fedlap vázlat

**A SZÉKELYFÖLDI MEGYÉK MUNKANÉLKÜLISÉGI RÁTÁINAK ELŐREJELZÉSE, ARMA ÉS A MESTERSÉGES NEURONHÁLÓ-ALAPÚ MODELLEK ALAPJÁN.**

Tartalom

[1. Bevezetés 3](#_Toc157889340)

[1.1 Szakirodalmi áttekintés 3](#_Toc157889341)

[1.2 A munkanélküliségi ráta fogalmának meghatározása Romániában 4](#_Toc157889342)

[1.3 Felhasznált statisztikai mutatók és fogalmak 5](#_Toc157889343)

[2 Az idősorok elemzése 6](#_Toc157889344)

[3 Előrejelzés ARMA módszerrel 7](#_Toc157889345)

[3.1 A stacionaritás vizsgálata 7](#_Toc157889346)

[3.2 Autoregresszív és mozgóátlag modellek (AR, MA, ARMA) 9](#_Toc157889347)

[3.3 Autokorrelációs és parciális autokorrelációs tesztek 10](#_Toc157889348)

[3.3.1 ACF 10](#_Toc157889349)

[3.3.2 PACF 11](#_Toc157889350)

[3.3 Előrejelzés pontosságának meghatározása a hibatesztek alapján 12](#_Toc157889351)

[4 Előrejelzés ANN módszerekkel 15](#_Toc157889352)

[4.1 MLP 15](#_Toc157889353)

[4.2 CNN 15](#_Toc157889354)

[5 A Django webalkalamzás bemutatása 16](#_Toc157889355)

[5.1 Python 16](#_Toc157889356)

[5.2 MVC 17](#_Toc157889357)

[5.3 Django 18](#_Toc157889358)

[5.4 Beolvasás Excel-ből 19](#_Toc157889359)

[5.3.1 Diagramok készítése 22](#_Toc157889360)

[5.5 Mutatók számítása 24](#_Toc157889361)

[5.5 ARIMA előrejelzések 24](#_Toc157889362)

[6 Következtetések 25](#_Toc157889363)

[5. Irodalomjegyzék 26](#_Toc157889364)

# Bevezetés

A munkanélküliség hosszú ideje jelentős gazdasági mutató és központi téma a közgazdaságtani kutatásokban. A munkanélküliség alakulása és változása az adott régió gazdasági egészségét tükrözi, és fontos információkat szolgáltathat a gazdasági kilátásokról. Az idősorok elemzése és az előrejelzés az egyik kulcsfontosságú eszköz a munkanélküliség változásainak megértésében és a gazdasági intézkedések megalapozásában.

IDE MAJD kellene pár cikk a romániai munkanélküliségről, hogy miért aktuális téma…

Ebben az államvizsga dolgozatban Székelyföld regionális munkanélküliségi rátáival foglalkozom. Hargita, Kovászna és Maros megye havi munkanélküliségi rátáit vizsgálom 2010 január és 2022 szeptember között. Az adatokat Románia országos statisztikai hivatalának (Institutul Național de Statistică) hivatalos oldaláról töltöttem le. Célom, hogy statisztikai elemzést készítsek ezekről az idősorokról, valamint megvizsgáljam, hogy ezen idősorok esetében, tíz hónap távlatában az ARMA (autoregresszív mozgóátlag), vagy a mesterséges neurális hálózatokon (ANN) alapuló modellek: CNN (Convolutional Neural Network), MLP (Multilayer Perceptron) nyújtanak pontosabb előrejelzéseket. A „teszt periódusnak” a 2022 október és 2023 július közötti adatokat használtam, ezeket jósoltattam meg a különböző modellekkel és hasonlítottam össze.

A kutatás során egy Python programot készítettem az adatok beolvasásához, feldolgozásához és az eredmények megjelenítéséhez, ehhez egy Django keretrendszerben írt webalkalmazás segítségével biztosítok felhasználói felületet, ezzel szemléletesebbé és egyszerűbbé téve a különböző statisztikai számításokat. A webalkalmazás lényegében bármennyi és bármilyen idősort képes elemezni a megfelelően előkészített adatforrásokból, tehát a jövőben még fel lehet használni más tematikájú elemzésekhez is.

# Szakirodalmi áttekintés

Madaras (2018) Hargita és Brassó megye esetében végzett hasonló regionális kutatást, és megállapította, hogy rövidtávon a mesterséges neuronháló-alapú NAR (nemlineáris autoregresszív) modell, középtávon az ARMA modell nyújtott pontosabb becséleseket.

Davidescu et al. (2021) Románia országos munkanélküliségi rátáival végeztek kutatást, SARIMA, SETAR, Holt-Winters, ETS, és NNAR modelleket hasonlítottak össze, amelyek közül több szempont alapján is az NNAR (neuronhálós autoregresszív) modell mutatkozott a legjobbnak.   
Madaras (2014) ARIMA (1, 1, 4) típusú autoregressziós modellel előrejelzést készített és megjósolta a munkanélküliek számának növekedését 2013 július és 2014 február között Romániában.

Mulaudzi & Ajoodha (2020) Dél-Afrika országos munkanélküliségi rátáinak előrejelzéséhez használtak többféle modellt (ARIMA, ENet, LASSO, LSTM, RNN, Holt-Winters, stb.), amelyekből az MLP (multi-layer perceptron) modell három rejtett réteggel pontosabb előrejelzést adott, mint a többi.

Tufaner & Sözen (2021) Törökország esetében hasonlítottak össze egy MLP két rejtett rétegű, és ARIMA (3, 1, 2) modellt, itt szintén az MLP volt a jobb.

## A munkanélküliségi ráta fogalmának meghatározása Romániában

A dolgozat során felhasznált adatok a Romániai Statisztikai Hivataltól (INSTITUTUL NATIONAL DE STATISTICA) származnak. Az ő módszertanuk a következőképpen definiálja a munkanélküliséget és a munkanélküli rátát:

A **munkanélküliek** a nemzetközi meghatározás (BIM[[1]](#footnote-1)) szerint azok a 15-74 év közötti személyek, akik egyidejűleg teljesítik a következő három feltételt:

* a mérés pillanatában nincs bejelentett munkahelyük
* a következő két héten belül munkába tudnának állni
* az elmúlt négy hétben aktívan munkát kerestek.

