SZAKDOLGOZAT

Magyar Tamás Olivér

2024

Magyar Agrár- és Élettudományi Egyetem
Szenti István Campus
Műszaki Intézet
Adattechnológus-adatelemző szakember szakirányú továbbképzési szak

A készletgazdálkodás hatékonyságának növelése adatelemző módszerekkel

Belső konzulens:	Orova Lászlóné dr. egyetemi docens
intézete/tanszéke:	Műszaki Intézet/ Mérnökinformatikai Tanszék
Külső konzulens:	Juhász Norbert Programtervező informatikus
Készítette:	**Magyar Tamás Olivér** **(PW2TUE)**

Gödöllő

2024

Tartalomjegyzék

1 Bevezetés és célkitűzések

Készletgazdálkodás alatt az anyagok beszerzését, áramoltatását és elosztását értjük. A készletezés szükségessége fizikai és gazdasági kényszerekből adódik. Minden vállalat, amely fizikai termékeket vagy anyagokat ad el, szükséges hogy tartson ezekből a tételekből valamekkora készletet, mivel a termelés és fogyasztás egyes szakaszai időben és térben többnyire elkülönülnek (Benkő, 2018).

A készletgazdálkodás módszereinek köszönhetően a cégek csökkenteni tudják a raktárköltségeket és javítják a pénzforgalmukat. Azoknak a cégeknek, amelyek foglalkoznak a készletek optimalizálásával, átlagban két és félszer gyorsabban tudják növelni a bevételüket. A készletek forgási sebességének növekedésének köszönhetően körülbelül 25-50%-os bevételnövekedésre lehet számítani. A készletkezelés pontossága akár 20%-kal is csökkentheti a munkaerőköltséget, illetve 16%-kal a készletek mennyiségét. Emellett az ügyfelek kiszolgálását is hatékonyabbá és gyorsabbá teszi, ezért az ügyfelek számában is körülbelül 50-100%-os növekedést tudnak elérni azok a cégek, amelyek jól kezelik a készleteiket (Lindner, 2023).

A készletezés szempontjából fontos tényező, hogy mennyire vagyunk tisztában az adott tételek iránti kereslet alakulásával. Egy anyag vagy termék iránti igény lehet folyamatos vagy diszkrét, illetve determinisztikus vagy sztochasztikus (Raisz és Fegyverneki, 2011). Determinisztikus keresletről beszélünk, amikor egy vevő egy adott periódusra előrejelzést ad az adott tételek várható fogyasztásáról, aminek alapján a gyártó vagy viszonteladó tervezni tud a készleteit illetően. Erre az esetre olyan modellt lehet építeni, amelyben a bemenő adatok rögzítve vannak, és nincs valószínűségi változó, tehát a kereslet nagy pontossággal előre jelezhető. A sztochasztikus igény kielégítése ellenben jellemzően egyedi megrendelések alapján történik. Ezt a folyamatot valószínűségi változók határozzák meg, amiből következik, hogy a kereslet véletlenszerűen változik, emiatt nehezebben jelezhető előre.

Szakdolgozatomban a sztochasztikus kereslet kiszolgálásának módszereit vizsgálom. Szakdolgozatom fő kérdése, hogy milyen előrejelzési módszereket alkalmazhatunk a várható igény meghatározására, amennyiben csak historikus adatok állnak rendelkezésünkre, tehát egyéb forrásból nem kapunk előrejelzést a jövőben kiszolgálandó igény alakulásáról. Itt elsősorban a készletgazdálkodás mennyiségi oldalával foglalkozom, tehát azt kívánom meghatározni, hogy milyen modell illeszkedik legjobban az adataimra, amely alapján meg tudom becsülni a jövőbeli fogyasztást, ami segítséget nyújt abban, hogy a lehető

legoptimálisabb készletszintet tudjuk fenntartani. A későbbiekben tárgyalt idősor- és regresszióanalízis különféle módszereit felhasználva fogok elemezni és kiértékelni külön-külön egy adatsort, amely alapján szeretném megállapítani, hogy mely módszerek a leghatékonyabbak az adott adatsorok vonatkozásában. A szakdolgozatban azt is megvizsgálom, hogy a fenti célokra milyen szoftvert érdemes használni.

2 Szakirodalmi áttekintés

2.1 Készletgazdálkodás

A beszerzést követő készletek keletkezésével foglalkozó mechanizmust nevezzük készlet-gazdálkodásnak (Raisz és Fegyverneki, 2011). A készletek feltöltése kapcsán is beszélhetünk determinisztikus és sztochasztikus, utóbbin belül stacioner és nem stacioner változatokról (Benkő, 2018).

A készletgazdálkodás kapcsán érdemes megemlítenünk az Újságárus (Newsvendor) problémát. Ennek a problémának az alapvető kérdése, hogy bizonytalan igény esetén milyen mennyiségben érdemes rendelnünk egy terméket, hogy költségeinket a lehető legalacsonyabban tartsuk, ugyanakkor a vásárlói igényeket is ki tudjuk elégíteni (Pat DeMarle, 2019).

Jelöljük egy termék beszerzési költségét c állandóval. Ha y mennyiséget vásárolunk, és az igény d, akkor elmondhatjuk, hogy az eladott mennyiség az igény és beszerzett mennyiség minimumával lesz egyenlő: $\min[y, d]$. Ha az eladási ár p, akkor a következőképpen határozhatjuk meg a profitot: $p \min[y, d] - cy$ (Scarf, 2022). A profit maximalizálása tehát nagyban függ a várható igény ismeretétől. Ha túl kevés készletet tartunk, potenciális bevételtől eshetünk el, fordított esetben pedig felesleges készletek tartási költségeivel, vagy akár azok elavulásával kell számolnunk.

2.2 Készletezési mechanizmusok

2.2.1 Rendelési időköz meghatározása

A készletgazdálkodásban fontos tényező a rendelési időközök meghatározása. Készleteinket feltölthetjük adott időközönként, vagy egy minimum készletszintet szem előtt tartva (Benkő, 2018). Az előbbi módszer jól működik abban az esetben, ha az igény egyenletes, és nem kell számolnunk lényeges fluktuációval. Az utóbbi módszer jobban megfelel olyan helyzetekben, amikor az igény alakulása kevésbé kiszámítható. Ekkor különösen fontos, hogy valamilyen módszerrel előrejelzést tudjunk adni a várható igényre, hiszen csak ennek ismeretében tudunk

időben reagálni a készletszint változásaira, és meghatározni az ideális rendelési időintervallumokat.

2.2.2 Rendelési tétel nagysága

A rendelési időközökhöz hasonlóan fontos, hogy optimalizáljuk az egyidejűleg rendelt mennyiségeket is. Ez az előzőekhez hasonlóan lehet rögzített, vagy feltölthetjük készleteinket egy meghatározott maximum szintre (Benkő, 2018). Utóbbi esetben a rendelési mennyiség az aktuális készlet illetve a maximum érték különbsége lesz.

A várható igény ismeretében optimalizálhatjuk a rendelt mennyiségeket. Stagnáló igény esetén jól működik az állandó időközönkénti, rögzített mennyiségű rendelés. Csökkenő vagy növekvő tendencia mellett a rendelt mennyiséget ehhez a tendenciához kell igazítanunk, hogy elkerüljük a készlethiányt, vagy felesleges készletek felhalmozását. A trendek monitorozása és előrejelzése különösen fontos például szezonálisan változó keresletű termékek esetében, hiszen ekkor nem hagyatkozhatunk csak egyetlen módszerre.

2.3 A készletfigyelés módszerei

Az optimális készletszint fenntartásához szükséges, hogy rendszeresen ellenőrizzük az aktuálisan raktározott mennyiségeket. Ezt végezhetjük folyamatosan vagy időszakosan (periodikusan). A készletfigyelés módszerét megint csak nagyban befolyásolja a kereslet alakulása. Amennyiben a kereslet állandó, elégséges lehet időszakosan ellenőriznünk készleteinket, ennek ellentétje esetén viszont szükséges annak folyamatos felülvizsgálata, hogy gyorsabban tudjunk reagálni a trendek változásaira.

A folyamatos készletfigyelés esetén, minden időpontban ismert az aktuális készlet mennyisége. Ahogy a készlet egy bizonyos szint alá csökken, azonnal megtörténik az újra rendelés (Benkő, 2018).

Periodikus készletfigyelésről akkor beszélünk, amikor a készletszintet csak bizonyos időközönként ellenőrizzük (pl. hetente vagy havonta), és a vonatkozó rendelések is csak ekkor kerülnek leadásra. Értelemszerűen ebben az esetben nem ismert minden időpontban az aktuális készletszint, kizárólag a legutóbbi ellenőrzésnél rögzített szint elérhető (Benkő, 2018).

A gyakorlatban inkább az előbbi modell van használatban. Mivel ma már minden cég valamilyen számítógépes rendszerben vezeti a készletnyilvántartását, ezek a folyamatok automatizálva vannak.

2.4 Készletezési stratégiák

A fentebb említett készletezési mechanizmusok alapján különböző készletezési stratégiákat alakíthatunk ki, amiket a rendelési idő és mennyiség alapján különíthetünk el. Összesen négyféle stratégiát határozhatunk meg.

Az egyenlő *t* időközönként, *Q* nagyságú tételek rendelését t-Q mechanizmusnak nevezzük. Ilyen módszer alkalmazásakor csak akkor tartható viszonylag állandó készletszint, ha a kereslet is közel állandó. Könnyen belátható tehát, hogy ez a modell nem alkalmas ingadozó kereslet kiszolgálására.

A t-R stratégia alapján adott időközönként egy megadott szintre töltjük a készleteket. Ezt nevezzük ciklikus működési politikának is. A ciklus végén rendelt mennyiség megegyezik az aktuális készletszint és maximum mennyiség különbségével. Ennek előnye, hogy nem szükséges a folyamatos készletellenőrzés és pontosan ütemezhető. Hátránya, hogy a rendelési mennyiségek folyamatosan változhatnak, emiatt ezeket kiszolgálni is nehezebb, továbbá ez a módszer sem alkalmas a kiugróan magas igények kiszolgálására.

r-Q vagy kétraktáras készletezési mechanizmusról akkor beszélünk, amikor egy előre meghatározott minimum készletszint elérésekor meghatározott mennyiségű terméket rendelünk. A „kétraktáras” elnevezés abból ered, hogy a ciklus első felében a felhasználás a minimum készletszint feletti készletből, a második szakaszon viszont a minimum szint alatti biztonsági készletből történik. Az r-Q mechanizmusnak jó az önszabályozó képessége, viszont folyamatos készletfigyelést igényel, és ingadozó kereslet esetén ez sem biztosítja a hiány elkerülését.

A csillapításos működési politika esetében egy minimális szint elérésekor meghatározott szintre töltjük vissza a készletet. Ezt röviden r-R mechanizmusnak nevezzük. Az r-Q stratégiához hasonlóan a ciklusidő itt is változik, viszont nem egy előre meghatározott mennyiséget rendelünk, hanem egy adott maximum szintre töltünk vissza. Emiatt a rendelési mennyiség a t-R stratégiához hasonlóan folyamatosan változni fog, a ciklus végi készletszint függvényében. Ennek előnye, hogy a rendelési időt és mennyiséget is a kereslethez tudjuk

igazítani, tehát összességében flexibilisebb rendszer kialakítását teszi lehetővé, ugyanakkor folyamatos ellenőrzést igényel és szintén nem küszöböli ki az ingadozó kereslet problémáját. Az *1. és 2. táblázat* foglalja össze a fent tárgyalt módszereket (Benkő, 2018; Mihály et al.).

1. táblázat: Készletezési stratégiák
(Forrás: Mihály et al. nyomán)

Rendelési idő	Rendelt mennyiség	Mechanizmus
Állandó időközönként	Állandó mennyiség	t-Q
Állandó időközönként	Maximális készletszintre	t-R
Minimum készletszint elérésekor	Állandó mennyiség	r-Q
Minimum készletszint elérésekor	Maximális készletszintre	r-R

2. táblázat: Készletezési stratégiák előnyei és hátrányai
(Forrás: Benkő, 2018 nyomán)

Mechanizmus	Előnyök	Hátrányok
t-Q	Egyszerűség, egyenletes kereslet esetén megbízható	Nem alkalmas szabályozásra, Ingadozó kereslet esetén nem garantálja a hiány elkerülését
t-R	Nem igényel folyamatos készletellenőrzést, Pontosan ütemezhető	Rendelési mennyiség ingadozása, Ingadozó kereslet esetén nem garantálja a hiány elkerülését
r-Q	Jó önszabályozás	Ingadozó kereslet esetén nem garantálja a hiány elkerülését
r-R	Kereslethez igazítható rendelési idő és mennyiségek	Folyamatos ellenőrzés, Ingadozó kereslet esetén nem garantálja a hiány elkerülését

A *2. táblázatból* jól látszik, hogy egyik készletezési politika sem igazán alkalmas ingadozó igény kiszolgálására. A hiány elkerülése kiküszöbölhető magasabb készletek tartásával, de ekkor számolnunk kell az ebből eredő többletköltségekkel.

A továbbiakban olyan módszerekkel foglalkozom, amelyek segítségével historikus adatok alapján előrejelzést tudunk adni a várható keresletre. Mint fentebb láthattuk, a kereslet meghatározása nagyban hozzájárul a készletgazdálkodás hatékonyságához, hiszen ennek minél pontosabb ismerete nélkül többletköltségek merülhetnek fel, például túl magas biztonsági készlet tartásából adódóan, vagy egy adott periódusban nem tudjuk a beérkező igényeket kielégíteni, ezáltal potenciális bevételtől eshetünk el.

2.4 Igény meghatározása

Minden vállalat számára fontos a tervezhetőség, és alkalmazkodás a piaci igények változásához. Ehhez mindenképp szükség van adatokra, amelyeket felhasználva előrejelzéseket készíthetünk (pl. pénzügyi) a jövőt illetően, ami a cég terveinek alapját képezheti. Ezek az adatok származhatnak külső forrásból (pl. vevői gyártási ütemterv és előre jelzett termékigény) vagy lehet a cég saját belső rendszereiből kinyert adat.

Az igények előrejelzésére használhatunk kvalitatív (pl. szakértői véleményalkotás) és kvantitatív módszereket (Benkő, 2018). Jelen szakdolgozatban csak utóbbival foglalkozom, hiszen ezek köthetőek alapvetően a dolgozat fő kérdéséhez. Az igénytervezéshez először is meg kell értenünk az előrejelzés alapelveit, illetve hogy milyen kimenetet szeretnénk kapni. Fontos továbbá az igényt befolyásoló tényezők ismerete, hiszen tisztában kell lennünk azzal, hogy milyen változókkal dolgozzunk. Nem utolsó sorban, ki kell választanunk a legmegfelelőbb előrejelzési módszert, és meg kell mérnünk az eredményeink pontosságát (Vlckova és Paták, 2010).

A továbbiakban tárgyalt két fő módszer az idősor- és regresszió-analízis, amin belül számos eljárásról írok részletesen.

2.5 Idősor-analízis

Idősoros adatokkal az élet minden területén találkozhatunk. Vállalatoknál kifejezetten gyakoriak a valamely jelenség változását az idő függvényében leíró kimutatások, mint például a tervezett és tényleges kiadások az előző költségvetési évben. A cél mindig az, hogy a vállalat minél pontosabban tudja előre jelezni költéseit és bevételeit, hiszen ezáltal tud megfelelően gazdálkodni az erőforrásaival.

