez a megközelítés egyszerűbb lehet a modell felépítése szempontjából, de megköveteli, hogy a modell képes legyen komplex, nemlineáris összefüggéseket is megtanulni a bemenetek és a kimenet között.

Képlet koncepciója (általánosított lineáris modellként szemléltetve):

Plain Text

Látogatószám = $w_0 + (w_1 * F_1) + (w_2 * F_2) + \dots + (w_n * F_n) + \epsilon$

Ahol:

- Látogatószám: A bevásárlóközpont előrejelzett látogatószáma (pl. napi, heti, havi).
- w_0 : Egy konstans tényező (elfoglaltság, bias), ami a modell alapértelmezett kimenetét adja, ha minden más faktor értéke nulla.
- F_1, F_2, \dots, F_n : Az összes releváns bemeneti faktor, amit az Excel táblázatban azonosítottunk (pl. Lokáció, Parkolás, Gazdasági ciklus, Horgonybérlek vonzereje, Időjárás, Marketing tevékenységek, Ünnepi időszakok, stb.). Fontos, hogy ezeket a faktorokat számszerűsíteni kell (pl. a pontozások, vagy valós adatok, mint hőmérséklet, reklámköltés).
- w_1, w_2, \dots, w_n : Az egyes faktorokhoz tartozó súlyok (koefficiensek). Ezeket a súlyokat az AI/gépi tanulási modell tanulja meg a történelmi adatok alapján. A súlyok azt fejezik ki, hogy az adott faktor mennyire befolyásolja a látogatószámot, és milyen irányban (pozitív vagy negatív korreláció).
- ϵ : Hiba tag, ami a modell által nem magyarázott varianciát reprezentálja.

Hogyan működne ez egy AI modellben?

Egy valós AI modell (pl. neurális háló, döntési fa alapú modell, mint az XGBoost vagy LightGBM) nem feltétlenül egy egyszerű lineáris képletet használna, hanem sokkal komplexebb, nemlineáris összefüggéseket is képes lenne megtanulni. Azonban a fenti képlet jól szemlélteti az alapelvet:

1.Adatgyűjtés: Szükség van nagy mennyiségű történelmi adatra, ahol minden egyes napra/hétre/hónapra rögzítve van a tényleges látogatószám, és az összes bemeneti faktor értéke (pl. az adott nap hőmérséklete, az éppen futó marketing kampány intenzitása, az aktuális gazdasági mutatók, stb.).

2.Adat előkészítés: A kvalitatív faktorokat (pl. "Horgonybérlek vonzereje") számszerűsíteni kell (pl. a már általad is megadott 1-7-es skála, vagy bonyolultabb kódolási módszerek). A hiányzó adatokat kezelni kell.

3.Modell tanítása: Az AI modell (pl. egy regressziós neurális háló) a történelmi adatokon "tanulja meg" a w súlyokat és az F faktorok közötti összefüggéseket, minimalizálva az előrejelzési hibát.

4.Előrejelzés: Amikor új látogatószámot szeretnénk előrejelezni, egyszerűen beadjuk a modellnek az adott időszakra vonatkozó faktorok értékeit, és a modell visszaadja az előrejelzett látogatószámot.

Előnyök:

- Egyszerűbb implementáció, ha a megfelelő adatok rendelkezésre állnak.
- Közvetlen kapcsolat a bemenetek és a kimenet között.

Hátrányok:

- A modellnek kell megtanulnia az összes komplex összefüggést, ami nagy adatmennyiséget és számítási kapacitást igényelhet.
- Nehezebb lehet értelmezni az egyes faktorok pontos hatását, ha a modell nagyon komplex (black box).

Ez a megközelítés akkor a leghatékonyabb, ha az összes faktor közvetlenül mérhető és számszerűsíthető, és a köztük lévő összefüggések viszonylag stabilak az idő múlásával.

Kétlépcsős előrejelző képlet koncepciója

A kétlépcsős megközelítés lényege, hogy a faktorokat először kategóriákba vagy alcsoportokba rendezzük, és ezekre az alcsoportokra külön-külön modelleket építünk, vagy aggregált mutatókat számolunk. Ezeknek az aggregált mutatóknak az eredményeit használjuk fel a második lépcsőben a végső látogatószám előrejelzésére.