A **munkanélküliségi ráta** tulajdonképpen egy százalékos arányszám: a munkanélküliek arányát számolja ki a munkaerőhöz viszonyítva egy adott térségre. Tehát beszélhetünk országos, regionális vagy megyei munkanélküliségi rátáról.

A gazdaságilag aktív népesség a bázisidőszakban az áruk és szolgáltatások előállítására rendelkezésre álló munkaerőt biztosító valamennyi személyt magában foglalja, beleértve a foglalkoztatottakat és a munkanélkülieket is. (INSSE, 2016)

Tehát a munkanélküliségi ráta megmutatja, hogy a munkaképes lakosság hány százaléka nem rendelkezik a mérés pillanatában munkahellyel, viszont tudna és akarna dolgozni.

A következő fejezetekben ismertetem a statisztikai mutatókat, modelleket, amelyeket felhasználtam a dolgozat során.

## Felhasznált statisztikai mutatók és fogalmak

**Átlag**: Az egyik legismertebb statisztikai mutató. Jelöljünk n megfigyelést x1, x2, ..., xn -nel. Ekkor az átlaguk:

**Szórás**: A szórás egy szóródási mutató, ami azt méri, hogy a megfigyelések mennyire esnek távol az átlagtól. Két minta lehet lényegesen különböző, még ha az átlaguk egyforma is, ugyanis az egyik minta megfigyelései eshetnek jóval közelebb az átlaghoz, mint a másik minta megfigyelései. A szóródás méréséhez az eltérések négyzetét veszi figyelembe, mert ezáltal fejezhető ki az összes megfigyelés távolsága az átlagtól. A szórás kiszámításához négyzetgyököt vonunk az eltérések négyzetének átlagából:

**A variancia (szórásnégyzet):**

Egy másik fontos szóródási mutató a variancia, ami egyszerűen a szórás négyzete, vagyis az átlagtól valóeltérések négyzeteinek az átlaga:

**Medián:** Jelöljük x1, x2, ..., xn –nel a megfigyeléseket, és jelöljük x(1), x(2), ... , x(n) –nel ugyanezeket a megfigyeléseket növekvő sorrendben. Tehát, x(1) a legkisebb, x(2) a következő, ... és x(n) legnagyobb:

x(1) ≤ x(2)  ≤ ⋯ ≤ x(n)

A medián a sorrendbe állított x1, x2, ..., xn megfigyelések középső megfigyelése. Ha n páratlan, akkor egészen egyszerű; a medián a (n + 1) / 2 sorrendű megfigyelés.

Ha n páros, akkor 2 középső megfigyelés van, n/2 és (n / 2) + 1, tehát a medián:

(Sándor & Tánczos, 2019)

# Az idősorok elemzése

A webalkalmazásomba feltöltöttem a 2010 január – 2022 szeptember közötti munkanélküliségi ráta adatsorokat a három megyére. A feldolgozás után a program a következő eredményeket jelenítette meg:



*A regionális munkanélküliségi ráták grafikonja*

Látszik, hogy Maros megyében szinte végig a legalacsonyabb a munkanélküliségi ráta, míg Kovásza megyében a legmagasabb. Sok periódusban megfigyelhető, hogy télen magasabb a mutató, mint nyáron, ilyenkor többen kérnek munkanélküliségi segélyt. Szerencsére a 2008-as válság óta folyamatosan csökken az arány, viszont a Covid járvány idején sajnos egy nehezebb periódus jelei láthatóak.

Kovászna és Hargita megyében a vizsgált időszakban 2010 februárjában volt a legmagasabb a munkanélküliségi ráta (12.5% valamint 11.4%), míg Maros megyében 2010 márciusában, 8.5%-os értékkel. Ez bizonyára a 2008-ban kirobbant gazdasági világválság hatása, amely elérte Romániát is, viszont az akkori vezetőség ezt kezdetben nem látta be. Románia erősen érintett volt a gazdasági válság által. Az ország nagymértékben ki volt téve az ingatlanspekulációnak, és külföldi banki tőkének. Az ország gazdasága jellemzően az alacsony és közepes képzettségű munkaerőt használó, viszonylag kevés technológiát felhasználó és kevés hozzáadott értékű iparágakon alapult. A gazdasági recesszió miatt rengeteg munkahely szűnt meg, vagy jelentősen csökkentette dolgozóinak létszámát. (Georgeta, 2015)

Azonban 2020 tavaszáig összeségében nézve folyamatosan csökkent a munkanélküliek száma, ez jól leolvasható a grafikonról, a gazdaság folyamatosan fejlődött, ehhez a technológiai fejlődés is hozzájárult. Maros megyében 2020 májusában volt a legalacsonyabb a mutató, 2.3%, Hargita megyében 2021 novemberében 3.2%, míg Kovászna megyében 2019 májusában 3% volt. Sajnos a koronavírus járvány miatt

A következő táblázat szemlélteti a három idősor átlagát, szórását, szórásnégyzetét (variancia), mediánját, minimum és maximum érétkeit. A táblázatban látható, hogy a téli hónapokban általában magasabb volt a munkanélküliség, mint a többi évszakban. Ez egyfajta szezonalitást is jelenthet.

|  |  |  |  |
| --- | --- | --- | --- |
| **mutató** | **Kovászna** | **Hargita** | **Maros** |
| Átlag | 5.96% | 5.41% | 4.67% |
| Szórás | 2.26% | 1.65% | 1.74% |
| Variancia | 5.09% | 2.71% | 3.01% |
| Medián | 5.2% | 5.1% | 4.7% |
| Minimum | *3.0% 2019 május* | *3.2% 2021 november* | *2.3% 2020 május* |
| Maximum | *12.5% 2010 február* | *11.4% 2010 február* | *8.5% 2010 március* |
| téli átlag | 6.29% | 5.94% | 4.91% |
| tavaszi átlag | 5.95% | 5.31% | 4.75% |
| nyári átlag | 5.86% | 5.10% | 4.47% |
| őszi átlag | 5.72% | 5.28% | 4.55% |

1. *Táblázat: Az idősorok statisztikai mutatói*

A következő lépésekben megvizsgálom ADF és KPSS teszt segítségével az idősorok stacionaritását, ismertetem és a program segítségével elvégeztem az autokorrelációs és parciális autokorrelációs teszteket, ezek segítségével megállapítom azokat az AR/MA/ARMA modelleket, amelyek a legjobban illeszkednek a megyék idősoraira, ezekkel pedig előrejelzéseket készítek.