Ehhez szorosan kapcsolódik a készletgazdálkodás kérdésköre, hiszen egy adott termékportfólióra építkező vállalat fő bevételi forrása az adott termékek értékesítéséből származik. Emiatt lényeges, hogy minél pontosabb képet kapjunk a várható keresletről, hogy a lehető legoptimálisabb készletezési stratégiát tudjuk kiválasztani. Erre számos idősor-elemző módszer létezik, amelyeknek a célja, hogy valamilyen matematikai modell segítségével múltbeli megfigyelések alapján megbecsüljük a jövőben elvárt adatokat.

2.5.1 Az utolsó értékre alapozott előrejelzési (naív) eljárás

Az utolsó értékre alapozott előrejelzés a legegyszerűbb becslési módszer. Lényegében az utolsó *t* periódus alapján mondjuk meg, hogy mennyi a *t+1* időszakra adott előrejelzésünk. Ezt a módszert csak abban az esetben érdemes használnunk, ha egyáltalán nincsenek múltbeli adataink, vagy a feltételes eloszlás szórása nagyon kicsi, tehát nem kell tartanunk jelentős becslési hibáktól (Benkő, 2018).

2.5.2 Átlagoló előrejelzési eljárás

Átlagoló előrejelzésről akkor beszélünk, amikor egy adott múltbeli periódus adatainak átlaga alapján becsüljük meg az azt követő időszakra várt értékeket. Ez a módszer csak akkor eredményes, ha az igény eloszlása egyenletes, mivel kiugró értékek jelentősen torzíthatják az előrejelzést. Érdemes továbbá kerülni a nagyon régi adatok használatát, mivel ez a módszer ugyanakkora súllyal veszi figyelembe az összes adatunkat (Benkő, 2018).

Az átlagszámítás az alábbi képlet alapján oldható meg, ahol X_i az adott időpontokban megfigyelt értékeket jelöli, *n* pedig a megfigyelések száma.

$$X_a = \frac{\sum_{i=1}^{n} X_i}{n}$$

2.5.3 Mozgóátlagokra alapozott előrejelzési eljárás (SMA)[1]

A sima mozgóátlagos módszer azon az elven alapul, hogy egy idősor aktuális értéke a korábbi időszakokban megfigyelt értékek átlagának függvénye. Ez az eljárás segít elsimítani a zajos adatokat, ami megkönnyíti a mögöttes mintázatok azonosítását és előrejelzését (Chavan, 2023).

Mozgóátlag számításakor, először meg kell határoznunk, hogy mekkora *n* időszak adatait akarjuk felhasználni. Ez általában tapasztalati úton történik, és legtöbbször 3 és 8 közé esik a vizsgált periódus hossza (Benkő, 2018). A következő érték meghatározásakor az első megfigyelt értéket mindig elhagyjuk, az utolsót pedig hozzáadjuk a számításhoz. Stacionárius idősorokra ez a módszer is kiválóan alkalmazható. Hátránya, hogy ugyanakkora súllyal vesz figyelembe minden értéket.

A módszer az alábbi képlettel írható le, ahol X_i az *i*-edik periódusban megfigyelt érték, *n* pedig a periódusok száma (Benkő, 2018).

$$E(X_{t+1}) = \frac{X_t + X_{t-1} + \cdots + X_{t-n+1}}{n} = \frac{1}{n} \sum_{i=t-n+1}^{t} X_i$$

2.5.4 Előrejelzési eljárás súlyozott mozgóátlaggal (WMA)[2]

A mozgóátlagos előrejelzési eljárásnak egy alternatívája a súlyozott mozgóátlag, amely az adatok időrendiségére alapozva, nagyobb súllyal veszi figyelembe a jelenhez közelebbi adatokat. A sima mozgóátlaghoz hasonlóan, ebben az esetben is meg kell határoznunk az átlagolt periódus hosszát. A súlyokat szabadon meg tudjuk választani az alapján, hogy az adataink milyen trendet mutatnak, illetve hogy vannak-e kiugró értékeink, amelyeket szeretnénk minél jobban elsimítani. A módszert az alábbi képlettel írhatjuk le, ahol X_t az adott időpontokhoz köthető megfigyeléseket, W_i pedig az alkalmazott súlyt jelöli (Radečić, 2021).

$$Y_t = \frac{\sum_{i=0}^{t} W_i X_{t-i}}{\sum_{i=0}^{t} W_i}$$

[1] Az angol „Simple Moving Average” kifejezésből
[2] Az angol „Weighted Moving Average” kifejezésből

2.5.5 Előrejelzési eljárás exponenciálissimítással (ES)[3]

Az exponenciális simítás, a súlyozott mozgóátlagos módszerhez hasonlóan, olyan előrejelzési eljárás, amely exponenciálisan nagyobb súlyokat rendel a jelenhez közelebb megfigyelt értékekhez. Ez egy adaptív módszer, mert automatikusan korrigálja a használt súlyokat a múltbeli adatok alapján (Malkari, 2023). A módszernek háromféle típusát különböztetjük meg:

1. A szimpla exponenciális simítás (SES)[4] a legegyszerűbb módszer a három közül, amely csak a korábbi megfigyeléseket és a hozzájuk tartozó előrejelzéseket veszi figyelembe. Ezt az alábbi képlettel tudjuk leírni, ahol F_t az előző időszakra vonatkozó előrejelzés, Y_t az ugyanebben az időszakban tett megfigyelésünk, és *alfa* a simítótényező (Benkő, 2018). Utóbbi 0 és 1 közötti értéket vehet fel.

$$F_{t+1} = \alpha * Y_t + (1 - \alpha) * F_t$$

   A szimpla exponenciális simítás alkalmas olyan idősorok elemzésére, amelyek nem mutatnak semmilyen trendet vagy szezonalitást.

2. A kettős exponenciális simítás (DES)[5] olyan idősorok elemzésére alkalmas kifejezetten, amelyek valamilyen jól meghatározható trendet mutatnak. A szimpla módszerhez képest kiegészíti az egyenletet egy trend (T_t) simítási szint (L_t) komponenssel. Ezek hozzáadásával az alábbiak szerint tudjuk felírni a módszer képletét, ahol *béta* a trend simítótényezője (Malkari, 2023).

$$L_t = \alpha * Y_t + (1 - \alpha) * (L_{t-1} + T_{t-1}) \qquad (1)$$
$$T_t = \beta * (L_t - L_{t-1}) + (1 - \beta) * T_{t-1} \qquad (2)$$
$$F_{t+1} = L_t + T_t \qquad (3)$$

3. A harmadik, egyben legkomplexebb módszer, a háromszoros exponenciális simítás (TES).[6] A korábbi két altípushoz képest ennél a módszernél meghatározunk egy szezonális komponenst (S_t) is, ezáltal a módszer alkalmas lesz szezonális trendeket mutató idősorok alapján való előrejelzésre. A szezonális komponens és előrejelzés

[3] Az angol „Exponential Smoothing” kifejezésből
[4] Az angol „Simple Exponential Smoothing” kifejezésből
[5] Az angol „Double Exponential Smoothing” kifejezésből
[6] Az angol „Triple Exponential Smoothing” kifejezésből

képletét így írhatjuk fel, ahol *gamma* a szezonális komponens simító tényezője, *m* pedig a szezonális időszak hossza (Malkari, 2023).

$$S_t = \gamma * \frac{Y_t}{L_t} + (1 - \gamma) * S_{t-m} \quad (1)$$

$$F_{t+1} = (L_t + T_t) * S_{t-m+1} \quad (2)$$

Az exponenciális simítás nagy előnye, hogy egyszerűen kivitelezhető, és alacsony a számítási költsége, ezért alkalmas nagy adathalmazok elemzésére. Mindemellett egy rendkívül adaptív módszerről van szó, mivel mindig a legfrissebb adatok alapján változtat az előrejelzéseken (Malkari, 2023).

2.6 Regresszió-analízis

Regressziós modelleket akkor használunk, amikor egy függő változó (*y*) értékét próbáljuk meghatározni egy vagy több független változó (X_1, X_2 ... X_n) értéke alapján. A regressziós modellek segítségével rávilágíthatunk a függő és független változók kapcsolatára, ezáltal kiváló eszköz lehet a függő változók értékének becslésére (Chatterjee és Simonoff, 2013). Egy regressziós modell felállításánál olyan függvényt keresünk, amely a lehető legjobban írja le a változók közötti összefüggést (Benkő, 2018).

Számos különféle regressziós modellt alkalmazhatunk előrejelzésre, amikor valamilyen változó függvényében próbáljuk meghatározni a várható eladásokat. A továbbiakban néhány ilyen módszerről írok részletesen.

2.6.1 Lineáris Regresszió

A lineáris regresszióval a függő és független változó között keresünk lineáris kapcsolatot. Megkülönböztetünk egy- vagy többváltozós lineáris regressziót. A módszer az alábbi képletekkel írható fel, ahol *y* a függő változó, β_0 a tengelymetszet, és β_i az *X* együtthatója. Az alábbi (1) képlet az egyváltozós, a (2) képlet pedig a többváltozós lineáris regressziót írja le.

$$y = \beta_0 + \beta_1 X \quad (1)$$

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_i X_i \quad (2)$$

A legjobban illeszkedő egyenest a legkisebb négyzetek módszerével határozhatjuk meg, amelynek célja olyan bemenő értékek meghatározása, amelyekkel a négyzetes hibák összege a legkisebb (Maulud és Abdulazeez, 2020). A korrelációs együtthatót (r) az alábbi képlettel számíthatjuk ki, ahol d_x a független változó értékeinek, d_y pedig a függő változó értékeinek átlagtól vett különbségét jelöli (Balázsiné et al., 2003). A determinációs együtthatót úgy kapjuk meg, hogy a korrelációs együttható értékét négyzetre emeljük. Egyenes illesztésekor a D legkisebb értékét keressük.

$$r = \frac{\sum d_x d_y}{\sqrt{\sum d_x^2 \sum d_y^2}} \quad (1)$$

$$D = r^2 * 100 \quad (2)$$

A korrelációs és determinációs együttható megmutatja a függő változóban a független változóval magyarázható variancia arányát. Előbbi -1 és 1, utóbbi pedig 0 és 1 közötti értéket vehet fel. Minél nagyobb számot kapunk, annál szorosabb lineáris kapcsolat áll fenn a változók között, tehát ez a modell kifejezetten alkalmas olyan adatsorok elemzésére, ahol a változók erős lineáris kapcsolatban van. Amennyiben nem figyelhető meg ilyen kapcsolat, érdemes más regressziós modellt használnunk a jobb eredmény elérése érdekében.

2.6.2 Kvantilis Regresszió

A kvantilis regresszió ötlete Koenker és Bassett (1978) nevéhez köthető. Ezek olyan modellek, amelyekben a válaszváltozó feltételes eloszlásának kvantiliseit[7] a megfigyelt kovariánsok függvényében fejezzük ki (Koenker és Hallock, 2001). Kvantilis regresszió során a függő változó mediánját, vagy valamilyen egyéb kvantilisét keressük. A kvantilis tulajdonképpen konfidenciaintervallumként működik, tehát a független változó értékei alapján meg tudjuk határozni a függő változó értékeinek maximumát valamekkora valószínűséggel.

A medián érték kiszámításához nem a lehető legkisebb determinációs együtthatót, hanem az abszolút eltérések összegének legkisebb értékét keressük. Minden egyéb kvantilis meghatározásakor a kvantilis értékének megfelelő súlyozást is alkalmaznunk kell (Koenker és

[7] Az az ismérvérték, amely két részre osztja a sokaságot

Hallock, 2001). Ezt legjobban az alábbi képlettel tudjuk leírni, ahol β_0 a tengelymetszet, β_i az *X* együtthatója, és *tau (τ)* a kvantilis számát jelöli.

$$Q_\tau(y_i) = \beta_0(\tau) + \beta_1(\tau)X_1 + \cdots + \beta_i(\tau)X_i$$

A képlet alapján elmondhatjuk, hogy a *béta* együtthatók ebben az esetben nem konstansok, hanem a kvantilis függvényei. A *béta* értékek maghatározásához a mediántól vett abszolút eltérések (MAD)[8] összegét kell minimalizálnunk. Ezt az alábbi képlet írja le, ahol *ró (ρ)* a veszteségfüggvény, amely aszimmetrikusan súlyozza a hibákat a kvantilis értékétől függően.

$$MAD = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau \left(y_i - (\beta_0(\tau) + \beta_1 X_1(\tau) + \cdots + \beta_i(\tau)X_i)\right)$$

A *ró* értékét a következő képlettel tudjuk kiszámítani, ahol *u* egy adott értékre vonatkoztatott hiba. Ha a hiba értéke pozitív, a hibát *τ* értékével szorozzuk, egyéb esetben (1-*τ*) értékével (Dye, 2020).

$$\rho_\tau(u) = \tau \max(u, 0) + (1 - \tau)\max(-u, 0)$$

2.6.3 Tartóvektor Regresszió (SVR)[9]

A tartóvektor regresszió (a továbbiakban SVR) egy gyakori adatelemző algoritmus, amelyet osztályozási és regressziós feladatok megoldására is használnak. A SVR lehetővé teszi egy bizonyos hibahatár deifiniálását a modellünkben. A legkisebb négyzetek módszerével ellentétben, a SVR célja az együtthatók minimalizálása. Az *ε* (epszilon) változó bevezetésével tudjuk állítani a megengedett hibahatárokat, ezáltal felállítunk egy olyan hipersíkot, amely a lehető legjobban illeszkedik az *ε* értékek által határolt tartományra (Rasifaghihi, 2023).

Matematikai egyenlettel az alábbiak szerint írhatjuk fel az *f(x)* folytonos célfüggvényt, ahol $\emptyset(X_i)$ az X_i bemeneti adat által kiválasztott magasabb dimenziós térbe való leképezése, W_i a súly, *b* pedig a torzítás, amely kiegyenlíti az előrejelzéseket (Rasifaghihi, 2023).

[8] Az angol „Median Absolute Deviation” kifejezésből
[9] Az angol „Support Vector Regression” kifejezésből

$$f(x) = \sum_{i=1}^{n} W_i \, \emptyset(X_i) + b$$

A SVR lényegében lineáris regressziót hajt végre egy magasabb dimenziós térben, ahol a bemeneti adatok alapján egy hiperpályát keres. A SVR különböző kernelfüggvényeket használhat a bemeneti adatok magasabb dimenziós térbe való leképezéséhez, így nemlineáris összefüggések modellezésére is alkalmas. Ezek közül a leggyakoribbak a lineáris, polinomiális, illetve a radiális bázis (Gauss-féle) függvény (RBF)[10] kernel. Utóbbi kettő alkalmas nemlineáris összefüggések modellezésére is, így ezeket előszeretettel használják gyakorlatban való hatékonyságuk miatt (Rasifaghihi, 2023).

2.6.4 Döntési fa (decision tree)

Döntési fákkal képesek vagyunk osztályozási és regressziós modelleket építeni. Ezeknek a modelleknek a struktúrája egy fára emlékeztet, amelynek vannak különféle csomópontjai (gyökér, levél, belső csomópontok). Ez a faszerű struktúra úgy jön létre, hogy az algoritmus az adott csomópontokban kérdéseket tesz fel, és az adathalmaz struktúrájától függően döntési szabályokat hoz létre (Gacar és Kocakoc, 2020).