Ez a megközelítés különösen hasznos lehet, ha:

- A faktorok különböző dinamikával rendelkeznek (pl. fix, lassan változó, dinamikusán változó).
- Az egyes kategóriákon belüli faktorok szorosan összefüggenek egymással.
- Szeretnénk jobban értelmezni, hogy melyik kategória hogyan járul hozzá a végső látogatószámhoz.

Képlet koncepciója (kétlépcsős megközelítés):

1. Lépcső: Kategória-specifikus indexek vagy rész-előrejelzések

Ebben a lépcsőben minden kategóriához (pl. Fix, Lassan változó, Dinamikusán változó) vagy akár alcsoporthoz (pl. Üzletmix, Élményfaktor) létrehozunk egy aggregált indexet vagy egy

rész-előrejelzést. Ez történhet egyszerű súlyozott átlagolással, vagy akár külön-külön gépi tanulási modellekkel.

- Fix Faktor Index (FFI): $FFI = w_{F1} * F_{Lokáció} + w_{F2} * F_{Tömegközlekedés} + w_{F3} * F_{Parkolás} + w_{F4} * F_{GyalogosForgalom}$ (Itt a w_F súlyok a faktorok látogatószámra gyakorolt súlyát tükrözhetik, vagy a modell által tanult súlyokat.)

- Lassan Változó Faktor Index (LVFI): $LVFI = w_{LV1} * F_{GazdaságiCiklus} + w_{LV2} * F_{JövedelmiViszonyok} + \dots + w_{LVn} * F_{VásárlóiSzokásokVáltozása}$

- Dinamikusan Változó Faktor Index (DVFI): $DVFI = w_{DV1} * F_{Időjárás} + w_{DV2} * F_{Turizmus} + \dots + w_{DVn} * F_{TársadalmiVálságok}$

- Üzletmix Index (UMI): $UMI = w_{UM1} * F_{Horgonybérlok} + w_{UM2} * F_{ÜzletekÖsszetétele} + w_{UM3} * F_{Vendéglátóhelyek}$

- Élmény Index (EI): $EI = w_{E1} * F_{DesignHangulat} + w_{E2} * F_{SzórakoztatóEgységek} + w_{E3} * F_{SajátRendezvények}$

2. Lépcső: Végső látogatószám előrejelzés

A második lépcsőben ezeket az aggregált indexeket vagy rész-előrejelzéseket használjuk fel a végső látogatószám előrejelzésére. Ez is lehet egy egyszerűbb regressziós modell, vagy egy komplexebb AI modell.

Plain Text

$$\text{Látogatószám} = \alpha_0 + (\alpha_1 * FFI) + (\alpha_2 * LVFI) + (\alpha_3 * DVFI) + (\alpha_4 * UMI) + (\alpha_5 * EI) + \dots + \varepsilon$$

Ahol:

- $\alpha_0, \alpha_1, \dots, \alpha_n$: A második lépcsőben tanult súlyok, amelyek azt fejezik ki, hogy az egyes indexek milyen mértékben járulnak hozzá a végső látogatószámhoz.
- FFI, LVFI, DVFI, UMI, EI: Az első lépcsőben számított indexek.
- ε : Hiba tag.

Előnyök:

- Jobb értelmezhetőség: Könnyebb megérteni, hogy melyik faktorcsoporthoz (pl. gazdasági tényezők, üzletmix) hogyan befolyásolja a látogatószámot.
- Moduláris felépítés: Az egyes lépcsők vagy kategóriák modelljei külön-külön fejleszthetők és finomíthatók.
- Robusztusság: Ha egy kategórián belül változnak a faktorok, az első lépcső modellje adaptálható anélkül, hogy a teljes rendszert újra kellene tanítani.
- Adatkezelés: Különböző típusú adatok (pl. statikus, dinamikus, kvalitatív, kvantitatív) kezelése egyszerűbb lehet a kategóriákon belül.

Hátrányok:

- Komplexebb felépítés: Több modell, több lépcső, ami bonyolultabbá teheti a rendszer tervezését és karbantartását.
- Hibák kumulálódása: Az első lépcsőben keletkező hibák átvihetők a második lépcsőbe. Ez a megközelítés akkor ideális, ha a faktorok között hierarchikus vagy csoportosítható kapcsolatok vannak, és szeretnénk mélyebb betekintést nyerni az egyes faktorcsoportok hatásába. Az általad is említett "Dinamika" kategóriák (Fix, Lassan változó, Dinamikusan változó) kiváló alapot szolgáltatnak egy ilyen kétlépcsős modell felépítéséhez.