# Előrejelzés ARMA módszerrel

## A stacionaritás vizsgálata

A stacionaritás az idősorok statisztikai tulajdonságainak időbeni állandóságát vagy közelítő állandóságát jelenti. Egy stacionárius idősor esetén az várható értéke, varianciája és autokorrelációs függvénye állandó, vagy csak időben állandó konstans eltolódásokkal változik. Stacionárius idősorok könnyebben modellezhetők és előrejelzhetők. A stacionaritás meglétét többféleképpen meg lehet állapítani, én a következő teszteket használtam:

* Augmented Dickey-Fuller (ADF) teszt: Ez a teszt azt vizsgálja, hogy az autoregresszív modellben lévő egységgyökér (unit root) mellett az idősor rendelkezik-e egyéb autoregresszív szerkezettel. A nullhipotézis az, hogy az idősor rendelkezik egy egységgyökérrel, tehát nem stacionárius. A programomban a Python statsmodels.tsa.stattools csomagjából az adfuller függvényt használtam fel, ami bekéri az adott idősor megfigyeléseit (lista), majd vissztatér az adfuller statisztika értékével, a p-értékkel, a kritikus értékekkel (1, 5 ,10%). Én csak az 5%-os kritikus értéket vettem figyelembe.
* Kwiatkowski-Phillips-Schmidt-Shin (KPSS) teszt: A nullhipotézis az, hogy az idősor szigorúan stacionárius (az ADF hipotézisével ellentétben), tehát nincs egységgyökér. A programomban a Python statsmodels.tsa.stattools csomagjából a kpss függvényt használtam fel, ami teljesen hasonlóan működik, mint az előbb említett függvény.

A Python webalkalamzásom segítségével elvégeztem a beolvasott adatsorokra ezeket a teszteket, és a következő eredményeket kaptam:

*2.. táblázat: ADF és KPSS tesztek*

|  | **ADF** | | | **KPSS** | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Statisztika** | **p-érték** | **Kritikus Érték (5%)** | **Statisztika** | **p-érték** | **Kritikus Érték (5%)** |
| **Kovászna** | -2.79 | 0.06 | -2.88 | 1.53 | 0.01 | 0.46 |
| **Hargita** | -2.64 | 0.08 | -2.88 | 1.48 | 0.01 | 0.46 |
| **Maros** | -2.37 | 0.15 | -2.88 | 1.65 | 0.01 | 0.46 |

Az ADF statisztika abszolút értéke nagyobb, mint a kritikus érték, tehát elutasíthatjuk a nullhipotézist (azaz az idősorok széria-stacionáriusok). A p-értékek (0.06, 0.08, 0.15) mind nagyobbak, mint az általában elfogadott szignifikanciaszint (például 0.05), tehát elfogadhatjuk a nullhipotézist (azaz az idősor nem széria-stacionárius).

A KPSS statisztika nagyobb, mint a kritikus érték, tehát elutasíthatjuk a nullhipotézist (azaz az idősor nem stacionárius). A p-értékek (0.01, 0.01, 0.01) mind kisebbek, mint az általában elfogadott szignifikanciaszint, tehát elutasítjuk a nullhipotézist (azaz az idősorok stacionáriusak).

A KPSS teszt szerint megállapított stacionaritás jelenlétének magyarázata valószílűleg az 1. táblázatban kimuatott, a téli hónapokban mért magasabb munkanélküliségi ráták.

## Autoregresszív és mozgóátlag modellek (AR, MA, ARMA)

**Az autoregresszió (AR)** azt jelenti, hogy az aktuális időpontbeli értéket a korábbi időpontbeli értékek határozzák meg. Az AR komponens arra utal, hogy az aktuális érték korrelál az előző időpontbeli értékekkel. A "p" paraméter megadja az autoregressziós rendszámot, vagyis hány darab előző időpontbeli értéket használunk az aktuális érték becsléséhez.

Egy p-rendű autoregresszív modellt, amelynek jelzése AR(p), a következőképpen értelmezünk:

Az ismeretlen paraméterek (autoregresszív együtthatók) és az a hibaváltozó, amit **fehérzajnak** feltételezünk, vagyis olyan folyamat, amelynek várható értéke 0, varianciája konstans és autokorrelációja 0, valamint a hibaváltozó kovarianciája az minden késleltetett értékével 0.   
Az AR(p) tulajdonképpen egy többváltozós **lineáris** modell, ahol a regresszorok (független változók) a függőváltozó késleltetett értékei.

Egy AR(p) folyamat akkor stacionárius, ha:

* Az autoregresszív paraméterek abszolútértékei kisebbek, mint 1
* Ezen paraméterek összegének abszolútértékei kisebb, mint 1

A **mozgóátlag** **(MA)** azt jelenti, hogy az aktuális időpontbeli értéket a korábbi időpontbeli hibák lineáris kombinációjaként becsüljük meg. Az MA arra utal, hogy az aktuális érték korrelál az előző időpontbeli hibákkal, és az "q" paraméter megadja a mozgóátlag rendszámát, azaz hány korábbi hibaértéket használunk az aktuális érték becsléséhez. Egy q-rendű mozgóátlag folyamat (MA(q)) jelölése:

Ezek a modellek mindig **stacionáriusok**, vagyis a folyamat autokorrelációi nem változnak az idő függvényében, tehát az idősorban nincsenek trendek is szezonális mintázatok

A két folyamat kombinációja az **ARMA (p, q)** (autoregresszív mozgóátlag) folyamat, amely komplexebb idősorokat is képes leírni csupán két paraméterrel.

ahol p az autoregressziós folyamat rendje, q a mozgóátlag folyamat rendje, az idősorozat aktuális értéke, a konstans érték, az autoregresszív együtthatók, az adott időpontbeli fehérzaj, a mozgóátlag együtthatók.