A döntési fa kiinduló pontja a gyökér csomópont, amely további ágakra, majd levelekre bomlik szét. A belső csomópontok a független változót, az ágak a döntési szabályokat, a levél csomópontok pedig a függő változót jelölik. A döntési fák képesek modellezni a változók közötti kapcsolatot homogén alosztályok létrehozásával, miközben figyelembe veszik az adathalmaz heterogenitását. Az elágazási folyamat a legkisebb négyzetek elve szerint történik (Gacar és Kocakoc, 2020).

A döntési fák előnye, hogy kevesebb adatelőkészítést igényelnek, nem szükséges normalizálni az adatainkat, a hiányzó adatok nem befolyásolják lényegesen a modellt, illetve nagyon egyszerű dolgozni velük, viszont hajlamosak a túlillesztésre.

2.6.5 Véletlen Erdő (Random Forest) Regresszió

A véletlen erdő egy olyan osztályozási és regressziós módszer, amely több különböző döntési fát hoz létre, és ezeknek az eredményeit átlagolja. A véletlen erdő regresszió során az

[10] Az angol „Radial Basis Function” kifejezésből

algoritmus különböző mintákat vesz az adathalmazból, és ezekből hozza létre a döntési fákat. A módszer különlegessége, hogy a fák készítésekor véletlenszerű elrendezés történik, vagyis minden fa csak egy részhalmazát használja a bemeneti adatoknak. Minden fa elkészíti a saját előrejelzését a saját bemeneti adatai alapján, amit a modell végül átlagol (Liaw és Wiener, 2001). A véletlen erdő nagy előnye, hogy a sima döntési fához képest, sokkal kevésbé hajlamos a túlillesztésre a véletlenszerű rendezésnek köszönhetően.

2.6.6 Mérőszámok

1. Az eddig tárgyalt módszerek pontosságának számszerűsítésére három mérőszámot használok: Az egyik az átlagos abszolút hiba (MAE)[11], amely a ténylegesen mért és előre jelzett értékek közötti különbségek abszolút értékének az átlaga. Az átlagos abszolút hiba az egyes hibákat egyenlően súlyozza, ezért az eredmény a hibák nagyságával egyenesen arányos. (Acharya, 2021). Az átlagos abszolút hibát az alábbi képlettel kapjuk meg, ahol y_i az egyes előre jelzett értékeket, X_i a tényleges értékeket, n pedig a megfigyelések számát jelöli.

$$MAE = \sum_{i=1}^{n} \frac{|y_i - x_i|}{n}$$

2. A második használt mérőszám az átlagos négyzetes hiba négyzetgyöke (RMSE)[12]. Ez a mérőszám sokkal érzékenyebb a hibákra, mivel ebben az esetben a hibák négyzetével számolunk, ezért néhány nagyobb hiba jelentős növekedést idézhet elő az RMSE értékében. Az RMSE értékét az alábbi képlettel tudjuk kiszámolni (Moody, 2019).

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{n}}$$

[11] Az angol „Mean Absolute Error” kifejezésből
[12] Az angol „Root Mean Squared Error” kifejezésből

3. A harmadik felhasznált mérőszámom az abszolút százalékos hiba (MAPE)[13], amely százalékosan határozza meg a tényleges és becsült értékek közötti eltérések abszolút értékének átlagát. A százalékos értéket úgy kapjuk meg, hogy az alábbi számítás eredményét százzal megszorozzuk. Az abszolút százalékos hiba hátránya, hogy nulla vagy ahhoz közeli értékek esetén, rendkívül nagy hibákat eredményezhet. Mivel az esetünkben használt adatsor nem tartalmaz ilyen elemeket, nyugodtan használhatjuk (Kim és Kim, 2016).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right|$$

Mindhárom mérőszám esetében a lehető legkisebb érték a legjobb, tehát ekkor kapjuk a lehető legpontosabb modellt.

[13] Az angol „Mean Absolut Percentage Error” kifejezésből

3 Alkalmazott módszerek (anyag és módszer)

Szakdolgozatomban az adattudományi folyamat módszereit használtam adatelemzésre, amit OSEMN[14] keretrendszernek is nevezünk. A szakdolgozatban leírt módszereket alapvetően Python programozási nyelv segítségével implementáltam az adatsorok elemzésére. Azért döntöttem a Python nyelv használata mellett, mert ez egy nyílt forráskódú, bőséges dokumentációval ellátott nyelv, illetve számos olyan Python könyvtár van, amelyek alkalmasak az általam használt adatelemző módszerek kidolgozására.

1. Az adattudományi folyamat első lépése az adatok megszerzése. A munkám során .csv kiterjesztésű fájlokat használtam, amit internetes keresővel találtam webes forrásból. Az adatokat a Pythonban elérhető *Pandas* könyvtár segítségével olvastam be DataFrame-be.[15]
2. A folyamat második lépése az adatok előkészítése, aminek a célja az adatminőség javítása. Ez magában foglalja az adatok szűrését, a formátum egységesítését, hiányzó, duplikált és kiugró értékek kezelését, üres oszlopok törlését, összetett oszlopok szétbontását, illetve dimenziócsökkentést. Ezt a lépést részben Python, részben pedig Excel segítségével oldottam meg.
3. A folyamat harmadik lépése az adatok feltárása, ami az adatokon végzett elsődleges vizsgálatot jelenti leíró statisztikai és adatvizualizációs módszerekkel. Ennek célja, hogy megismerjük az adatsor sajátos jellemzőit. Erre a célra a *Pandas* könyvtár *DataFrame.describe()* függvényét használtam, amely megadja a kiválasztott változók alapvető statisztikai adatait.
4. A folyamat negyedik lépése az adatok modellezése. Készíthetünk modelleket különböző célokra, például előrejelzésre, osztályozásra vagy klaszterezésre. Első lépésben az adataimból létrehoztam x és y tömböt, amelyek az összehasonlítandó változókat tartalmazzák. A regresszió esetében ezeket 1:4 arányban felosztottam tanuló és teszt adatokra a *Scikit-learn* könyvtár *train_test_split()* függvényével. Idősor-elemzéshez az *iloc*[] függvénnyel végeztem el a Dataframe két részre bontását. A modellek létrehozásához a *Numpy, Statsmodels* és *Sci-kit Learn* könyvtárakat használtam. Vizuális ábrázoláshoz a *Matplotlib, Plotly és Seaborn* könvytárakkal

[14] Az angol Obtain, Scrub, Explore, Model, iNterpret szavakból.
[15] Olyan adattípus, amely két dimenziós struktúrába rendezi az adatokat.

dolgoztam. A modellek kiértékelését a *sklearn.metrics* könyvtár segítségével végeztem el.

5. A folyamat ötödik lépése az adatok értelmezése, ami magában foglalja az adatok kommunikálását és vizualizációját. Ebben a lépésben adjuk meg a választ az eredeti kérdéseinkre, és írjuk le végső megállapításainkat (Lau, 2019).

Összességében minden vizsgált módszer esetében ugyanazt a sémát követtem. Első lépésben beolvastam az adatokat, amiből meghatároztam azokat a bemeneti változókat, amelyek a modellek alapjául szolgálnak. Az adatokat tanuló és teszt adatokra bontottam. A tanuló adatokkal elvégeztem az adott modell illesztését, ami alapján meg tudtam határozni az előre jelzett értékeket. Végül kiértékeltem a modellt tanuló és teszt adatok alapján is. Az adatvizualizáció során arra koncentráltam, hogy a tényleges és előre jelzett értékek viszonya jól látható és könnyen értelmezhető legyen. A felhasznált módszerek Pythonban való implementálására az *1. ábrán* mutatok be egy példát.

1. ábra: Példa az alkalmazott módszerek implementálására Python kóddal

(Forrás: saját szerk.)

```
# könyvtárak importálása
import pandas as pd                              # adatok beolvasása
from statsmodels.tsa.holtwinters import
     SimpleExpSmoothing,
     ExponentialSmoothing                        # modellek
import plotly.graph_objects as go                # adatvizualizáció
from sklearn.metrics import
     mean_squared_error,
     mean_absolute_error,
     mean_absolute_percentage_error              # modellek kiértékelése

# dataframe létrehozása .csv fájlból
data = pd.read_csv('filename.csv')

# felosztás tanuló és teszt adatokra
data_train = data.iloc[:22]
data_test = data.iloc[22:]

# adatvizualizáció (függvénnyel meghívható az egyes modellekre)
def plot_func(train, forecast: list[float], title: str) -> None:
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=data_train[PERIÓDUS],
             y=data_train[MENNYISÉG], name='Tanuló adatok'))
    fig.add_trace(go.Scatter(x=data_test[PERIÓDUS],
             y=data_test[MENNYISÉG], name='Teszt adatok'))
    fig.add_trace(go.Scatter(x=data_train[PERIÓDUS],
             y=train, name='Tanuló előrejelzés'))
    fig.add_trace(go.Scatter(x=data_test[PERIÓDUS],
             y=forecast, name='Teszt előrejelzés'))
    fig.show()

# modell illesztése
model_tes = ExponentialSmoothing(data_train[MENNYISÉG],
             trend='add',
             seasonal='add',
             seasonal_periods=11).fit(optimized=True)

# előrejelzés
forecasts_tes = model_tes.forecast(len(data_test))
tes_train = model_tes.fittedvalues

# függvény meghívása az egyik modellre
plot_func(tes_train, forecasts_tes, 'Háromszoros exponenciálsimítás')

# modell kiértékelése
print('MAE: ', mean_absolute_error(data_test[MENNYISÉG], forecasts_tes))
print('RMSE: ', mean_squared_error(data_test[MENNYISÉG], forecasts_tes,
             squared=False))
print('MAPE: ', mean_absolute_percentage_error(data_test[MENNYISÉG],
             forecasts_tes))
```

4 Eredmények és értékelésük

Ebben a fejezetben egy-egy minta adatsor felhasználásával összehasonlítom a 2. fejezetben tárgyalt előrejelzési módszerek eredményességét egymáshoz képest. Részletesen kifejtem, hogy miért kaptam az adott eredményt, melyik modell a legjobb és legrosszabb, illetve ennek lehetséges okait.

1. példa:
Egy motoros járművek gyártásával foglalkozó cég havi bontású eladási adatait alapul véve hasonlítom össze a tényleges és becsült eladásokat. Az adatsoron belül a motorkerékpárokra szűrtem, amelyeknek a havi eladott darabszámát vizsgálom. Az előrejelzéseket a 2005. Január időszakra hasonlítom össze.
(*Forrás:* https://www.kaggle.com/datasets/kyanyoga/sample-sales-data)

Az 1. példa adatainak első öt sorát a *3. táblázat* mutatja be. A táblázat az eladás évét, negyedévet, hónapot és az eladott mennyiséget tartalmazza.

3. táblázat: 1. példa adatainak bemutatása
(Forrás: https://www.kaggle.com/datasets/kyanyoga/sample-sales-data, saját szerk.*)*

	Ev	Negyedev	Honap	Periodus	Mennyiseg
0	2003	1	2	2003-2	229
1	2003	1	3	2003-3	170
2	2003	2	4	2003-4	212
3	2003	2	5	2003-5	251
4	2003	2	6	2003-6	23

Az adatsor alapvető statisztikáit a 4. táblázatban foglaltam össze. Ezt Pythonban a „Mennyiseg” oszlopra szűrve a *DataFrame.describe()* függvénnyel hoztam létre. A *4. táblázat* fentről lefelé tartalmazza a vizsgált időszakok számát, az eladott mennyiségek átlagát, szórását, minimum értékét, első, második (medián), és harmadik kvartilisát, illetve maximum értékét. A statisztikai értékek alapján elmondható, hogy nagy az eladott mennyiség terjedelme (minimum és maximum érték különbsége), valamint, hogy várhatóan vannak kiugró értékek, mivel az eladott mennyiség az esetek 75%-ában 493 alatti értéket vesz fel, ami nagyban eltér az 1,403 db maximum értéktől. Ezt a *2. ábra* alátámasztja, amelyen látszik, hogy két kiugró érték is van

az adatsorban. Vizsgálódásom részét képezi egy olyan módszer keresése, amellyel a szezonálisan kiugró értékeket is előre tudjuk jelezni, ezért a kiugró értékeket nem törlöm ebben az esetben.

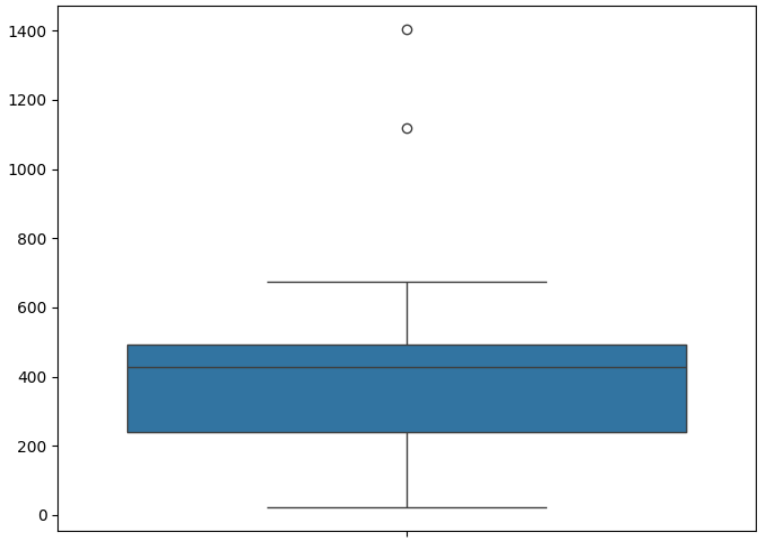
4. táblázat: 1. példa adatainak statisztikái

(Forrás: saját szerk.)

	Mennyiseg
count	27
mean	431.96
std	290.52
min	23
25%	240
50%	427
75%	493
max	1,403

2. ábra: Kiugró értékek vizsgálata az 1. példa esetében

(Forrás: saját szerk.)



2.5.1 Az utolsó értékre alapozott előrejelzési (naív) eljárás

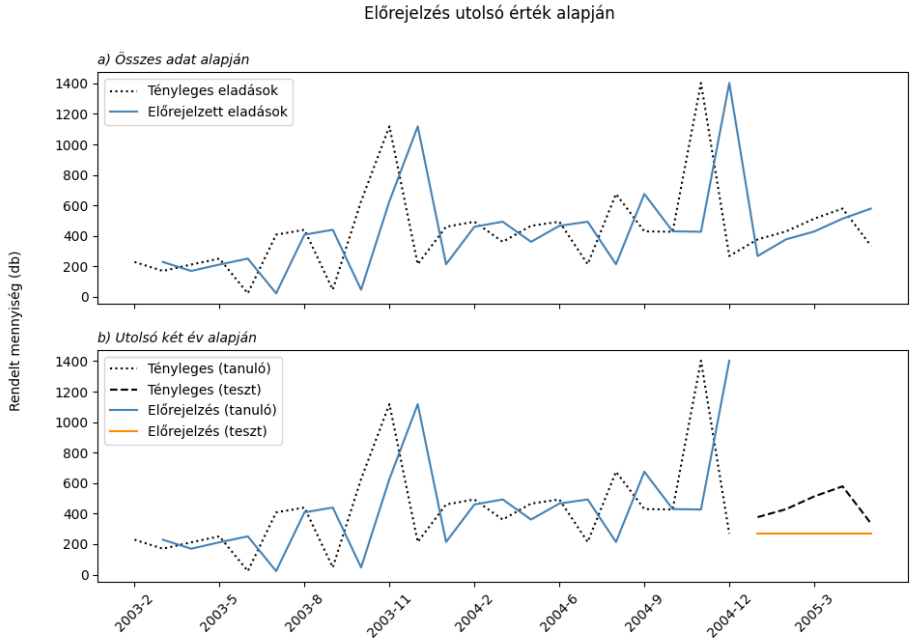
Ennél a módszernél, az előre jelzett érték megegyezik az előző megfigyelt értékkel, tehát egyszerűen létrehoztam egy új oszlopot, amely az egy sorral fentebbi tényleges értékeket tartalmazza. Ezek az értékek lesznek az előrejelzések a következő időszakra. Ezzel a módszerrel csak a következtő értéket lehet megmondani, amely alapján minden azt követő időszakra vonatkozó érték megegyezik egymással. Ezzel a módszerrel a következő hónapra előre jelzett mennyiség 268 db, a tényleges 377 db-al szemben.

