

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM  
MAROSVÁSÁRHELYI KAR,  
INFORMATIKA SZAK**



**SAPIENTIA  
ERDÉLYI MAGYAR  
TUDOMÁNYEGYETEM**

Eloszlás előrejelzés letapogatása Amazon Forecast segítségével

**DIPLOMADOLGOZAT**

Témavezető:  
dr. Kolumbán Sándor,  
dr. Iclánzan David Andrei,  
Egyetemi docens

Végzős hallgató:  
Burszán Hunor

**2023**

**UNIVERSITATEA SAPIENTIA DIN CLUJ NAPOCA  
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,  
SPECIALIZAREA INFORMATICĂ**



**UNIVERSITATEA  
SAPIENTIA**

Explorarea prognozelor de distribuție cu ajutorul Amazon  
Forecast

**LUCRARE DE DIPLOMĂ**

Coordonator științific:  
dr. Kolumbán Sándor,  
dr. Iclănanzán David Andrei,  
Conferențiar universitar

Absolvent:  
Burszán Hunor

**2023**

**SAPIENTIA HUNGARIAN UNIVERSITY OF  
TRANSYLVANIA**  
**FACULTY OF TECHNICAL AND HUMAN SCIENCES**  
**COMPUTER SCIENCE SPECIALIZATION**



**SAPIENTIA**  
HUNGARIAN UNIVERSITY  
OF TRANSYLVANIA

Exploring Distribution Forecasting with Amazon Forecast

**BACHELOR THESIS**

Scientific advisor:

dr. Kolumbán Sándor,  
dr. Iclănanz David Andrei,  
Associate professor

Student:

Burszán Hunor

**2023**

## LUCRARE DE DIPLOMĂ

Îndrumător: Conf. dr. Iclănanz David Andrei  
Coordonator științific: Lect. dr. Kolumbán Sándor

Candidat: Burszán Hunor  
Anul absolvirii: 2023

**a) Tema lucrării de licență:**

Explorarea prognozelor de distribuție folosind Amazon Forecast.

**b) Problemele principale tratate:**

O varietate largă a soluțiilor de logistică (ex. planificare stocurilor de siguranță, planificare achizițiilor în avans) necesită estimări a cantităților diferite (ex. vânzări a mărfurilor). În plus soluțiile mai avansate necesită informații mai detaliate a estimărilor, cum ar fi cuantile specifice sau informații detaliate despre distribuția estimărilor.

Platforma Amazon Forecast este un serviciu livrat de Amazon care oferă diferite opțiuni pentru modelarea datelor încărcate și oferă posibilitate de estimare, oferind cel mult 5 cuantile specificate.

Studentul trebuie să:

- exploreze, înțelege și prezintă în lucrare conceptele principale și pipe-line-ul folosit în platforma Forecast,
- prezinte structura datelor anonimizate, furnizate de coordonator științific,
- realizează componente software pentru analizare datelor și gestionarea procesului de preconizare în Amazon Forecast,
- în mod deosebit, platforma trebuie evaluată din punct de vedere a consistenței, adică cuantilele livrate poate fi combinate într-o descriere neparametrică a distribuției,
- evaluateze costurile implicate în folosirea platformei Forecast pentru cantități mari de date.

**c) Desene obligatorii:**

- Diagrama use-case pentru softului creat.
- Prezentarea interfeței a softului creat.
- Prezentarea generală arhitecturii a softului.
- Tratarea problemelor prezentate implică realizarea desenelor ce prezintă cuantilele obținute de la Amazon Forecast ca o funcție de distribuție cumulativă.

**d) Softuri obligatorii:**

- Software desktop pentru gestionarea datelor și pipe-line-ului în Amazon Forecast.
- Script pentru analizarea și vizualizarea rezultatelor obținute de la Amazon Forecast.

**e) Bibliografia recomandată:**

Documentație Amazon Forecast: <https://aws.amazon.com/forecast/>

Biblografie despre planificare stocurilor: Juan R. Trapero, Manuel Cardós, Nikolaos Kourentzes, Quantile forecast optimal combination to enhance safety stock estimation, International Journal of Forecasting, Volume 35, Issue 1, 2019, Pages 239-250.