Egy ARMA folyamat akkor stacionárius, ha az AR komponense stacionárius.

(Sándor, 2019)

## Autokorrelációs és parciális autokorrelációs tesztek

Ebben a részben megvizsgálom és elemzem az autokorrelációs (a továbbiakban ACF) és parciális autokorrelációs (a továbbiakban PACF) teszteket annak érdekében, hogy   
meg tudjam határozni a lehető legjobban illeszkedő idősor modelleket és   
azok paramétereit (AR(p), MA(q), ARMA(p, q)) az egyes idősorokra.

## ACF

Egy idősor autokorrelációs függvénye (ACF) az autokorrelációk sorozata:  
 ahol a k-ad rendű autokovariancia.  
Az ACF segít azonosítani a mozóátlag (MA) folyamatot azáltal, hogy megmutatja, hány időegységnyi korreláció van az aktuális és az előző időpontok között, miközben figyelmen kívül hagyja a köztes időpontokat. Ha egy stacionárius folyamat ACF-je teljesíti azt a feltételt, hogy

(Sándor, 2019)

Az általam vizsgált idősorok autokorrelációs tesztjei a következőképpen néznek ki:



*Elvégzett ACF tesztek a három megyére*

Ha AR folyamat lenne, akkor az első néhány lépés után az autokorrelációk értékei hirtelen zuhannának, viszont a fent látható grafikonok nem ezt mutatják, hanem lineáris, fokozatos csökkenést, emiatt kizárható az, hogy MA(q) folyamatokról lenne szó bármelyik megye esetében is.

## PACF

Egy AR(1) folyamat esetében a pk autokorrelációkfolyamatosan (exponenciálisan) csökkennek. Általában minden AR(p) folyamatnak megvan ez a tulajdonsága, viszont p > 1 rendű folyamatok esetében a csökkenés nem feltétlenül monoton. Ha egy stacionárius folyamat PACF értékei csak a p-edik időbeli lépésben nem nullák (vagyis minden további lépésben megközelítőleg nullák), akkor AR(p) folyamatról van szó.

(Sándor, 2019)

Az általam vizsgált idősorok parciális autokorrelációs tesztjei a következőképpen néznek ki:



*Elvégzett PACF tesztek a három megyére*

Itt mindhárom megye esetében az látszik a PACF tesztek eredményein, hogy az első kettő lépésben az autokorrelációs érték 1, míg az összes többiben elhanyagolhatóak az autokorrelációs kapcsolatok, tehát valószínűleg AR(2) folyamatról beszélünk.

A grafikonok azt szemléltetik, hogy valószínűleg mindhárom megye esetében az AR(2) modellel érdemes próbálkozni az előrejelzéshez, viszont megnéztem még az ARMA(1, 1) és ARMA(1, 2) eseteket is, majd ezekből kiválasztottam a legkisebb AIC[[2]](#footnote-2) (Akaike Information Criterion) értékű modellt a Sándor (2019) alapján, mert valószínűleg ez a modell fog a legjobban illeszkedni az adott idősorra. A következő táblázatban összefoglalom, hogy a különböző modellekre milyen AIC értékeket kaptunk az egyes megyék esetében (a minimum AIC-k celláit beszíneztem):

|  |  |  |  |
| --- | --- | --- | --- |
| Modell | AIC | | |
| Maros | Hargita | Kovászna |
| AR (1) | 9.65 | 128.96 | 123.91 |
| AR (2) | 1.59 | 114.34 | 124.09 |
| ARMA (1,1) | 2.15 | 116.51 | 123.70 |
| ARMA (1,2) | 3.81 | 113.73 | 124.86 |