Vizuálisan ábrázolva jól látszik, hogy az előre jelzett értékek mindig egy periódussal el vannak maradva a tényleges értékektől. Ez kifejezetten szembetűnő, amikor szezonálisan változó adatokkal dolgozunk, amit a *3. ábra* jól szemléltet. Külön diagramon ábrázoltam azokat az eseteket, amikor vagy kizárólag a következő értéket jelezzük előre (*a) diagram*) , illetve amikor az idősort tanuló és teszt adatokra osztjuk, és az előrejelzésünket kizárólag a tanuló adatokra építjük (*b) diagram*).

Összességében elmondhatjuk, hogy ez a módszer nem kifejezetten alkalmas a mi idősorunk alapján előrejelzések készítésére, az egyenletesebb keresleti trendet mutató periódusok kivételével. A kiugró értékeknél jelentkező tényleges és becsült értékek közötti eltérések sokat rontanak a modell teljesítményén.

3. ábra: Utolsó értékre alapozott eljárás

(Forrás: saját szerk.)



2.5.2 Átlagoló előrejelzési eljárás

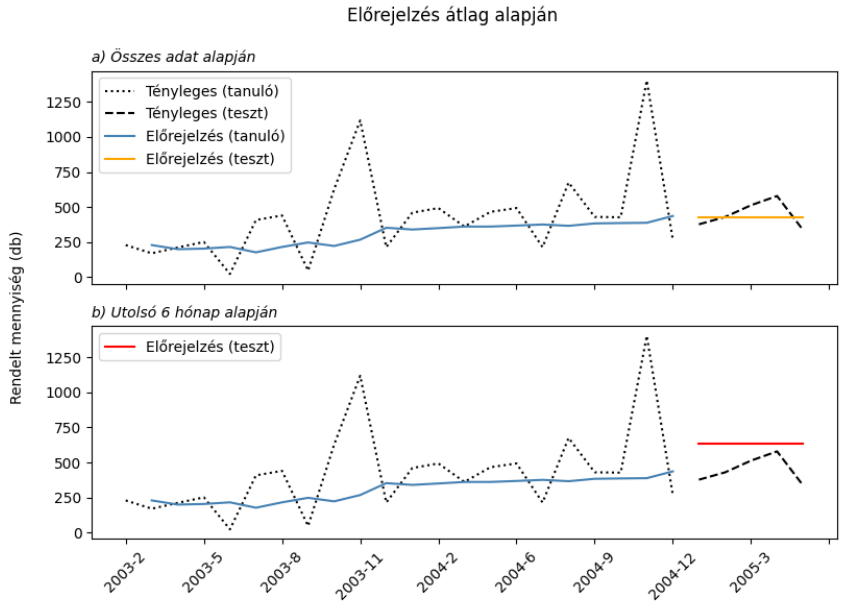
Mivel összesen majdnem három évnyi adat áll a rendelkezésre, átlagot számoltam az összes, illetve csak az utolsó hat hónap adatiból is. Ezt Pythonban a *Numpy* könyvtár segítségével, azon belül pedig a *mean()* függvény segítségével oldottam meg. Így a következő hónapra az előrejelzésünk az összes adat átlagával számolva 429 db, az utolsó hat hónappal számolva pedig 632 db.

Ebben az esetben is elvégeztem a tanuló és teszt adatokra való felbontást 1:4 arányban. A teszt adatokra külön-külön számoltam átlagot az összes adat (*a) diagram*), illetve az utolsó hat hónap adata (*b) diagram*) alapján. Az utóbbi módszer eredményeképpen alapvetően magasabb előrejelzéseket kapunk, mivel az utolsó hat hónapban megfigyelt értékek között van kiugró értékünk, amely jelentősen torzítja az átlagot. A mi esetünkben érdemes az alacsonyabb értékkel számolni, ha figyelembe vesszük a szezonális periódusok hosszát.

Következik tehát, hogy átagszámításnál figyelembe kell vennünk az átlagoláshoz használt adatmennyiséget, és annak időben való elhelyezkedését, mivel a régebbi adatok veszíthetnek relevanciájukból, illetve szezonalitást mutató idősorok esetében a vizsgált időszak hossza és az időben való elhelyezkedése is befolyásolja az eredményt. Mivel az adatok szezonális trendet mutatnak, ezzel a módszerrel nem tudtam pontos hosszútávú előrejelzést adni. Az előbbi megállapításokat a *4. ábra* jól vizualizálja.

4. ábra: Előrejelzés átlag alapján

(Forrás: saját szerk.)



2.5.3 Mozgóátlagokra alapozott előrejelzési eljárás (SMA)

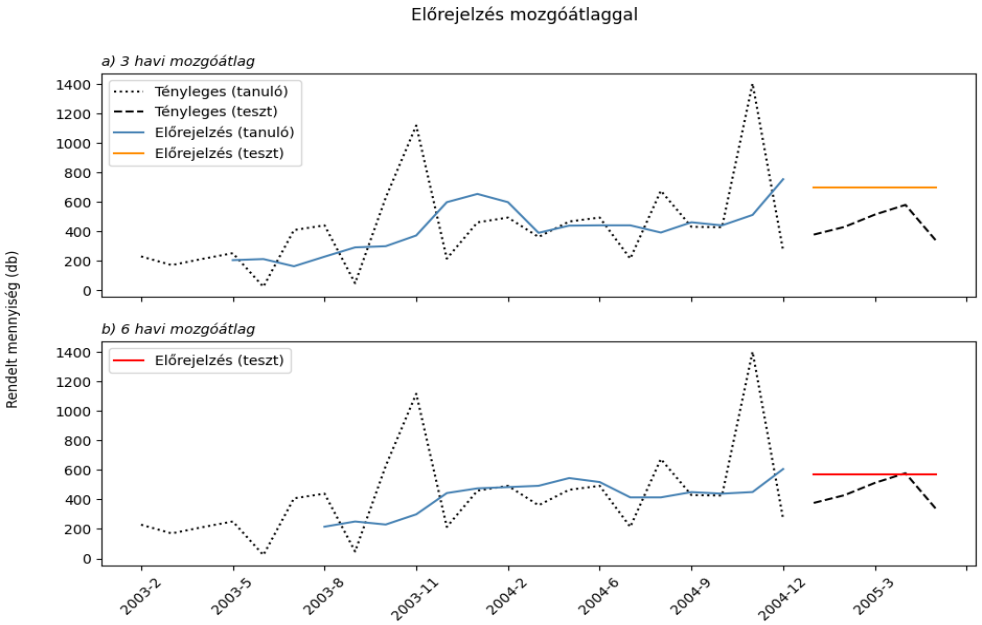
Mozgóátlag használatakor meg kell határoznunk az átlag számításánál használt periódusok hosszát. Pythonban való modellezéshez ebben az esetben is elegendő a *Pandas*, ezen belül pedig a *rolling()* függvény, amelynek argumentumaként meg tudjuk adni a mozgó periódus hosszát, illetve *Numpy* és *Matplotlib* könyvtárak használata. Mivel a periódusok számának meghatározása tapasztalati úton történik, és a legtöbb esetben 3 és 8 közé esik, ebben a sorozatban kerestem egy rövidebb és hosszabb időszakot (Benkő, 2018). Három és hat havi mozgóátlagot számoltam, hogy össze tudjam vetni a kapott eredményeket. A hosszabb periódusok átlagolása egyenletesebb előrejelzést tesz lehetővé, míg a rövidebb periódusok több részletet biztosítanak, de érzékenyebbek a zajra.

Három havi mozgóátlag alapján 699 db az előrejelzés az eladásokat illetően, míg hat hónappal számolva ez csak 570 db, tehát ebben az esetben is lényeges a számítás alapjául szolgáló periódus hossza.

Az *5. ábrán* szemléltetem, hogy a rövidebb időszakra számított mozgóátlag alkalmazása valamivel jobban követi a tényleges eladási tendenciát. Jól látszik, hogy az előre jelzett értékek el vannak csúszva a tényleges értékekhez képest, és a modell nem reagál jól az adatok hirtelen változásaira.

5. ábra: Előrejelzés mozgóátlaggal

(Forrás: saját szerk.)



Előrejelzési eljárás súlyozott mozgóátlaggal (WMA)

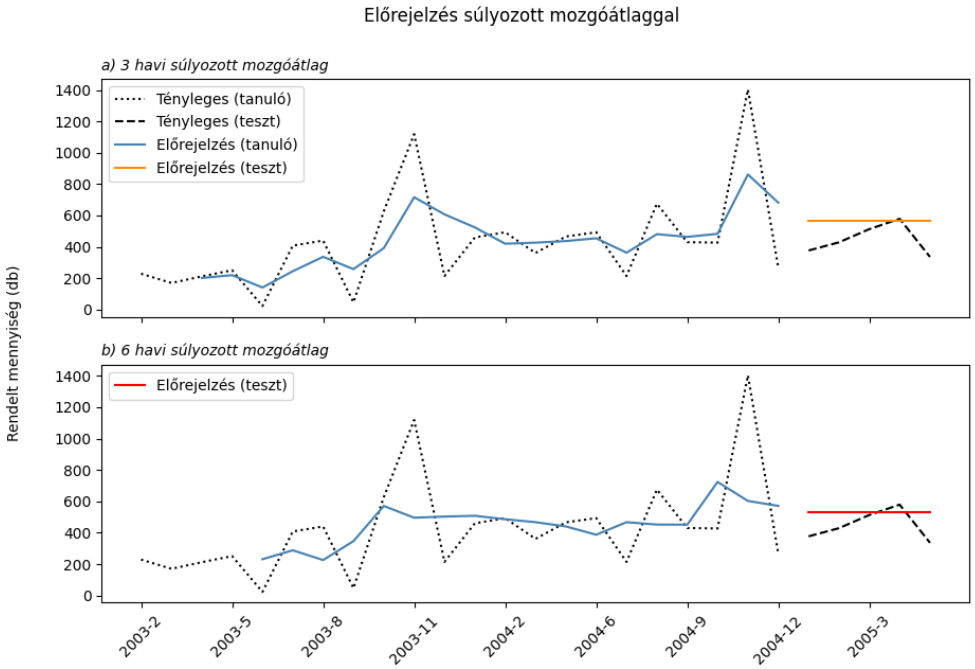
A súlyozott mozgóátlag kiszámításához szintén a *numpy.rolling()* függvényt használtam, de a sima mozgóátlaghoz képest előre meghatároztam a három és hat hónapra vonatkozó súlyokat, amelyek lineárisan növekednek az időben frissebb adatok felé haladva.

A *6. ábrán* azt mutatom be, hogy a súlyozott mozgóátlag összességében jobban illeszkedik a tényleges eladási trendhez, és az előre jelzett értékek az x tengely mentén is alapvetően jobban követik a tényleges eladási adatokat. A súlyozott mozgóátlag módszerével jobban le tudjuk fedni a keresletben fellépő fluktuációt, ugyanakkor ez a kiugróan magas eladások után, a ténylegesnél magasabb előrejelzést fog eredményezni, tekintve, hogy az utolsó érték nagyobb súllyal kerül bele a kalkulációba.

Három havi súlyozott mozgóátlaggal számolva 569 db a következő hónapra előre jelzett eladásunk, míg ugyanez hat hónappal számolva csak 528 db, ami közelebb van a tényleges 377 db-hoz. A teszt adatok tekintetében tehát az utóbbi módszerrel jobb az előrejelzésem, viszont összességében az előbbi modell jobban illeszkedik a tanuló adatokra.

6. ábra: Előrejelzés súlyozott mozgóátlaggal

(Forrás: saját szerk.)



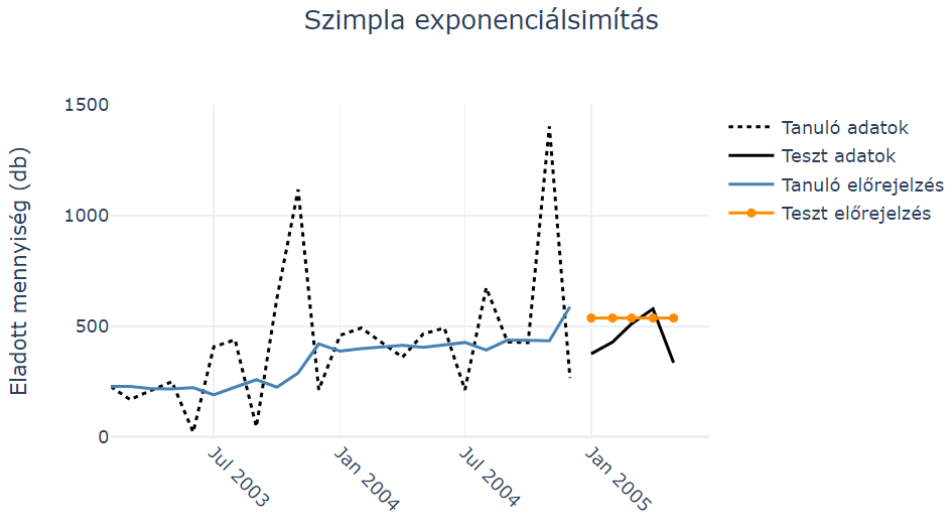
2.5.5 Előrejelzési eljárás exponenciálissimítással (ES)

Mindhárom módszer esetében 0.5 értékre állítottam be az egyes simítótényezőket (*alfa, béta, gamma*) a modellek megalkotásához, hogy a lehető legjobb eredményt kapjam, ugyanakkor megfigyelhetőek legyenek az egyes módszerek sajátosságai is.

A *7. ábra* jól szemlélteti, hogy a szimpla exponenciális simítás rosszul kezeli a szezonálisan változó idősorokat, és ilyen esetekben az előrejelzések folyamatos csúszásban vannak a ténylegesen mért értékekhez képest, mert a modell nem képes a szezonálisan kiugró értékeket megbízhatóan kezelni.