Az előrejelzéshez szükséges adatok és a modellfejlesztés lépései

Egy AI alapú látogatószám előrejelző modell felépítéséhez elengedhetetlen a megfelelő adatok gyűjtése és a strukturált fejlesztési folyamat.

1. Szükséges adatok

Az Excel táblázatban azonosított faktorok mindegyikéhez szükség van történelmi adatokra. Az adatok típusától függően ezek lehetnek:

A. Belső adatok (Westendtől származó adatok):

- Látogatószám: Napi, heti vagy havi bontásban, minél hosszabb időszoron (több év). Ez a modell kimeneti változója.
 - Parkolás: Napi/heti parkoló kihasználtság, bevétel, esetleges promóciók időpontjai.
 - Horgonybérlok vonzereje / Üzletek összetétele / Vendéglátóhelyek vonzereje:
 - Bérloki mix változásai (új üzletek nyitása, régiék bezárása).
 - Bérlok forgalmi adatai (ha hozzáférhető, anonimizáltan).
 - Bérloki elégedettségi felmérések (kvalitatív adat, számszerűsíthetővé tehető).
 - Egyéb szolgáltatások megléte: Új szolgáltatások bevezetése, meglévők kihasználtsága.
 - Design és hangulat: Felújítások, dekorációk, nagyobb beruházások időpontjai.
 - Biztonság: Incidensek száma, biztonsági intézkedések változásai.
 - Nyitvatartás: Változások a nyitvatartási időben.
 - Digitális szolgáltatások: Weboldal látogatottság, applikáció letöltések/használat, Wi-Fi használat.
 - Marketing tevékenységek: Marketing kampányok időpontjai, típusa, költsége, elérése (pl. hirdetési adatok, közösségi média statisztikák).
 - Saját rendezvények: Rendezvények időpontjai, típusa, látogatottsága, költsége.
 - Szórakoztató egységek: Mozi látogatottság, játszóház kihasználtság, új attrakciók bevezetése.
- B. Külső adatok (nyilvánosan elérhető vagy megvásárolható adatok):
- Lokáció / Tömegközlekedés / Gyalogos forgalom: Ezek alapvetően fixek, de a környező irodaházak, lakóparkok építése, népességváltozás befolyásolhatja a gyalogos forgalmat.
 - Gazdasági ciklus / Jövedelmi viszonyok / Fogyasztói bizalom: KSH adatok (GDP, infláció, átlagkereset, fogyasztói bizalmi index), MNB jelentések.
 - Demográfia: KSH népszámlálási adatok, népességváltozási statisztikák.

- E-kereskedelem: E-kereskedelmi forgalmi adatok (pl. GKI Digital, eNET felmérések).
- Időjárás: Napi hőmérséklet, csapadék, napsütéses órák száma (meteorológiai adatok).
- Turizmus: NTAK adatok (vendégéjszakák száma Budapesten, küldő országok), KSH turisztikai statisztikák.
- Versenytársak akciói/eseményei: Nyilvánosan meghirdetett akciók, rendezvények (versenytársak weboldalai, közösségi média).
- Nagyobb helyi rendezvények: Sportesemények, koncertek, fesztiválok időpontjai és várható látogatószáma (pl. BKK, BVA, sportlétesítmények naptárai).
- Ünnepi időszakok / Iskolakezdés / Akciós időszakok: Naptári adatok, kereskedelmi naptárak.
- Közelekedési változások: BKK közlemények, útlezárások, felújítások időpontjai.
- Társadalmi/egészségügyi válságok: Járványügyi adatok, kormányzati intézkedések (pl. korlátozások).

2. Modellfejlesztés lépései

Az AI alapú látogatószám előrejelző modell fejlesztése tipikusan az alábbi lépésekből áll:

- 1.Probléma definiálása és célkitűzés: Pontosán meghatározni, mit szeretnénk előrejelezni (napi/heti/havi látogatószám), milyen pontossággal, és milyen időtávra.
- 2.Adatgyűjtés és integráció: A fent említett belső és külső adatok összegyűjtése, különböző forrásokból származó adatok egységesítése és egy adatbázisba rendezése.
- 3.Adat előkészítés (Data Preprocessing):
 - Tisztítás: Hiányzó adatok kezelése (kitöltés vagy eldobás), hibás adatok javítása.
 - Transzformáció: Kategorikus változók számszerűsítése (pl. One-Hot Encoding), idősoros adatok megfelelő formátumra hozása.
 - Normalizálás/Skálázás: A különböző skálán lévő numerikus adatok egységesítése a modell jobb teljesítménye érdekében.
 - Feature Engineering: Új, releváns jellemzők létrehozása a meglévő adatokból (pl. "hét napja", "hónap", "ünnepnap-e", "átlaghőmérséklet az elmúlt 3 napban").
- 4.Modell kiválasztása: A probléma jellegéhez (regresszió, idősor előrejelzés) és az adatok mennyiségéhez/típusához illeszkedő AI/gépi tanulási algoritmus kiválasztása. Lehetséges opciók:
 - Lineáris regresszió: Egyszerű, de nemlineáris összefüggéseket nem kezel.
 - Döntési fák / Random Forest / Gradient Boosting (XGBoost, LightGBM): Nagyon hatékonyak, jól kezelik a nemlineáris összefüggéseket és a hiányzó adatokat.
 - Neurális hálók (pl. LSTM idősoros adatokhoz): Képesek komplex mintázatokat felismerni, különösen idősoros adatok esetén.
 - ARIMA/SARIMA: Hagyományos idősoros modellek, jó alapot adhatnak.
- 5.Modell tanítása (Training): Az előkészített adatok egy részén (tanító halmaz) a modell betanítása.
- 6.Modell validálása és tesztelése: A modell teljesítményének értékelése független adatokon (validációs és teszt halmaz). Metrikák: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), R-squared.
- 7.Modell finomhangolása (Hyperparameter Tuning): Az algoritmus paramétereinek optimalizálása a legjobb teljesítmény elérése érdekében.

8. Deployment és monitoring: A betanított modell éles környezetbe helyezése, ahol valós időben képes előrejelzéseket adni. Folyamatos monitoring szükséges a modell teljesítményének nyomon követéséhez és az esetleges romlás (drift) észleléséhez.

9. Iteráció és frissítés: Az adatok folyamatosan változnak, új trendek jelennek meg. A modellnek rendszeres frissítésre és újratanításra van szüksége a pontosság fenntartásához.

Ez a strukturált megközelítés biztosítja, hogy a modell megbízható és pontos előrejelzéseket adjon, segítve a Westend Managementjét a stratégiai döntések meghozatalában.

Hackathon

Hackathon Célkitűzés: Egyszerűsített Látogatószám Előrejelző Prototípus

Cél: Pár óra alatt egy működő, egyszerűsített prototípus elkészítése, amely demonstrálja a bevásárlóközpont látogatószámának előrejelzését néhány kulcsfontosságú faktor alapján.

Fókuszpontok:

1. Adat előkészítés demonstrációja: Megmutatni, hogyan lehet a különböző forrásokból származó adatokat (pl. Excel táblázat, időjárási adatok) egy egységes formátumba hozni.
2. Egyszerűsített modell implementációja: Egy alapvető regressziós modell (pl. lineáris regresszió vagy egy egyszerű döntési fa) betanítása és használata a látogatószám előrejelzésére.
3. Vizualizáció: Az előrejelzések és a valós adatok összehasonlítása egy egyszerű grafikonon.
4. Interaktivitás (opcionális): Egy minimális felhasználói felület (pl. Streamlit vagy Flask alapú weboldal) létrehozása, ahol a felhasználó beállíthatja a faktorok értékeit, és láthatja az előrejelzést.

Miért reális pár óra alatt?

- Egyszerűsített adathalmaz: Nem valós idejű, hatalmas adatbázisokkal dolgozunk, hanem egy előre elkészített, tisztított, kis méretű adathalmazzal.
- Alapvető modell: Nem komplex neurális hálókat vagy idősoros modelleket használunk, hanem egyszerűbb, gyorsan betanítható algoritmusokat.
- Prototípus szint: A cél nem egy éles rendszer, hanem egy koncepció bizonyítása és egy működő demonstráció.
- Fókuszált feladat: Csak a legszükségesebb lépésekre koncentrálnunk: adatbetöltés, modell tanítás, előrejelzés, vizualizáció.

Példa kimenet: Egy weboldal vagy Jupyter Notebook, ami bemutatja:

- A felhasznált faktorokat.
- Az előrejelzési folyamatot.
- A modell által generált látogatószám előrejelzést egy adott időszakra.
- A tényleges és előrejelzett látogatószámok összehasonlítását egy grafikonon.