*A lefuttatott modell tesztek eredményei*

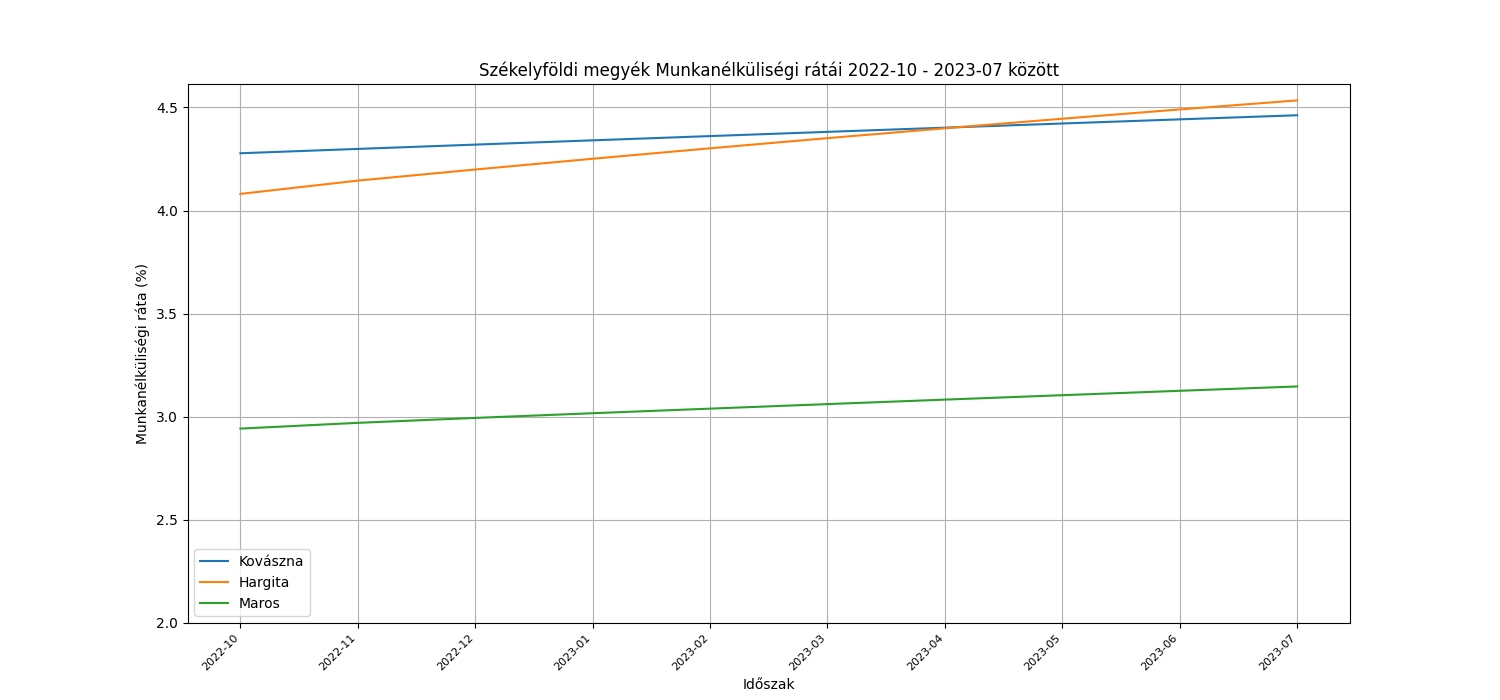
Azt látjuk, hogy az előrejelzéseknél Maros megyénél AR(2), viszont Hargitánál ARMA(1, 2), míg Kovászna megyénél ARMA(1, 1) modellt érdemes használni, nem az AR(2)-t.

## 3.3 Előrejelzés pontosságának meghatározása a hibatesztek alapján

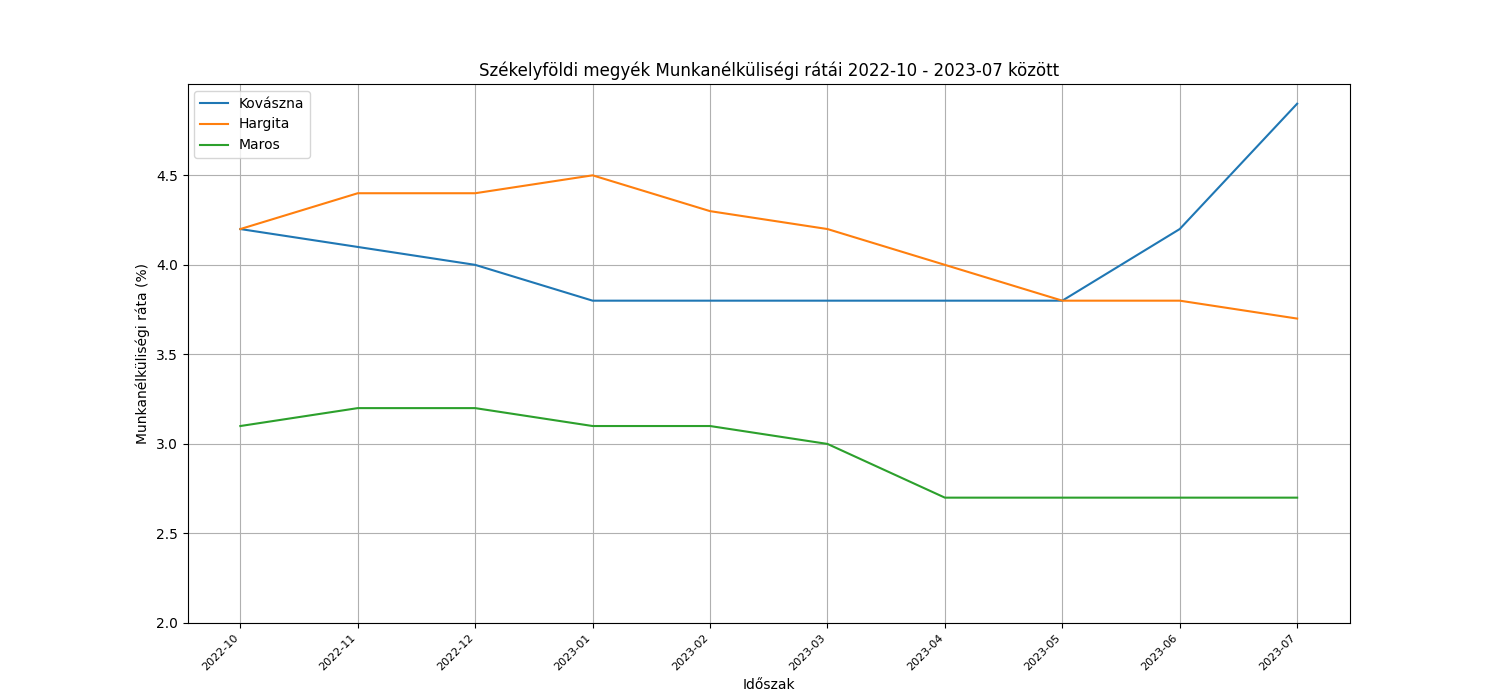
Az adatforrásból szándékosa meghagytam a legfrissebb 10 bejegyzést, hogy az egyes modellek becsléseinek hibáit meg tudjam állapítani a valódi adatokhoz képest.