7. ábra: Szimpla exponenciálsimítás

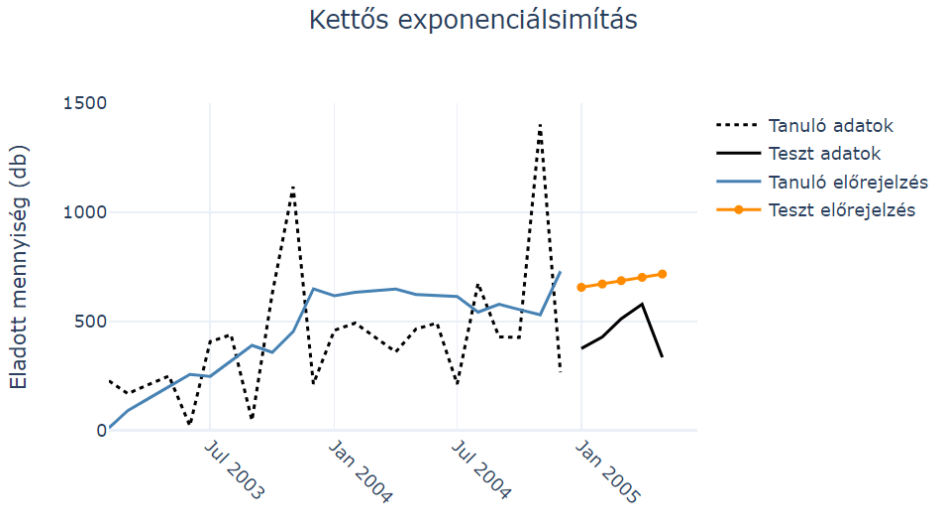
(Forrás: saját szerk.)



A *8. ábrán* megfigyelhetjük, hogy a kettős exponenciális simítás modell jobban lefedi a kiugró értékeket, viszont még mindig jelentős elcsúszás tapasztalható a tényleges értékekhez képest. Ez a modell nem illeszthető jól a mi adatainkra, hiszen nem figyelhető meg alapvető trend, így a *béta* simítótényező beépítése az egyenletbe inkább ront a végeredményen. Ez jól látszik a teszt előrejelzésen is, ami a ténylegesnél jóval magasabb értékeket eredményez.

8. ábra: Kettős exponenciálsimítás

(Forrás: saját szerk.)



A *9. ábrán* első ránézésre egyértelmű, hogy ez a modell sokkal jobban működik az én szezonálisan változó adataimra illesztve. A modell sokkal jobban lefedi a kiugró értékeket, továbbá az előző simítási módszereknél tapasztalt lemaradás sem figyelhető meg ugyanakkora mértékben. A modellek pontosságát számosítva itt kaptam a legkisebb hibákat, tehát ez a modell lesz várhatóan a legmegfelelőbb előrejelzésre.

9. ábra: Háromszoros exponenciális simítás

(Forrás: saját szerk.)

Pythonban a Statsmodel könyvtár segítségével hoztam létre az eddig tárgyalt modelleket. A *statsmodels.tsa.SimpleExpSmoothing()* függvénnyel illesztettem az adatokra a szimpla modellt, illetve határoztam meg az *alfa* simítótényező értékét. Az *ExpinentialSmoothing()* függvény alkalmas a kettős és háromszoros exponenciálsimítás modellek megalkotásához. Ezeknél a trend és szezonális komponens simítótényezőjét is meg tudjuk határozni.

Az egyes módszerekkel eltérő előrejelzéseket kaptam. A szimpla módszerrel 538 db, a kettőssel 657 db, a háromszorossal pedig 588 db a 2005. január időszakra várt eladás.

3.1 Idősor-elemző módszerek kiértékelése mérőszámokkal

A modellek kiértékelése során elsősorban az egymáshoz viszonyított eredményekre koncentráltam. A 3.1 fejezetben tárgyalt és *1. példán* keresztül bemutatott előrejelzési eljárások kiértékelési metrikáit az *5. és 6. táblázat* foglalja össze. A modelleket külön-külön értékeltem tanuló és teszt adatok alapján. A módszereket a táblázat az átlagos abszolút hiba értéke alapján növekvő sorrendben tartalmazza, tehát a legjobb eredmény van legfelül. Az egymáshoz viszonyított eredményeket továbbá színkódolással is jelöltem, ahol a zöld jó, a sárga közepes, a piros szín pedig rossz eredményt jelent. Az eredmények három kategóriáját egyenlő arányban osztottam fel, tehát az alsó harmadban végzett eredményeket tekintem rossznak, a középső harmadot közepesnek, a felsőt pedig jónak.

5. táblázat: Idősor-elemző eljárások kiértékelése tanuló adatok alapján

(Forrás: saját szerk.)

Idősor-elemzés modell	**MAE**	**RMSE**	**MAPE**
Háromszoros exponenciálsimítás	120	120	0.56
Súlyozott mozgóátlag (3 hónap)	182	237	0.92
Szimpla exponenciálsimítás	206	320	0.97
Átlag (összes adat)	219	334	0.96
Súlyozott mozgóátlag (6 hónap)	222	301	1.27
Mozgóátlag (6 hónap)	245	365	0.70
Kettős exponenciálsimítás	256	326	1.38
Mozgóátlag (3 hónap)	266	365	0.85
Utolsó érték	324	460	1.63

A felhasznált módszerek közül kimagaslóan jól teljesített a háromszoros exponenciális simítás. Az összes tárgyalt módszer közül ez az egyetlen, amely figyelembe veszi a szezonalitást, amit a pontossági mutatók is igazolnak. A három hónappal kalkulált súlyozott mozgóátlaggal is jó eredményt kaptam. Az eredmény jól mutatja az adatok időrendben való súlyozásának jelentőségét. Mivel csak egy-egy hónapban figyelhető meg szezonális kiugrás, három hónap átlagolásával a valósághoz közelebbi eredményt kaptam, mint abban az esetben, ha ugyanezt hat (vagy több) hónappal számoltam volna.

A szimpla exponenciálsimítás közepes eredményt produkált, ami várható volt a modell egyszerűsége miatt. Ehhez képest a kettős exponenciálsimítással jelentősen rosszabb eredményt kaptam, tehát az *8. ábra* alapján tett megállapításom igaznak bizonyult, és a modell valóban rosszabbul teljesít a trend simítótényező növelésével.

A sima átlagoló módszer szintén közepes eredményt produkált. Ezt alapvetően annak tulajdonítható, hogy az átlagok jelentősen torzulnak a kiugró értékek jelenléte miatt. Ez a módszer tehát sokkal alkalmasabb egyenletes, kiugró értékeket nem tartalmazó idősorok elemzésére.

A sima mozgóátlagokkal sem értem el közepesnél jobbnak mondható eredményt. Mind a két esetben közepes vagy rossz MAE és RMSE értékeket kaptam, viszont a MAPE értékét tekintve mindkettő esetében viszonylag jónak bizonyul a modell. Ez a használt mérőszámok jellegéből adódik, mivel a MAE és RMSE az egyes különbségek abszolút, illetve négyzetes értékével dolgoznak, a MAPE viszont az előre jelzett értékekhez százalékosan viszonyít, ezért utóbbi értéke nagyban függ attól, hogy mekkora számokkal dolgozunk.

Az utolsó értékre alapozott előrejelzési eljárás teljesített a legrosszabbul, ami jól mutatja, hogy ez a módszer egyáltalán nem alkalmas a szezonális adatok kezelésére, és ez megmutatkozik mindegyik mérőszámban is, amelyek ennél a módszernél a legmagasabbak az összes vizsgált módszer közül.

6. táblázat: Idősor-elemző eljárások kiértékelése teszt adatok alapján

(Forrás: saját szerk.)

Idősor-elemzés modell	MAE	RMSE	MAPE
Átlag (összes adat)	76	91	0.17
Súlyozott mozgóátlag (6 hónap)	90	104	0.22
Háromszoros exponenciálsimítás	96	125	0.25
Súlyozott mozgóátlag (3 hónap)	101	120	0.25
Szimpla exponenciálsimítás	108	127	0.28
Mozgóátlag (6 hónap)	134	164	0.35
Utolsó érték	179	200	0.38
Átlag (utolsó 6 hónap)	185	205	0.47
Mozgóátlag (3 hónap)	208	220	0.50
Kettős exponenciálsimítás	240	256	0.60

A tanuló adatok alapján mért eredmények után összehasonlítottam a teszt adatokra kalkulált mérőszámokat is. A *6. táblázat* alapján látható, hogy különböző eredményeket kaptam a tanuló adatokhoz képest, ami alapvetően abból következik, hogy a két adathalmaz alapján egészen más a kereslet eloszlása.

Több olyan modellem is van, ami jobban illeszkedik a teszt adatokra, mint a tanuló adatokra. Ez rávilágít arra, hogy a különböző modellek használatával a kereslet változásának függvényében eltérő eredményeket kaphatunk, tehát a megfelelő modell kiválasztásához szükséges az alapvető trendek és esetleges szezonalitás ismerete.

A teszt adatokhoz viszonyítva sokkal jobb eredményt kaptam az összes adat alapján való átlagolás és a súlyozott mozgóátlagok alapján, ami azzal magyarázható, hogy ezek a módszerekkel alapvetően egyenletesebb és alacsonyabb előrejelzést kaptam, ami jobban lefedi a 2004. december utáni keresletet.

Az utolsó érték és kettős exponenciálsimítás módszere a teszt adatok vonatkozásában is igen gyengén teljesített, megerősítve a korábban tett megállapításomat, miszerint ezek a módszerek egyáltalán nem alkalmasak szezonalitást mutató adatok elemzésére.

A legjobb illesztést a háromszoros exponenciálsimítással értem el, mind a tanuló és teszt adatok tekintetében, tehát az a megállapításom, hogy szezonalitást mutató idősor elemzésére ez a leghatékonyabb módszer.

A *7. táblázat* azt mutatja be, hogy az egyes eljárásokkal előre jelzett értékek mennyire térnek el a 2005. januári aktuális értékektől. Amennyiben a becslésünk pontosságát csak a következő hónap vonatkozásában szeretném meghatározni, a legjobb előrejelzést az összes adat

átlagolásával kapom meg. Ez természetesen nem jelenti azt, hogy minden esetben ez a módszer lesz a leghatékonyabb.

7. táblázat: Előrejelzések pontossága 2005. január időszakra

(Forrás: saját szerk.)

Idősor-elemzés modell	**Előrejelzett érték (2005. Január)**	**Tényleges értéktől való eltérés**
Utolsó érték	268	-109
Átlag (összes adat)	429	52
Súlyozott mozgóátlag (6 hónap)	528	151
Szimpla exponenciálsimítás	538	161
Súlyozott mozgóátlag (3 hónap)	569	192
Mozgóátlag (6 hónap)	570	193
Háromszoros exponenciálsimítás	588	211
Átlag (utolsó 6 hónap)	632	255
Kettős exponenciálsimítás	657	280
Mozgóátlag (3 hónap)	699	322

2. példa:

A Wish webáruházról származó, nyári termékek adatait tartalmazó adatsort alapul véve keresek kapcsolatot a termékek vásárlói értékelése és az eladott darabszám között, amely alapján előrejelzést próbálok adni a várható eladásokról. A regresszió-analízis során a 3.5 értékeléshez tartozó eladásokat próbálom meg megbecsülni.

(Forrás: jfreex/summer-products-and-sales-performance-in-e-commerce-on-wish | Workspace | data.world)

A 2. példa esetében is megnéztem az alapvető statisztikákat, amit a *8. táblázat* tartalmaz. A táblázatból kiolvasható, hogy összesen 1,573 értékeléssel dolgoztam. Ebben az esetben is kiolvasható a táblázatból, hogy kiugró értékek vannak az adatsorban, tekintve a harmadik kvartilis és maximum érték közötti jelentős különbséget.

8. táblázat: 2. példa adatainak statisztikái

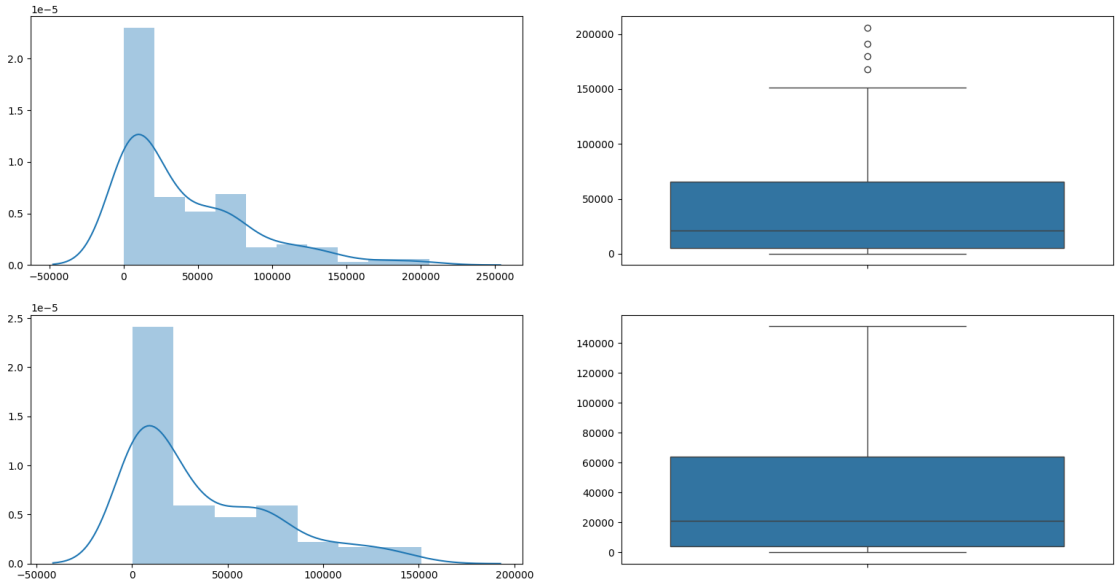
(Forrás: saját szerk.)

	Mennyiseg
count	1,573
mean	4,339.01
std	9,356.54
min	1
25%	100
50%	1,000
75%	5,000
max	100,000

Az adatok betöltése után megvizsgáltam az eloszlás sűrűségfüggvényét. Azt tapasztaltam, hogy az eloszlás jobbra ferde, vagyis a függvény jobb oldalán magasabb értékeket látunk. Emiatt a kiugró értékek keresésekor az interkvartilis terjedelem (IQR)[16] alapján szűrtem, mert ez a módszer figyelembe veszi az adatok terjedelmét és az eloszlás ferdeségét is. A kiugró értékeket dobozdiagramon is ábrázoltam szűrés előtt és után is, amit a *10. ábra* mutat be. A *11. ábrán* ennek a módszernek a Python kóddal való leírása található.

10. ábra: Eloszlásfüggvény és kiugró értékek vizsgálata

(Forrás: saját szerk.)



[16] Az angol „Inter Quartile Range” kifejezésből.

11. ábra: Kiugró értékek szűrése Pythonban interkvartilis terjedelem alapján

(Forrás: saját szerk.)

```
import pandas as pd


df = pd.read_csv('filename.csv')

# percentilisek meghatározása
percentile25 = df[MENNYISÉG].quantile(0.25)
percentile75 = df[MENNYISÉG].quantile(0.75)

# alsós és felső limit meghatároPzása
upper_limit = percentile75 + 1.5 * (percentile75-percentile25)
lower_limit = percentile25 - 1.5 * (percentile75-percentile25)

df[df[MENNYISÉG] > upper_limit]
df[df[MENNYISÉG] < lower_limit]

# szűrt dataframe
new_df = df[df[MENNYISÉG] < upper_limit]
```

2.6.1 Lineáris Regresszió

A lineáris regresszió modelljének megalkotásához a Python *Scikit-learn* könyvtárát, azon belül pedig a *LinearRegression()* függvényt használtam. Az idősor-analízis eljárásokhoz hasonlóan olvastam be az adatokat, majd szintén a *Scikit-learn* könyvtáron belül elérhető *train_test_split()* függvénnyel felosztottam az adatokat 1:4 arányban tanuló és teszt adatokra. Az értékeléseket minden modell esetében 2.5 és 4.5 érték között szűrtem, mivel ebben a sávban történt az eladások 99.84%-a.