Ez a célkitűzés lehetővé teszi, hogy a csapat a hackathon rövid időkeretén belül egy kézzelfogható, működő eredményt mutasson be, ami demonstrálja az AI alapú előrejelzés potenciálját.

Egyszerűsített Modell Konceptió Hackathonra

Egy pár órás hackathon keretében a cél egy működő, de egyszerűsített modell demonstrálása. Ehhez a következő koncepciót javaslom:

1. Adatbázis:

- Fókusz: A korábban azonosított faktorok közül válasszunk ki 5-10 legfontosabbat, amelyekhez könnyen hozzáférhető, számszerűsíthető adatok tartoznak.
- Adatforrás: Használjunk egy előre elkészített, kis méretű (pl. 1-2 évnyi napi vagy heti adat) CSV vagy Excel fájlt, ami tartalmazza:
 - Dátum
 - Látogatószám (célváltozó)
 - Időjárás (pl. átlaghőmérséklet, csapadék mennyisége)
 - Ünnepnap (bináris: 0/1)
 - Iskolai szünet (bináris: 0/1)
 - Marketing kampány intenzitása (pl. 1-5 skála, vagy marketing költség)
 - Versenytárs akció (bináris: 0/1)
 - Hét napja (numerikus kódolással, pl. Hétfő=1, Vasárnap=7)
 - Hónap (numerikus kódolással)

2. Modellválasztás:

- Algoritmus: A legegyszerűbb és leggyorsabban implementálható modell a Lineáris Regresszió vagy egy Egyszerű Döntési Fa (Decision Tree Regressor).
- Lineáris Regresszió: Könnyen értelmezhető, gyorsan betanítható. Jól demonstrálja az alapvető összefüggéseket.
- Döntési Fa: Képes nemlineáris összefüggéseket is kezelni, és viszonylag robusztus a kiugró értékekre. Vizualizálható a döntési logika.

3. Modell felépítése (Python példa, scikit-learn könyvtárral):

Python

```
import pandas as pd from sklearn.model_selection import
train_test_split from sklearn.linear_model import LinearRegression #
vagy DecisionTreeRegressor from sklearn.metrics import
mean_absolute_error, r2_score # 1. Adatok betöltése (feltételezve,
hogy a CSV fájl készen van) df = pd.read_csv("hackathon_data.csv") #
2. Jellemzők (X) és célváltozó (y) definiálása features =
```

```

['atlag_homerseklet', 'unnepnap', 'iskolai_szunet',
'marketing_intenzitas', 'het_napja', 'honap'] X = df[features] y =
df['latogatoszam'] # 3. Adatok felosztása tanító és teszt halmazra
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42) # 4. Modell inicializálása és
tanítása model = LinearRegression() # vagy
DecisionTreeRegressor(max_depth=5) az egyszerűség kedvéért
model.fit(X_train, y_train) # 5. Előrejelzés teszt adatokon y_pred =
model.predict(X_test) # 6. Modell teljesítményének értékelése mae =
mean_absolute_error(y_test, y_pred) r2 = r2_score(y_test, y_pred)
print(f"Mean Absolute Error: {mae:.2f}") print(f"R-squared:
{r2:.2f}") # 7. Demonstrációs előrejelzés (pl. egy jövőbeli napra) #
Példa: Holnapra vonatkozó adatok (ezeket manuálisan kell beállítani
vagy szimulálni) future_data = pd.DataFrame([{'atlag_homerseklet':
20, 'unnepnap': 0, 'iskolai_szunet': 0, 'marketing_intenzitas': 3,
'het_napja': 3, 'honap': 10 }]) predicted_visitors =
model.predict(future_data) print(f"Előrejelzett látogatószám
holnapra: {int(predicted_visitors[0])}") # 8. Vizualizáció (egyszerű
scatter plot) plt.figure(figsize=(10, 6)) plt.scatter(y_test, y_pred,
alpha=0.6) plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--',
lw=2) # Tökéletes előrejelzés vonala plt.xlabel("Tényleges
látogatószám") plt.ylabel("Előrejelzett látogatószám")
plt.title("Tényleges vs. Előrejelzett Látogatószám") plt.grid(True)
plt.savefig("prediction_scatter_plot.png") plt.close()

```

4. Hackathon Deliverables:

- A fenti Python szkript (.py fájl).
- Az előkészített hackathon_data.csv fájl.
- A generált prediction_scatter_plot.png kép.
- Egy rövid prezentáció, ami összefoglalja a megközelítést, a felhasznált adatokat, a modell működését és az eredményeket.