Miután a programon keresztül kértem a kiválasztott modellekre a 10 hónapos előrejelzéseket, a következő eredményt kaptam:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dátum | *Kovászna* | | *Hargita* | | Maros | |
| mért | ARMA(1, 1) | mért | ARMA(1, 2) | mért | AR(2) |
| 2022 október | *4.2* | 4.28 | 4.2 | 4.08 | 3.1 | 2.94 |
| 2022 november | *4.1* | 4.3 | 4.4 | 4.15 | 3.2 | 2.97 |
| 2022 december | 4 | 4.32 | 4.4 | 4.2 | 3.2 | 2.99 |
| 2023 január | 3.8 | 4.34 | 4.5 | 4.25 | 3.1 | 3.02 |
| 2023 február | 3.8 | 4.36 | 4.3 | 4.3 | 3.1 | 3.04 |
| 2023 március | 3.8 | 4.38 | 4.2 | 4.35 | 3 | 3.06 |
| 2023 április | 3.8 | 4.4 | 4 | 4.4 | 2.7 | 3.08 |
| 2023 május | 3.8 | 4.42 | 3.8 | 4.45 | 2.7 | 3.1 |
| 2023 június | 4.2 | 4.44 | 3.8 | 4.49 | 2.7 | 3.13 |
| 2023 július | 4.9 | 4.46 | 3.7 | 4.53 | 2.7 | 3.15 |



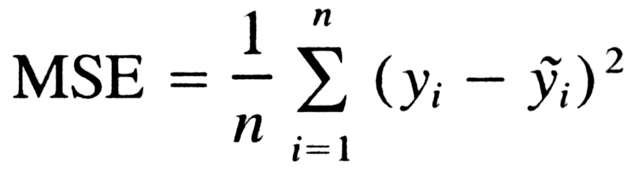
*13. ábra: ARMA előrejelzések grafikonon ábrázolva*



*14. ábra: A valós adatok grafikonon ábrázolva*

A becslések pontosságának megállapításához két mutatót használtam:

* az **átlagos négyzetes eltérést** (Mean Squared Error, MSE), amely a tényleges és becsült adatok közötti különbségek négyzeteinek az átlaga:



* **a relatív átlagos négyzetes** eltérések gyökét (Relative Root Mean Square Error, RRMSE), amely relatívan, normalizálva adja meg a hibákat, százalékos értékben. Az RRMSE az MSE-t normalizálja az aktuális értékek átlagával, és azok szórásával.  
  Minél kisebb ez két mutató, a modell annál jobb becslést ad.

A képletekben n az adatok száma, yi a valóságos, a becsült érték az i-edik mintában. a valós értékek átlaga. A következő táblázatban összefoglaltam az MSE és RRMSE értékeket az egyes modellekre:

|  |  |  |
| --- | --- | --- |
| Modell | MSE | RRMSE |
| Kovászna ARMA(1, 1) | 0.209 | 0.113 |
| Hargita ARMA(1, 2) | 0.195 | 0.107 |
| Maros AR(2) | 0.0826 | 0.0974 |

*MSE és RRMSE értékek*

Mind a Kovászna ARMA(1, 1), mind a Hargita ARMA(1, 2) modell viszonylag alacsony RRMSE-t mutat, ami jó. Azonban a Maros AR(2) modell az RRMSE alapján még jobban illeszkedik a modellhez. A következő lépésben megvizsgélom, a mesterséges neuronhálón alapuló modellekkel milyen MSE és RRMSE mutatókat tudok elérni, vagyis melyik modell tud pontosabb becsléseket produkálni.

# Előrejelzés ANN módszerekkel

## MLP

## CNN

# A Django webalkalamzás bemutatása

## Python

A Python egy magas szintű, objektum-orientált, általános célú programozási nyelv, amely számos különböző problémaosztályra alkalmazható.

A nyelvhez tartozik egy nagy szabványos könyvtár, amely olyan területeket fed le, mint a karakterlánc-feldolgozás, az internetes protokollok (pl. HTTP, FTP), szoftvermérnőki csomagok (pl. egységtesztelés), naplózás, profilozás, Python-kód elemzése és operációs rendszer interfészei (rendszerhívások). (Foundation, General Python FAQ, 2024)

## MVC



Az MVC rövidítés az "Model-View-Controller" (Modell-Nézet-Vezérlő) kifejezést jelenti, és egy szoftvertervezési mintázatot vagy architektúrát takar. Az MVC célja az alkalmazások strukturális szervezésének javítása, hogy könnyebben karbantarthatók és kiterjeszthetők legyenek. Az MVC három fő komponenst tartalmaz:

* Model (Modell): A modell reprezentálja az alkalmazás adatstruktúráit és logikáját. Ez felelős az adatok kezeléséért, az üzleti logika végrehajtásáért, és értesíti a View-t, amikor adatai megváltoznak.
* View (Nézet): A nézet a felhasználói felületet vagy az adatok megjelenítését kezeli. A View értesül a Model változásairól, és frissíti magát, hogy megjelenítse az aktuális adatokat.
* Controller (Vezérlő): A vezérlő a felhasználói bemeneteket kezeli, például gombok lenyomásait vagy más eseményeket. Ezután a vezérlő frissíti a Model-t vagy a View-t a felhasználói interakciók eredményeként.

Az MVC minta alkalmazása segíthet javítani az alkalmazások karbantarthatóságát, kiterjeszthetőségét és tesztelhetőségét. Sok keretrendszer és fejlesztési környezet támogatja az MVC architektúrát, például a Ruby on Rails, a Django (Python), az ASP.NET, Laravel (PHP) és mások.

## Django

A Django egy magas szintű Python-alapú webes keretrendszer, amely biztonságos és karbantartható webhelyek gyors fejlesztését teszi lehetővé. Ingyenes és nyílt forráskódú, aktív fejlesztői közösséggel.

Ez a keretrendszer is követi az MVC szemléletet, viszont a kontroller fájl szerepét itt a views.py fájl tölti be, ahol ugyanúgy függvényekben dolgozzuk fel a szükséges adatot, majd előállítjuk dinamikusan a nézetet,. A nézetek szerepét a hagyományos „view” fájlok helyett „template”, azaz html sablon fájlok veszik át.



URL-ek: Egy URL leképezőt használnak arra, hogy az HTTP kéréseket az érintett nézethez irányítsák a kérés URL-je alapján. A URL leképező képes meghatározott karakterláncok vagy számok mintázataira is illeszkedni a URL-ben, és ezeket adatként továbbítani egy nézetfüggvénynek.