Az eredményeket a *12. ábra* mutatja, amin jól látható, hogy egyenes illesztésével nem tudjuk jól lefedni az adatpontokat, tehát a modellem pontossága feltételezhetően nem lesz kielégítő. Ezt a *Scikit-learn* könyvtáron belül elérhető *r2_score()* függvény is igazolja, amely alapján a determinációs együttható értéke a tanuló adatok alapján 0.22, míg a teszt adatok alapján csupán 0.03, ami alacsony korrelációra enged következtetni a termékek értékelése és az eladási adatok között.

Szintén a *Scikit-learn* könyvtár segítségével kiszámoltam a lineáris függvény meredekségét és tengelymetszetét az *intercept_* és *coef_* függvényekkel. Ezeket felhasználva írtam fel az egyenes egyenletét, amiből kiszámoltam a 3.5 értékeléshez tartozó becsült eladott darabszámot.

Így a tényleges 16,610 darabhoz képest az előrejelzésem 33,783 db lett, ami jóval meghaladja azt.

$$y = 36{,}798x - 95{,}010 \quad (1)$$

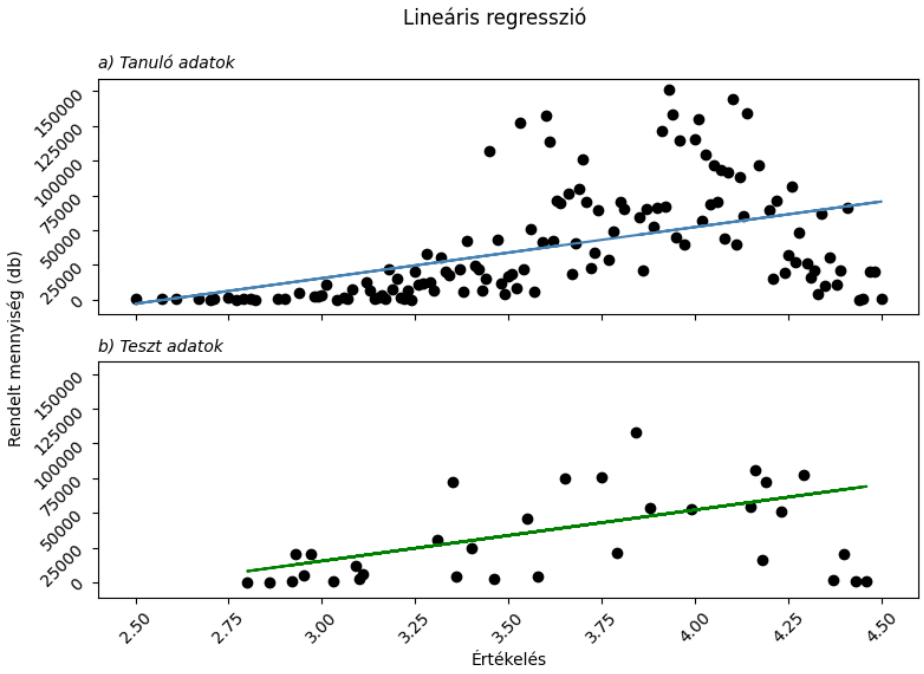
$$y = (36{,}798 * 3.5) - 95{,}010 \quad (2)$$

$$y = 33{,}783 \quad (3)$$

A lineáris modell kapcsán létrehoztam Pythonban a *Seaborn* könyvtár *residplot()* függvényével egy reziduális diagramot is az illeszkedési hibák szemléltetésére (*13. ábra*). Ideális esetben a reziduális diagram véletlenszerű eloszlást mutat, de ebben az esetben jól látszik, hogy ez a feltétel nem teljesül. A hibák szórása láthatóan nem állandó, ami heteroszkedaszticitásra utal. „A heteroszkedaszticitás általában egy modellben a szórások különbözőségét jelenti" (Hunyadi, 2006). Ez is arra utal, hogy a lineáris függvény nagyon rosszul illeszthető a felhasznált adatokra.

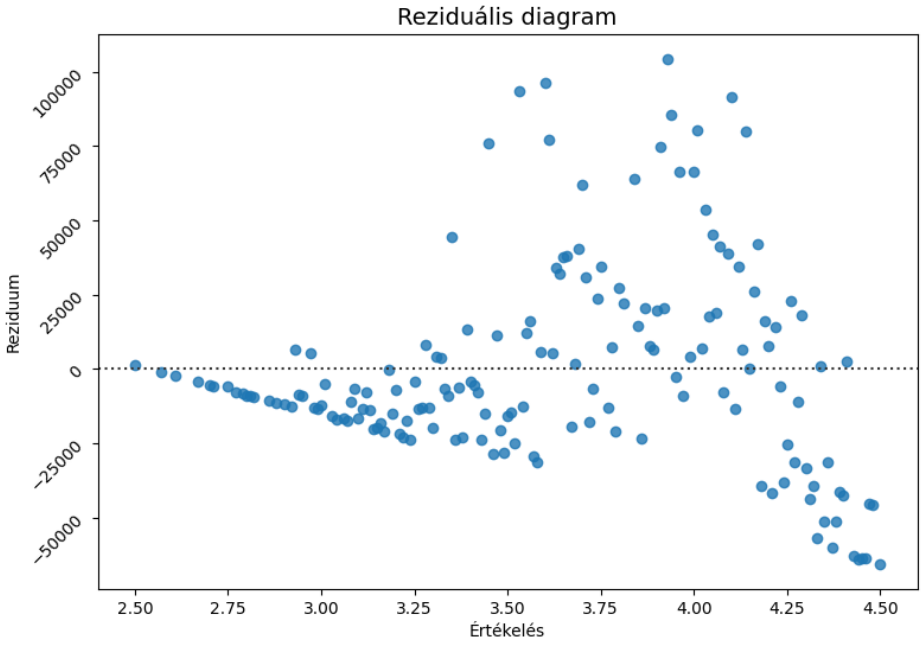
12. ábra: Lineáris regresszió

(Forrás: saját szerk.)



13. ábra: Reziduális diagram

(Forrás: saját szerk.)



2.6.2 Kvantilis Regresszió

A kvantilis regresszió modell esetében a tanuló és teszt adatokat ezúttal is 1:4 arányban osztottam fel. Meghatároztam a predikált értékeket medián (Q_2) értékre, illetve az első (Q_1) és harmadik (Q_3) kvartilis értékére is. A modell létrehozásánál figyelembe vettem az alábbi kvantilis veszteségfüggvényt, ahol *alfa* a kvantilis értékét jelöli.

$$\ell(y, \hat{y}) = \begin{cases} \alpha(y - \hat{y}), & \hat{y} \leq y \\ (1 - \alpha)(\hat{y} - y), & \hat{y} > y \end{cases}$$

A harmadik kvartilis számításánál ez azt jelenti, hogy a veszteségfüggvény háromszor jobban bünteti az alábecsült előrejelzéseket, mint a túlbecsülteket, tehát a modell sokkal kritikusabb az alulbecsült hibákkal szemben, és emiatt gyakrabban jelez előre magasabb értékeket. Az illesztett modellünk emiatt átlagosan az esetek 75%-ban túlbecsüli az eredményeket, míg 25%-ban alulbecsüli azt.

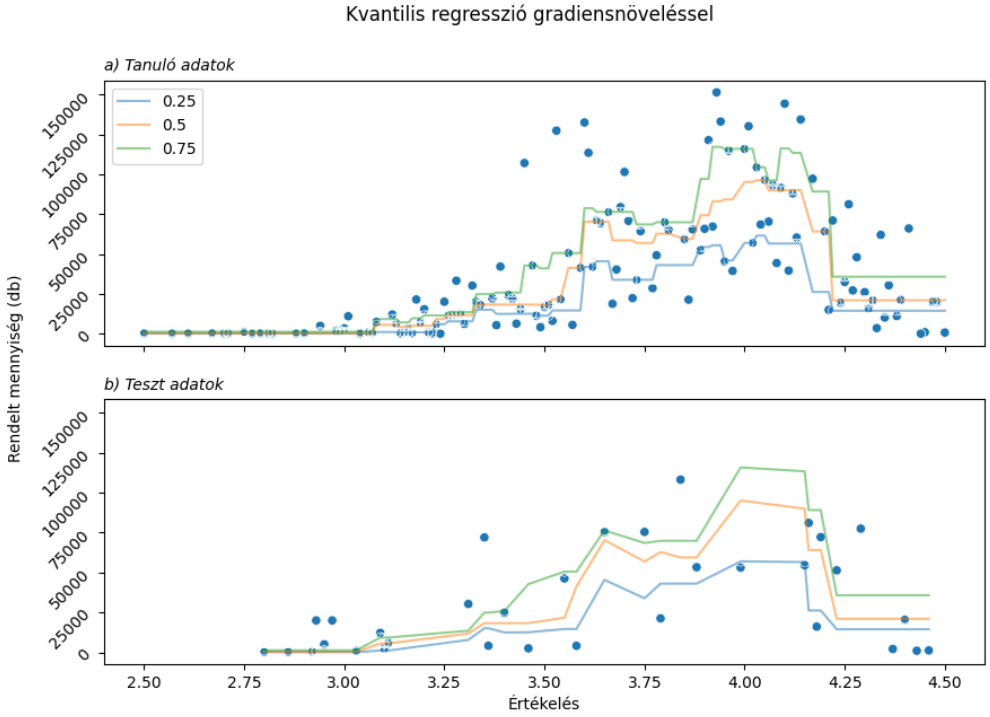
A modell megalkotásánál a gradiensnövelés (gradient boosting) módszerét is alkalmaztam. A gradiensnövelés egy olyan gépi tanulási módszer, amely során egymást követően újabb és újabb döntési fákat hozunk létre. A gradiensnöveléssel minden ismétlésnél új modellt

illesztünk, ami figyelembe veszi az addigi hibák összességét, és ezek alapján minimalizálja a hibák különbségét, ezáltal finomhangolva a következő előrejelzéseket (Natekin és Knoll, 2013).

Pythonban erre a célra a *LightGBM* könyvtárat használtam, azon belül az *LGBMRegressor()* függvényt, ahol kvantilis regressziót határoztam meg feladatként a különböző *alfa* értékekre. Mediánnal kalkulálva az előre jelzett eladás 18,024 db a 3.5 értékelésű termékekre, ami nagyon közel van a tényleges eladási adathoz. Az eredményeket a *14. ábrán* szemléltetem. A módszer Pythonban való kivitelezését a *15. ábrán* írom le.

14. ábra: Kvantilis regresszió

(Forrás: saját szerk.)



15. ábra: Gradiensnövelés Pythonnal (kvantilis regresszió)

(Forrás: saját szerk.)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from lightgbm import LGBMRegressor


data = pd.read_csv('filename.csv')

# tanuló és teszt adatokra bontás 1:4 arányban
data_train, data_test = train_test_split(data, test_size=0.20,
      random_state=42)

# LGBM modell: 3 különböző kvartilissal végigiterálva
models_train = {}
data_pred_train = data_train.copy()
for alpha in [0.25, 0.5, 0.75]:
    model = LGBMRegressor(objective='quantile', alpha=alpha)
    model.fit(x_train, y_train)
    data_pred_train[str(alpha)] = model.predict(x_train)
```

2.6.3 Tartóvektor Regresszió (SVR)

A tartóvektor regresszió modellemhez a *Scikit-learn* könyvtár *SVR()* függvényét használtam. A modellemnél radiális bázis függvényt használtam (kernel='RBF'), mivel ez a függvény kifejezetten alkalmas nemlineáris problémák megoldására. A modellnek három további paramétere van. A *C* érték szabályozza a túlillesztés és túlágazás közötti egyensúlyt. Az *epszilon* értéke határozza meg a toleranciasávot a modell kimenete és az aktuális célértékek között. A *gamma* értéke a függvény hatékonyságát befolyásolja. Nagyobb *gamma* érték összetettebb modellt eredményez, ami túlillesztéshez vezethet (Lessmann et al., 2005).

A modell paramétereit a *Scikit-learn* Python könyvtárban elérhető *GridSearchCV()* függvénnyel határoztam meg, ami egy keresztvalidációs technika. Az algoritmus egy előre meghatározott paraméterhalmaz alapján minden lehetséges kombinációval létrehoz egy modellt, és ezek közül kiválasztja a legjobbat. A modell a 3.5 értékeléshez 35,382 db eladott darabszámot jelez előre, ami az eddigi legmagasabb érték. A *17. ábrán* jól látszik, hogy a lineáris regresszió egyeneséhez képest magasabb az előrejelzésünk. A modell összességében nem illeszkedik jól az adatokra. Jobb illeszkedést magasabb *C,* illetve *gamma* értékekkel tudunk elérni, viszont ebben az esetben növekszik a túlillesztés kockázata.

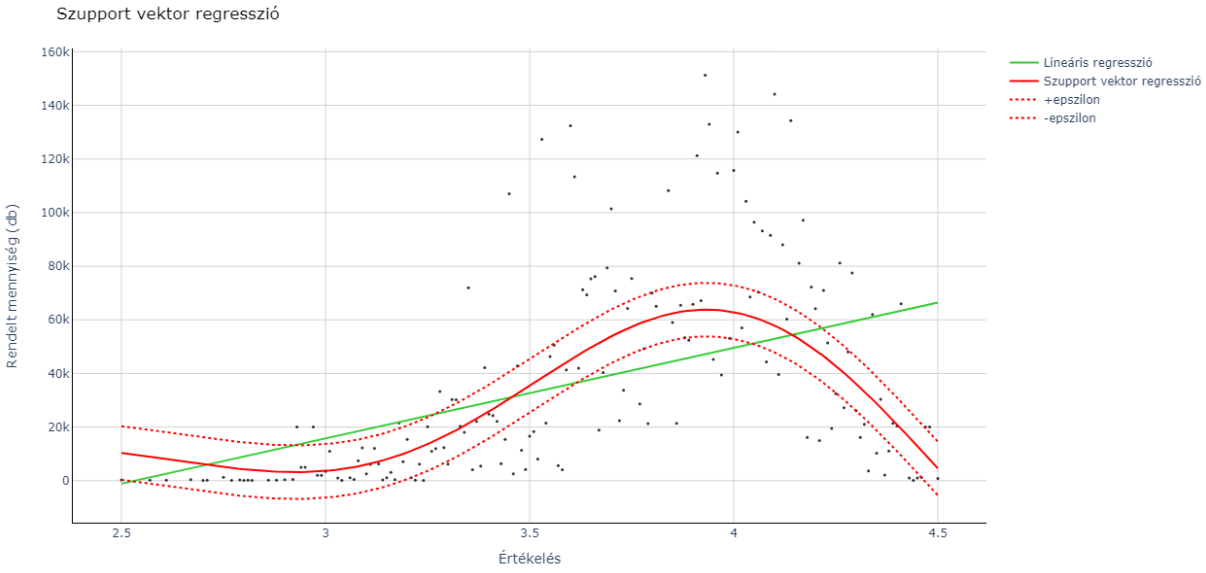
16. ábra: Rácskeresés (grid search) Pythonban

(Forrás: saját szerk.)