Demonstrációs Terv és Szükséges Eszközök Hackathonra

Ez a terv részletezi, hogyan lehet bemutatni az egyszerűsített látogatószám előrejelző prototípust egy hackathonon, és milyen eszközökre lesz szükség hozzá.

1. Demonstrációs Folyamat (kb. 10-15 perc)

1. Bevezetés (2 perc):

- Rövid bemutatkozás és a probléma felvázolása: Miért fontos a látogatószám előrejelzése egy bevásárlóközpont számára?
- A hackathon célkitűzésének ismertetése: Egy egyszerűsített AI modell demonstrálása.

2. Adatok bemutatása (3 perc):

- Az előkészített hackathon_data.csv fájl struktúrájának bemutatása (oszlopok, adatok típusa).
- Elmagyarázni, mely faktorokat vettük figyelembe és miért (pl. időjárás, ünnepnapok, marketing).
- Kiemelni, hogy ez egy egyszerűsített adathalmaz a demonstrációhoz.

3. Modell felépítése és tanítása (5 perc):

- A create_model.py (vagy hasonló nevű) szkript bemutatása (átfutni a kódon, magyarázni a főbb lépéseket: adatbetöltés, felosztás, modell tanítása).
- Elmagyarázni a választott algoritmust (pl. Lineáris Regresszió) és annak működési elvét nagyon röviden.
- A szkript futtatása élőben (ha van rá idő és stabil a környezet, különben előre futtatott eredmények bemutatása).
- A modell teljesítmény metrikáinak (pl. MAE, R-squared) értelmezése.

4. Előrejelzés és Vizualizáció (3 perc):

- Bemutatni, hogyan lehet új adatokkal (pl. holnapi időjárás, marketing tervek) előrejelzést készíteni.
- Az előrejelzett látogatószám bemutatása.
- A prediction_scatter_plot.png grafikon bemutatása, magyarázva a tényleges és előrejelzett értékek viszonyát.

5. Összefoglalás és Következő Lépések (2 perc):

- A prototípus sikereinek kiemelése.
- Milyen további lépésekre lenne szükség egy éles rendszerhez (több adat, komplexebb modell, valós idejű adatintegráció, felhasználói felület).
- Kérdések és válaszok.

2. Szükséges Adatok

- hackathon_data.csv: Egy előre elkészített CSV fájl, amely tartalmazza a történelmi látogatószám adatokat és a kiválasztott bemeneti faktorok értékeit (pl. dátum, látogatószám, átlaghőmérséklet, csapadék, ünnepnap, iskolai szünet, marketing kiadás, hét napja, hónap). Fontos, hogy ez a fájl már tiszta és előkészített legyen, hogy ne kelljen adat előkészítéssel foglalkozni a hackathonon.

3. Szükséges Eszközök/Könyvtárak

- Fejlesztői környezet: Python 3.x
- Függőségek (Python könyvtárak):
 - pandas: Adatok kezelésére és manipulálására.
 - scikit-learn: A gépi tanulási modell (pl. LinearRegression, DecisionTreeRegressor) implementálásához.
 - matplotlib és seaborn: Vizualizációk elkészítéséhez.
- Kód szerkesztő/IDE: Pl. VS Code, PyCharm, vagy egy Jupyter Notebook környezet (utóbbi különösen alkalmas demonstrációra).
- Prezentációs eszköz: Pl. Google Slides, PowerPoint a bevezetéshez és összefoglaláshoz.

4. Várható Kimenet

- Konzol kimenet: A modell teljesítmény metrikái (MAE, R-squared) és egy példa előrejelzés.
- Grafikon: Egy PNG kép (pl. prediction_scatter_plot.png), amely a tényleges és előrejelzett látogatószámokat hasonlítja össze.
- Opcionális (ha van idő): Egy egyszerű webes felület (pl. Streamlit vagy Flask), ahol a felhasználó interaktívan próbálhatja ki az előrejelzést a faktorkok módosításával. Ez jelentősen növelné a demonstráció hatását, de időigényes lehet. Ha nincs idő, a Python szkript futtatása és a generált grafikon bemutatása is elegendő.