Nézet: A nézet egy kéréskezelő függvény, amely fogadja az HTTP kéréseket, és HTTP válaszokat ad vissza. A nézetek az adatokhoz azokhoz a modellekhez férnek hozzá, amelyekre a kérések teljesítéséhez szükség van, és a válasz formázását sablonokra bízzák.

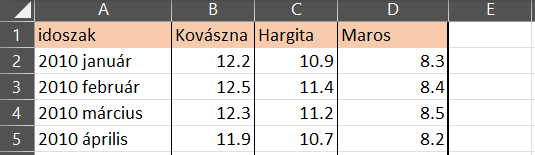
Modellek: A modellek olyan Python objektumok, amelyek meghatározzák egy alkalmazás adatstruktúráját, és mechanizmusokat biztosítanak a rekordok kezeléséhez (hozzáadás, módosítás, törlés) és lekérdezéséhez az adatbázisban.  
Sablonok: Egy sablon egy szöveges fájl, amely meghatározza egy fájl struktúráját vagy elrendezését (például egy HTML oldalét), ahol a helykitöltők a tényleges tartalmat képviselik. Egy nézet dinamikusan létrehozhat egy HTML oldalt egy HTML sablon segítségével, adatokkal feltöltve azt a modellből.

(Foundation, developer.mozilla.org, 2024)

## Beolvasás Excel-ből

A beolvasásra egy HTML űrlapot készítettem, ez a projekt kezdőlapja is egyben. Egyelőre csak Excel fájlokat fogad el (.xls/.xlsx), mivel az adatsorokat rendkívül egyszerűen lehet előkészíteni Excelben.

Fontos, hogy a táblázatok a következőképpen legyenek elkészítve:



Az első sor fejléc kell legyen, ami tartalmazza az idősor(ok) megnevezését, az én esetemben a megyék nevét. Természetesen nem kötelező több idősor megléte. Az első oszlop a megfigyelésekhez tartozó időpontokat kell tartalmazza, ebből készülnek a grafikonok x tengelyen levő feliratai is. A többi sor oszlop pedig a megfigyelt értékeket tartalmazzák.  
Fontos persze az is, hogy időrendi sorrenben legyenek a megfigyelések.   
Az én esetemben például a „data.xslx„ fájl „data” nevű munkalapja tartalmazza a 12 év (2010 január – 2022 szeptember) megfigyeléseit a fenti formában, növekvő időrendi sorrenben.   
A „teszt” nevű munkalap pedig 2022 október és 2023 júlis közötti megfigyeléseket tartalmazza.

Lehetőség van két különböző fájlt is feltölteni, hogyha a teszt adatok - vagyis amelyeket nem vesz figyelembe a program az előrejelzési modellhez, hanem segítségével az előrejelzés pontosságát mutatja ki – külön fájlban vannak. Ha ugyanabban a fájlban van a két munkalap, akkor be kell jelölni az „ugyanaz a fájl” feliratú jelülőnégyzetet, és akkor nem kell ugyanazt a fájlt kétszer feltölteni. Ebben az esetben JavaScript kóddal elrejtem a második fájlfeltöltő bemenetet.



Mindkét esetben meg kell adni a munkalapok nevét, mivel a pandas modullal beolvasott többdimenziós adatszerkezet (DataFrame) asszociatív listákban tárolja az oszlopokat, kulcsai a munkalapok nevei. Ezenkívül lehetőség van megadni azt, hogy az adatsorból generált grafikon x tengelye milyen sűrűséggel legyen feliratozva.

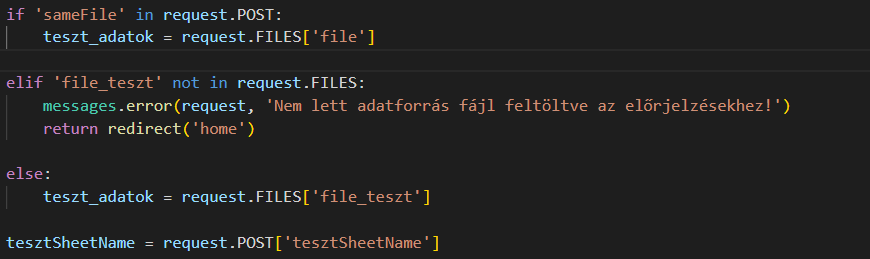
A Feltöltésre kattintva POST kéréssel (ami tartalmazza a feltöltött fájlt, stb.) meghívódik az „upload” nevű URL, amihez az ugyanilyen nevű függvény van rendelve a views.py fájlban.

def upload(request):

    if 'file' not in request.FILES or 'suruseg' not in request.POST or   
 'sheet' not in request.POST:

        messages.error(request, 'Hiányzó paraméter(ek) (sűrűség/munkalap nevek)!')

        return redirect('home')

A függvény hiányzó paraméterek esetén hibaüzenettel visszatéríti a feltöltő oldalra a felhasználót. Különben eltárolom a megkapott adatokat a POST kérésből. Megnézem, hogy be lett-e jelölve a jelölőnégyzet, mert ha igen, akkor ugyanabból a fájlból vesszük a teszt periódus munkalapját, különben ott is leellenőrzöm, hogy fel lett-e töltve a másik fájl.   
   