```
from sklearn.model_selection import GridSearchCV


# paraméterek listázása a tesztelendő értékekkel
param_grid = {'C': [0.1, 1, 10, 100, 1000, 10000, 100000],
              'gamma': [100000, 10000, 1000, 100, 10, 1, 0.1, 0.01,
                        0.001, 0.0001],
              'epsilon': [1, 10, 100, 1000, 10000, 100000],
              'kernel': ['rbf']}

# rácskeresés SVR modellre implementálva
grid = GridSearchCV(SVR(), param_grid, refit = True, verbose = 3)

# rács illesztése a tanuló adatokra
grid.fit(x_train, y_train)

# legjobb paraméterek kiírása
print(grid.best_params_)
```

17. ábra: Tartóvektor regresszió

(Forrás: saját szerk.)



Döntési fa (decision tree)

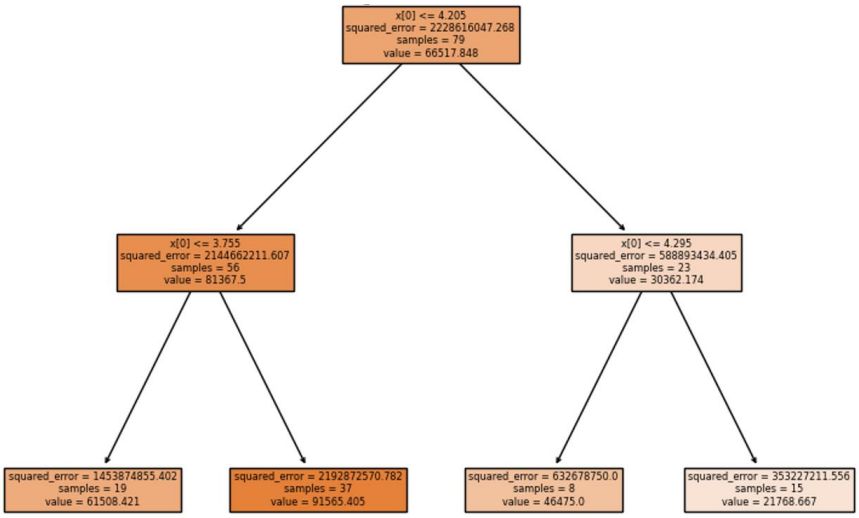
A döntési fa létrehozásához szintén a *Scikit-learn* Python könyvtárat használtam. Ezen belül a *DecisionTreeRegressor()* függvénnyel dolgoztam. A túlillesztés elkerülése végett a fa mélységét ötre korlátoztam. Így megközelítőleg ugyanolyan jól illeszkedik a modell a tanuló

és teszt adatokra, míg ennél kisebb érték esetén jelentősen romlik a modell pontossága, nagyobb érték esetén pedig a teszt adataink sokkal kevésbé illeszkednek.

A döntési fa struktúráját a *18. ábra* szemlélteti. Az áttekinthetőség kedvéért ebben az esetben háromra korlátoztam a fa mélységét. Az ábra jól mutatja, hogy az adott jellemző alapján a fa elágazik, különféle értékek felé haladva. Az algoritmus kiszámolja, hogy az adott osztályozás mennyire járul hozzá az információnyereséghez vagy hibacsökkentéshez, és ez alapján hoz létre újabb ágakat. A fa egészen addig ismétli ezt, amíg nem teljesül egy meghatározott feltétel. Azt is láthatjuk az ábrán, hogy az algoritmus minden csomópontban kiszámolja a négyzetes hibát, ami meghatározó az elágazási folyamatban.

18. ábra: Döntési fa struktúrája

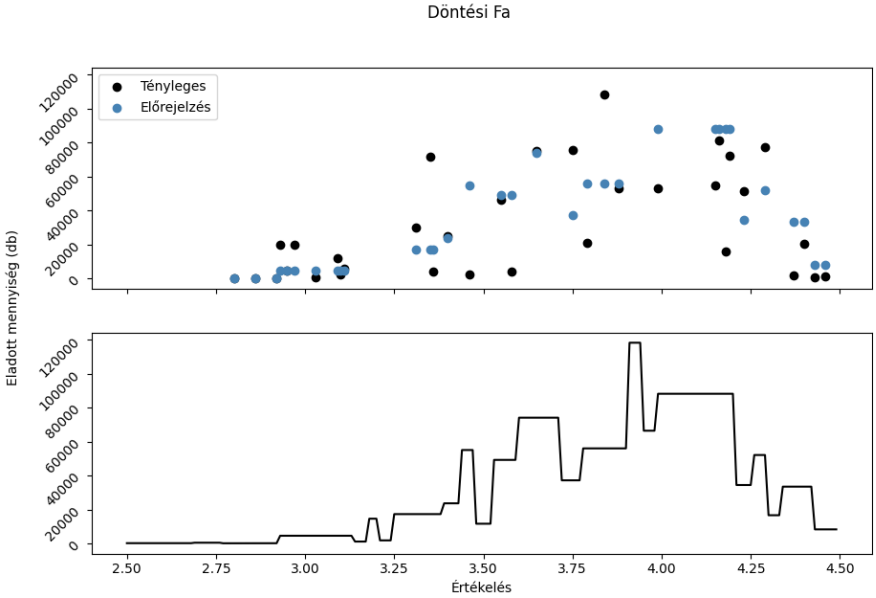
(Forrás: saját szerk.)



A döntési fa modelljét a *19. ábrán* mutatom be. A felső pontdiagramon láthatjuk a tényleges és előre jelzett értékek elhelyezkedését a teszt adatokra vetítve. Az alsó ábrán az előre jelzett értékek láthatóak vonaldiagramon. Jól látható, hogy a modell mélységéből adódóan sok helyen általánosít, tehát nem értem el ugyanolyan részletességet, mintha nagyobb mélységet állítottam volna be, viszont így el tudtam kerülni a túlillesztést. Döntési fával a 3.5 értékeléshez tartozó előre jelzett eladás 11,802 db.

19. ábra: Döntési fa modellje

(Forrás: saját szerk.)

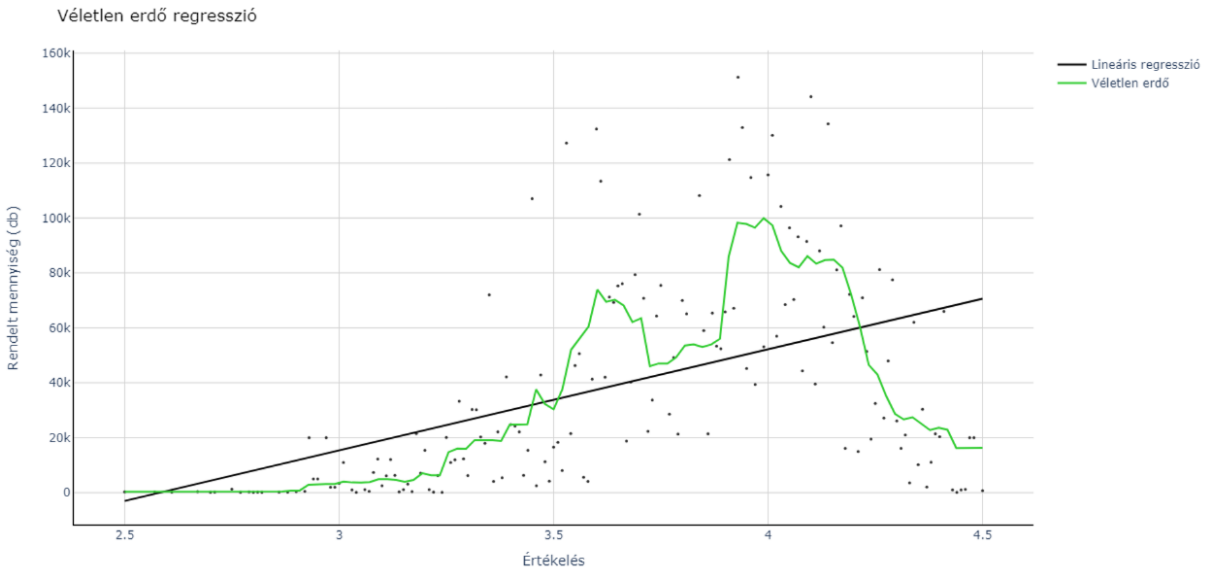


Véletlen Erdő (Random Forest) Regresszió

A véletlen erdő modellemhez is a *Scikit-learn* könyvtárat használtam, amelyen belül elérhető a *RandomForestRegressor()* függvény. Az *n_estimators* értékén nem változtattam, tehát a modell az alapbeállítás szerinti 100 db fát hozza létre, amelyeknek eredményeit átlagolja. A *20. ábrán* a véletlen erdő regressziót hasonlítom össze a lineáris regresszió egyenesével. Jól látszik, hogy a véletlen erdő sokkal dinamikusabb, a tényleges értékeket jobban megközelítő előrejelzést tesz lehetővé. A véletlen erdő modell produkálta a legalacsonyabb előre jelzett eladást. Ezzel a módszerrel 3.5 értékeléshez 30,334 a becsült eladott darabszám.

20. ábra: Véletlen erdő regresszió

(Forrás: saját szerk.)



3.2 Regresszió-analízis modellek kiértékelése mérőszámokkal

A 3.1 fejezetben tárgyalt elemzéshez hasonlóan, a regresszió-analízis modelleket is tanuló és teszt adatokra bontva értékeltem. Az eredményeket most is az átlagos abszolút hiba értéke alapján rendeztem csökkenő sorrendbe. A mérőszámokat a *9.* és *10. táblázatban* foglaltam össze.

9. táblázat: Regresszió-analízis modellek kiértékelése tanuló adatok alapján

(Forrás: saját szerk.)

Regressziós modell	**MAE**	**RMSE**	**MAPE**
Döntési fa	13,873	21,164	2.03
Véletlen erdő (Random Forest) regresszió	15,218	22,374	3.46
Kvantilis regresszió (medián)	16,005	25,361	3.69
Tartóvektor regresszió	18,711	28,051	4.56
Lineáris regresszió	25,837	34,916	16.09

A táblázat alapján azt állapítottam meg, hogy a legjobb eredményt a sima döntési fa és a véletlen erdő regresszió modelljével értem el. Ez köszönhető annak, hogy ezeknek a fajta modelleknek

alapvetően nagyon magas az előrejelzési teljesítménye. A véletlen erdő modell jól működik olyan adathalmazokon, amelyek zajosak, illetve amikor a változók között bonyolult nemlineáris kapcsolat áll fenn. A döntési fával is jó eredményt kaptam. A döntési fa sokkal kevésbé robosztus a túlillesztéssel szemben, emiatt ennél a modellnél korlátoztam a fa mélységét, ami magasabb hibákat eredményezett, viszont – ahogy azt látni fogjuk – így nem térnek el jelentősen a tanuló és teszt adatokra kapott mérőszámok.

A kvantilis és tartóvektor regresszió esetében hasonló eredményt kaptam. A kvantilis regresszió során alkalmazott gradiensnövelés láthatóan sokat segít a modell pontosságának növelésében. Ez a módszer, robosztussága miatt, alkalmas a zajos adatok kezelésére, továbbá kiválóan alkalmazható együttes (ensemble) módszerként. Együttes módszernek azt nevezzük, amikor több gépi tanulási algoritmust használunk fel, amelyek együttesen hoznak döntéseket, és generálnak ezáltal megbízhatóbb és pontosabb előrejelzéseket (Re és Valentini, 2012). A tartóvektor regresszió jól kezeli a nemlineáris összefüggéseket a változók között, viszont nehézkes a megfelelő paraméterek kiválasztása és finomhangolása miatt, illetve sokkal érzékenyebb a zajra és a kiugró értékekre.

A legrosszabb eredményre a lineáris regresszióval jutottam. Ez a modell csak akkor ad pontos előrejelzéseket, ha a változók között erős lineáris kapcsolat mutatható ki, ami ebben az esetben nem áll fenn.

10. táblázat: Regresszió-analízis modellek kiértékelése teszt adatok alapján*(Forrás: saját szerk.)*

Regressziós modell	**MAE**	**RMSE**	**MAPE**
Tartóvektor regresszió	16,675	22,265	2.57
Véletlen erdő (Random Forest) regresszió	18,209	26,322	2.77
Döntési fa	18,882	27,047	2.53
Kvantilis regresszió (medián)	19,471	25,822	2.56
Lineáris regresszió	23,174	30,516	11.21

A teszt adatok kiértékelésével némileg árnyaltabb képet kaptam a modellek teljesítményéről. Egyértelmű, hogy a lineáris regresszió a teszt adatokra sem illeszkedik egyáltalán, ami összhangban van a korábban tett megállapításommal.

A döntési fán alapuló modellek hasonlóan jól teljesítenek az összes mérőszám alapján, ami azt jelenti, hogy ezek a modellek jól illeszkednek.

A kvantilis és tartóvektor regresszióval is hasonló végeredményt kaptam. A tartóvektor regresszió bizonyult a legmegbízhatóbb előrejelzési módszernek, abban a tekintetben, hogy a tanuló és teszt adatok alapján elvégzett mérések eredményeinek különbsége ennél a modellnél a legkisebb.

A regressziós modellek esetében is elvégeztem a *7. táblázathoz* hasonló összehasonlítást. A *11. táblázat* a 3.5 értékeléshez tartozó célváltozó előre jelzett értékeit foglalja össze a különböző módszerek esetében. Ebben az összehasonlításban láthatóan nagy a különböző modellekkel végzett előrejelzések szórása. A ténylegeshez legközelebbi előrejelzéseket a kvantilis regresszióval és döntési fával kaptam. A lineáris és tartóvektor regresszió viszont egyáltalán nem bizonyul megbízhatónak erre a feladatra.

11. táblázat: Előrejelzések pontossága 3.5 értékelésű termékek alapján

(Forrás: saját szerk.)

Regressziós modell	**Előrejelzett érték (Értékelés=3.5)**	**Tényleges értéktől való eltérés**
Döntési fa	11802	-4808
Kvantilis regresszió (medián)	18024	1414
Véletlen erdő (Random Forest) regresszió	30334	13724
Lineáris regresszió	33783	17173
Tartóvektor regresszió	35382	18772

A determinációs együttható (r^2) vizsgálatával árnyaltabb képet kaphatunk a modellek illeszkedésével kapcsolatban. A modellek megalkotásakor az volt a célom, hogy a lehető legjobb r^2 értéket kapjam mind a tanuló, illetve a teszt adatokra való illesztéskor. Az egyes modellekhez tartozó r^2 értékeket a *12. táblázat* tartalmazza.

12. táblázat: Determinációs együttható

(Forrás: saját szerk.)

Regressziós modell	**Tanuló**	**Teszt**
Döntési fa	0.714	0.243
Véletlen erdő (Random Forest) regresszió	0.681	0.283
Kvantilis regresszió (medián)	0.529	0.044
Tartóvektor regresszió	0.499	0.487
Lineáris regresszió	0.224	0.037

A *12. táblázat* alapján megállapítottam, hogy a döntési fa és véletlen erdő illeszkedik a legjobban a tanuló adatokra. A teszt adatok alapján viszont a véletlen erdő jobban teljesít, tehát igaznak bizonyult az a megállapításom, hogy a véletlen erdő ellenállóbb a túlillesztéssel szemben az egyszerű döntési fához képest.

A kvantilis és lineáris regresszió modellek nagyon rosszul illeszkednek a teszt adatokra, így ezeket érdemes elvetni. A tartóvektor regresszióval értem el a legkiegyensúlyozottabb modellt, mivel a tanuló és teszt adatokra is hasonló eredményt kaptam. Ebből a szempontból érdemes ennek a modellnek a használatát is fontolóra venni.

4 Következtetések és javaslatok

Az idősor-elemző módszerek tárgyalása során megállapítottam, hogy a háromszoros exponenciálsimítás modellje illeszkedik legjobban a tanuló adatokra, illetve a teszt adatok tekintetében is a legjobb háromban benne van, ezért érdemes erre az előrejelzési eljárásra alapozni a becsléseket. Kifejezetten szembetűnő a modell eredményessége a szezonálisan megugró eladások előrejelzésére, aminek köszönhetően elkerülhető ezekben a periódusokban a készlethiány.

Mivel a kereslet ebben az esetben nem állandó, ajánlott folyamatosan figyelni a készletet, hogy minden időpontban ismerjük az aktuális készletszintet. Készletezési stratégiák tekintetében egyik tárgyalt módszer sem kielégítő ebben az esetben. Mivel periodikusan kiugró értékekkel találkozunk, maximum készletszint meghatározása nem ajánlott, mivel akkor azt a legmagasabb eladáshoz kellene kötni, viszont láthattuk, hogy a legtöbb hónapban az eladás ettől jóval elmarad. Állandó mennyiség rendelése sem ajánlott, mivel azzal szintén nem fedhető le teljesen az igény. Érdemes az idősor-elemző modell által előre jelzett értékekre alapozni a rendelési mennyiséget, tehát állandó időközönként az előrejelzések szerint tölteni a készletet. Ez az év nagy részében nagyjából egyenletes készletszintet fog jelenti, évente egyszer pedig az átlagtól jóval magasabbat.

Számos idősor-elemző módszer van, amelyek alkalmasak lehetnek előrejelzésre. A megfelelő módszer kiválasztásához szükség van a múltbeli kereslet alakulásának ismeretére, meg kell vizsgálni, hogy az milyen trendeket mutatnak az adatok, ha egyáltalán beszélhetünk ilyenről, illetve vannak-e kiugró értékek, aminek a kezelését bele kell építeni a modellbe. A 2.3 fejezetben tárgyalt stratégiákhoz képest egy sokkal rugalmasabb készletezési módszert tudunk így kialakítani, ami nagyban hozzájárul, hogy hatékonyan tudjuk kezelni a változó keresletet.