Ezután a pandas read\_excel függvényével eltárolom egy-egy DataFrame szerkezetbe a táblázatokat. Továbbá külön listákba eltárolon az időszakokat, az idősorok neveit (pl. megyék neveit), és persze a megfigyeléseket (többdimenziós lista, a kulcsok az idősorok nevei), azért, hogy könnyebben fel tudjam dolgozni az adatokat a diagram készítésekor és a weboldalon való megjelenítéskor.

try:

        df\_teszt = pd.read\_excel(teszt\_adatok, sheet\_name=tesztSheetName)

        beolvasott\_teszt\_idoszakok = df\_teszt[df\_teszt.columns[0]].tolist()

        df = pd.read\_excel(uploaded\_file, sheet\_name=sheetName)

        fejlec = df.columns.tolist()

        idoPontok = df[fejlec[0]].tolist()

        for i, col in enumerate(fejlec[1:]):

            adatsorNevek.append(col)

            adatsorok.append(df[col].tolist())

A nézet számára egy könnyen feldolgozható lsitába teszem a beolvasott adatokat, legererálom az adatsorok diagramját, létrehozom a Stat objektumokat (amelyek egy-egy idősor mutatóit, adattagjait tartalmazzák), végül megcsinálom a nézetet az adatokkal és elküldöm a felhasználónak.

statisztikak = createStatObjects(adatsorNevek, adatsorok, idoPontok)

return render(request, 'showData.html',   
{  
 'data\_rows': data\_rows,  
 'adatsorNevek': adatsorNevek,   
 'statisztikak': statisztikak,   
 'diagram': diagram  
})

A render függvény jeleníti meg a html sablont, amely megkapja a szükséges változókat és adatszerkezeteket, tehát a beolvasás után egyből megjelennek az elemzések a felhasználónak.

## 5.5 Diagramok készítése

Mivel többször is kellett grafikont előállítanom különböző adatsorokkal, készítettem egy függvényt (AbrazolEgyben), ami visszaad a bemeneti adatokból egy bájtfolyamot (IO.Bytes), ami a grafikon png formátumú képét tartalmazza. Feltöltés során az upload függvény tehát meghívja ezt a függvényt, elmenti azt png képként és továbbadja a showData.html nézetnek. A függvény a következőképpen működik:

def AbrazolEgyben(adatsorok, idoszakok, megnevezesek, suruseg, Cim="", yFelirat="", y\_min=None, y\_max=None, y\_step=None):

    plt.figure(figsize=(15, 7))

    for i, megye in enumerate(megnevezesek):

        plt.plot(idoszakok, adatsorok[i], label=megye)

    plt.ylabel(yFelirat)

    plt.title(f"{Cim} {idoszakok[0]} - {idoszakok[-1]} között")

    plt.grid(True)

    try:

        plt.xticks(idoszakok[::suruseg], rotation=45, ha="right", fontsize=8)

    except Exception as e:

        print(f"Error: {e}")

    if all((y\_min, y\_max, y\_step)):

        plt.yticks(np.arange(y\_min, y\_max, y\_step))

    plt.legend()

    buffer = io.BytesIO()

    plt.savefig(buffer, format="png")

    buffer.seek(0)

    return buffer

Az adatsorok nevű paraméter egy többdimenziós lista, a beolvasott adatsort vagy adatsorokat tartalmazza, az én esetemben a három megye megfigyeléseit. Az idoszakok egy lista, amely a megfigyelésekhez tartozó időpontokat tartalmazza (x tengely feliratai). A megnevezesek az idősorok neveit tartalamzó lista, az én esetemben például a három megye neve. Ezenkívül a grafikonnak lehet opcionális címet adni, y tengely feliratot, léptéket, minimum és maximum értéket. A beolvasást követően így ábrázolom az adatokat a függvény segítségével:

diagram = AbrazolEgyben(adatsorok, idoPontok, adatsorNevek, suruseg, "Székelyföld munkanélküliségi rátái")

diagram = base64.b64encode(diagram.read()).decode('utf-8')

Ezt a diagram nevű változót átadva a nézetnek, meg lehet jeleníteni egy img tag-ben:

 return render(request, 'showData.html', {'data\_rows': data\_rows, 'adatsorNevek': adatsorNevek, 'statisztikak': statisztikak, 'diagram': diagram})

A showData.html kódja:

 <div id="diagram" class="has-text-centered">

     <img src="data:image/png;base64,{{ diagram }}" alt="Diagram" style=" height: 30%;">

</div>

Kimenet:

## Mutatók számítása

## 5.7 ARIMA előrejelzések

### 5.7.1

# Következtetések

# 5. Irodalomjegyzék

Batuhan TUFANER, M., & SÖZEN, İ. (2021). Forecasting Unemployment Rate in the Aftermath of the Covid-19 Pandemic: The Turkish Case. *İzmir Journal of Economics*, Volume: 36 Issue: 3, 685 - 693, 02.09.2021.

Davidescu, A. A., Apostu, S.-A., & Paul, A. (2021. március 9). Comparative Analysis of Different Univariate Forecasting Methods in Modelling and Predicting the Romanian Unemployment Rate for the Period 2021–2022. *Entropy, 23 (3)*, 325.

Georgeta, E. S. (2015). The economic and social situation in Romania. Európai Unió: European Economic and Social Committee.

Madaras. (2014). A gazdasági válság hatása a munkanélküliség alakulására országos és megyei szinten Romániában. *Közgazdász Fórum, 17 (1-2)*, 136-149.

Madaras, S. (2018). Forecasting the regional unemployment rate based on the Box-Jenkins methodology vs. the Artifi cial Neural Network approach. Case study of Brașov and Harghita counties. *Közgazdász Fórum (Forum on Economics and Business)*, 66.

Mulaudzi, R., & Ajoodha, R. (2020). An Exploration of Machine Learning Models to Forecast the Unemployment Rate of South Africa: A Univariate Approach. IEEE.

Sándor, Z. (2019). *Bevezetés az ökonometriába.* Sepsiszentgyörgy: T3 Kiadó.

Sándor, Z., & Tánczos, L. J. (2019). *Gazdasági statisztika jegyzet.* Sepsiszentgyörgy: T3 Kiadó.

1. Biroul Internaţional al Muncii [↑](#footnote-ref-1)
2. *Az AIC (Akaike Information Criterion) célja az, hogy a modellek közötti összehasonlításra szolgáljon, figyelembe véve a modell illeszkedését és a paraméterszámot. Az AIC egy olyan kompromisszumot keres, amely az illeszkedés minőségét és a modell bonyolultságát egyensúlyozza.* [↑](#footnote-ref-2)