Azt is megvizsgáltam, hogy egy bemeneti változó és az eladási adatok (célváltozó) kapcsolata alapján tudok-e használható becslést adni egy adott termék várható eladásaira. Alapvetően nem találtam olyan modellt, ami jól illeszkedik az adatokra, tehát a gyakorlatban nem érdemes ebben a vonatkozásban előrejelzéseket készíteni, mert azok nagy valószínűséggel pontatlanok lesznek. Ugyanakkor a tárgyalt módszereket összehasonlítva tehetünk megállapításokat azok egymáshoz viszonyított pontosságával kapcsolatban. Mivel a felhasznált adatokra alapvetően heteroszkedaszticitás jellemző, nem tudtam az egyszerű lineáris modellel jó eredményt elérni, tehát mindenképp szükség van egy komplexebb algoritmus használatára, amivel leírható a változók közötti nemlineáris kapcsolat.

A regressziós modellek közül a MAE, RMSE és MAPE értékek alapján a döntési fák teljesítettek a legjobban. A 3.5 értékeléshez a legközelebbi becslést a kvantilis regresszióval értem el, viszont ez a modell nagyon rosszul illeszkedik a teszt adatokra, tehát várhatóan nem lennének megbízhatóak a más értékekkel számolt becslések. A tartóvektor regresszióval értem el a legjobb illeszkedést összességében, viszont a 3.5 értékelésre ezzel a módszerrel kaptam a legnagyobb eltérést az aktuális eladási adathoz képest. A *17. ábrán* látható, hogy körülbelül 3.2 értékelés felett rendkívül pontatlanná válik a modell.

Összességében arra a megállapításra jutottam, hogy döntési fákkal nagyon jó becsléseket tudunk készíteni. Az eredményeken látszik, hogy az általam felhasznált adatok vonatkozásában ez még mindig jócskán elmarad a 100%-hoz közeli pontosságtól, viszont egy egyszerű lineáris modellhez képest nagyságrendekkel pontosabb becsléseket tudtam adni.

A készletgazdálkodás vonatkozásában hasznosnak bizonyulhat, ha különféle regresszió-analízis módszerekkel megvizsgáljuk egy vagy több változó és az eladási adatok kapcsolatát. Amennyiben valamilyen jól modellezhető kapcsolatot fedezünk fel, az sokat segíthet nekünk abban, hogy előre felmérjük a várható igényt egy-egy termék iránt.

5 Összefoglalás

Szakdolgozatom célja az volt, hogy olyan előrejelzési módszereket találjak, amelyekkel javítani tudom a készletgazdálkodás hatékonyságát. Az alapfeltevésem az volt, hogy semmilyen más forrásból származó előrejelzésem nincsen, tehát kizárólag múltbeli eladási adatokkal dolgozhatok, amelyek esetében a kereslet eloszlása nem egyenletes.

A dolgozatot a készletgazdálkodás lényeges komponenseinek tárgyalásával kezdtem, mint a rendelési időköz, rendelési tétel nagysága, illetve alapvető készletezési stratégiák összefoglalása, hogy ismertessem ezek jelentőségét a kereslet alakulásának függvényében, illetve a várható kereslet ismeretének fontosságát hangsúlyozandó. Ezek alapján olyan módszereket kerestem, amelyekkel többé-kevésbé áthidalható az a probléma, hogy az alapvető készletezési stratégiák, ingadozó kereslet esetén, nem garantálják a hiány elkerülését.

A téma kifejtése során megvizsgáltam több idősor-elemző és regressziós modellt, amelyeket minta adatsorokon teszteltem. A célom ezzel az volt, hogy a vizsgált modelleket egymással összevetve fel tudjam mérni azok erősségeiket, gyengeségeiket, és egymáshoz viszonyított pontosságukat. A pontosság mérésére több mérőszámot is használtam, hogy minél részletesebb képet kapjak a modellek teljesítményéről. Mérőszámaim az átlagos abszolút hiba (MAE), átlagos négyzetes hiba négyzetgyöke (RMSE) és az abszolút százalékos hiba (MAPE) voltak. Az eredményeim természetesen csak az általam felhasznált adatokra vonatkoznak. Minden esetben szükséges többféle modellt illesztenünk az általunk aktuálisan használt adatokra, hogy az eredmények alapján ki tudjuk választani a legjobbat. A modellek egyenként való ismertetése és összehasonlítása mellett foglalkoztam a kiugró értékek kezelésével, ensemble módszerekkel, illetve a modellek paramétereinek finomhangolásával és rácskereséssel (grid search), amely módszerekről az egyes regressziós modellekről szóló fejezetekben tértem ki.

Az idősor-elemző módszerek tárgyalása során találtam olyan modellt, amely kimagaslóan jól teljesített, és alapvetően jól illeszkedett az általam használt adatsorra. Itt alapvetően azt vizsgáltam, hogy szezonálisan ingadozó eladások esetén milyen előrejelzési módszer a leghatékonyabb, és arra a megállapításra jutottam, hogy a legjobb modell ebben az esetben a háromszoros exponenciálsimítás.

A regressziós modellek illesztésével kevésbé jutottam jó eredményre, az általam vizsgált változók közötti nagyon alacsony korreláció miatt, viszont átfogó képet kaptam a modellek

egymáshoz viszonyított teljesítményét illetően. Mivel alapvetően egyik regressziós modellem sem illeszkedett jól az adataimra, érdemes egyéb modelleket tesztelni. Amennyiben ezekkel sem kapunk jobb eredményeket, érdemes más független változókat keresni, és azokra szintén összehasonlítani a különböző modelleket.

Összefoglalva az eddigieket, számos olyan adatelemző módszer létezik, amelyekkel meg tudjuk becsülni a keresletet, amelynek ismerete elengedhetetlen ahhoz, hogy hatékonyan tudjunk gazdálkodni a készleteinkkel. Szakdolgozatomban ezek közül ismertettem és értékeltem 5-5 idősor-elemző és regressziós módszert. A méréseim alapján ki tudtam választani a legjobb módszereket az egyes esetekben, tehát alapvetően elértem a bevezetésben kitűzött célokat.

Irodalomjegyzék

1. Benkő J. (2018): *Készletgazdálkodás*. Gödöllő: Logisztika Oktatásért és Kutatásért Alapítvány
2. Raisz P., Fegyverneki S. (2011): *Sztochasztikus modellezés*. Nemzeti Tankönyvkiadó
3. Pat DeMarle (2019): *The Basics of the Newsvendor Model*. Letöltés dátuma: 2024.03.07. Forrás: https://medium.com/@pdemarle/the-basics-of-the-newsvendor-model-ef756f203433
4. H. E. Scarf (2002): *Inventory Theory*. New Haven, Connecticut: Cowles Foundation for Research in Economics, Yale University
5. Mihály Zs., Lelkes Z., Dósai T. (2019): *Logisztikai Alapismeretek*. Kecskemét: Neumann János Egyetem
6. V. Vlckova, M. Paták (2010): *Role of demand planning in business process management*. Pardubice: University of Pardubice. Letöltés dátuma: 2024.02.27. DOI: 10.3846/bm.2010.151
7. A. Chavan (2023): *Forecasting the Future: A Comprehensive Guide to Moving Averages in Time Series Analysis*. Letöltés dátuma: 2024.02.20. forrás: https://medium.com/@anuj_chavan/forecasting-the-future-a-comprehensive-guide-to-moving-averages-in-time-series-analysis-880fa17e877a
8. D. Radečić (2021): *Time Series From Scratch — Exponentially Weighted Moving Averages (EWMA) Theory and Implementation*. Letöltés dátuma: 2024.02.20. forrás: https://towardsdatascience.com/time-series-from-scratch-exponentially-weighted-moving-averages-ewma-theory-and-implementation-607661d574fe
9. N. Malkari (2023): *Exponential Smoothing: A Method for Time Series Forecasting*. Letöltés dátuma: 2024.02.20. Forrás: https://medium.com/@nikhilmalkari18/exponential-smoothing-a-method-for-time-series-forecasting-7ea35ca2c781
10. S. Acharya (2021): *What are RMSE and MAE?* Letöltés dátuma: 2024.03.07. Forrás: https://towardsdatascience.com/what-are-rmse-and-mae-e405ce230383
11. J. Moody (2019): *What does RMSE really mean?* Letöltés dátuma: 2024.03.07. Forrás: https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e
12. S. Kim, H. Kim (2016): *A new metric of absolute percentage error for intermittent demand forecasts*. International Journal of Forecasting 32 (2016) 669-679. Letöltés dátuma: 2024.03.02. DOI: 10.1016/j.ijforecast.2015.12.003
13. S. Chatterjee, J. S. Simonoff (2013): *Handbook of Regression Analysis*. Hoboken, New Jersey: John Wiley & Sons, Inc.
14. D. H. Maulud, A. M. Abdulazeez (2020): *A review on linear regression comprehensive in machine learning*. Journal of Applied Science and Technology Trends 1(4):140-147. Letöltés dátuma: 2024.03.15. DOI: 10.38094/jastt1457
15. S. Dye (2020): *Quantile Regression*. Letöltés dátuma: 2024.03.15. Forrás: https://towardsdatascience.com/quantile-regression-ff2343c4a03

16. B. K. Gacar, I. D. Kocakoc (2020): *Regression Analyses or Decision Tree?* Celal Bayar Üniversitesi Sosyal Bilimler Dergisi. Letöltés dátuma: 2024.03.15. DOI: 10.18026/cbayarsos.796172
17. A. Liaw, M. Wiener (2001): *Classification and Regression by randomForest*. Letöltés dátuma: 2024.03.15. Forrás: https://www.researchgate.net/publication/228451484_Classification_and_Regression_by_RandomForest
18. A. Natekin, A. Knoll (2013): *Gradient boosting machines, a tutorial*. Frontiers in Neurobiotics December 2013 Volume 7 Article 21. Letöltés dátuma: 2024.03.18. DOI: 10.3389/fnbot.2013.00021
19. S. Lessmann, S. F. Crone, R. Stahlbock (2005): *Optimizing Hyperparameters of Support Vector Machines by Genetic Algorithms*. 2005 International Conference on Artificial Intelligence, ICAI 2005, Las Vegas, Nevada, USA, June 27-30, 2005, Volume 1. Letöltés dátuma: 2024.03.19. Forrás: https://www.researchgate.net/publication/220835507_Optimizing_Hyperparameters_of_Support_Vector_Machines_by_Genetic_Algorithms
20. M. Re, G. Valentini (2012): *Ensembe methods: A review*. Advances in Machine Learning and Data Mining for Astronomy (563-594. o.) Kiadó: Chapman & Hall. Letöltés dátuma: 2024.03.20. Forrás: https://www.researchgate.net/publication/230867318_Ensemble_methods_A_review
21. Hunyadi L. (2006): *A heteroszkedaszticitásról egyszerűbben*. Statisztikai Szemle 84. évfolyam 1. szám. Letöltés dátuma: 2024.03.24. Forrás: https://www.ksh.hu/statszemle_archive/2006/2006_01/2006_01_075.pdf
22. J. Lindner (2023): *Inventory Management Statistics: Market Report & Data*. Letöltés dátuma: 2024.04.13. Forrás: https://gitnux.org/inventory-management-statistics/
23. C. H. Lau (2019): *5 Steps of a Data Science Project Lifecycle*. Letöltés dátuma: 2024.04.13. Forrás: https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

Táblázatjegyzék

Ábrajegyzék

Melléklet

NYILATKOZAT

a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve:	Magyar Tamás Olivér
A Hallgató Neptun kódja:	PW2TUE
A dolgozat címe:	A készletgazdálkodás hatékonyságának növelése adatelemző módszerekkel
A megjelenés éve:	2024
A konzulens intézetének neve:	Műszaki Intézet
A konzulens tanszékének a neve:	Mérnökinformatikai Tanszék

Kijelentem, hogy az általam benyújtott szakdolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, és az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlant állítottam, tudomásul veszem, hogy a záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemitulajdon-kezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrár- és Élettudományi Egyetem könyvtári repozitori rendszerébe. Tudomásul veszem, hogy a megvédett és

- nem titkosított dolgozat a védést követően
- titkosításra engedélyezett dolgozat a benyújtásától számított 5 év eltelte után

nyilvánosan elérhető és kereshető lesz az Egyetem könyvári repozitori rendszerében.

Kelt: Gödöllő, 2024. év április hó 21. nap

Magyar Tamás

Hallgató aláírása

NYILATKOZAT

Magyar Tamás Olivér (hallgató Neptun azonosítója: PW2TUE konzulenseként nyilatkozom arról, hogy a szakdolgozatot áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A szakdolgozatot a záróvizsgán történő védésre **javaslom / nem javaslom**[1].

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem

Kelt: Gödöllő, 2024. év április hó 20. nap

Orosz Katalin dr.

belső konzulens

[1] A megfelelő aláhúzandó.