**f) Termene obligatorii de consultații:**

Săptămânal, preponderant online.

**g) Locul și durata practicii:** Universitatea „Sapientia” din Cluj-Napoca,  
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, sala / laboratorul 513.

Primit tema la data de: 13. martie, 2023.

Termen de predare: 06. iulie, 2023.

Semnătura Director Departament

Semnătura responsabilului  
programului de studiu

## Declarație

Subsemnatul/a BURSEAN HUMOR, absolvent(ă) al/a specializării INFORMATICA, promoția 2020/2023 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea,

Data: 06.06.2023

Absolvent

Semnătura.....BL.....

# Kivonat

A kereslet előrejelzése kiemelkedően fontos és értékes tevékenység számos iparágban és üzleti környezetben. A pontos előrejelzés lehetővé teszi a vállalatok számára, hogy hatékonyan kezeljék készleteiket, optimalizálják tervezési és gyártási folyamataikat, kialakítsák eladási és marketingstrategiáikat, csökkentsék költségeiket és növeljék hatékonyúságukat. Az egyik fő előnye a kereslet előrejelzésnek a hatékony készletkezelés lehetősége, mivel az előrejelzések alapján a vállalatok minimalizálhatják készlethiányait és túlkészleteiket, optimalizálhatják raktárkészleteiket, valamint csökkenthetik tárolási és raktárköltségeiket. A kereslet előrejelzése segítséget nyújt a tervezési és gyártási folyamatok optimalizálásában is. Az előrejelzés alapján a vállalatok megtervezhetik gyártási kapacitásukat, beállíthatják termelési ütemterveiket, és hatékonyabban kihasználhatják rendelkezésre álló erőforrásait. Ezenkívül a kereslet előrejelzése alapján a vállalatok jobban megérthetik ügyfeleik igényeit és viselkedését. Ez lehetővé teszi személyre szabott termék kínálat kialakítását, eladási és marketingstrategiák optimalizálását, valamint a jobb ügyfélelmény nyújtását. A vásárlók elégedettsége és hűsége növelhető azzal, hogy a vállalatok az előrejelzések alapján pontosan elérhetik és kielégíthetik az ügyfelek igényeit.

A hatékony készletgazdálkodás részben a jó keresleti előrejelzéseken alapul. Az elég-telen készlet nem csak a vásárlók elégedetlenségét okozza, és bevételkiesést okoz, de ha ez elég gyakran előfordul, vagy egy ügyfél számára elég fontos alkalommal, akkor a jövőbeni üzlet elvésztesét is eredményezheti. A túlkínálat azonban a raktározás és a logisztika szempontjából költséges, és egyes készletek hosszú ideig - vagy akár örökre - eladatlanul maradhatnak, ami a befektetés teljes elvésztesét eredményezheti. A megfelelő készlet-egyensúly megtalálása tehát a jó kereslet-előrejelzés nélkülözhetetlen szempontja.

Az évek során a kereslet előrejelzés területén jelentős fejlődés tapasztalható. Az előrejelzési módszerek és technikák folyamatos fejlesztése, valamint a rendelkezésre álló adatok mennyiségének és minőségének növekedése jelentős hatást gyakorolt az előrejelzés pontos-ságára és megbízhatóságára. Emellett az adatok kezelésének és feldolgozásának fejlődése is jelentős előrelépést hozott. Az adatbányászat és a gépi tanulás algoritmusainak elő-rehaladása lehetővé teszi az összetett adathalmazok elemzését és a rejtett mintázatok azonosítását, ezzel segítve a pontosabb és megbízhatóbb előrejelzések elkészítését.

A szakdolgozat célja néhány termék jövőbeni kereslet eloszlásának előrejelzése, egy idősor előrejelzés elkészítési folyamatának bemutatásával az Amazon Forecast internetes szolgáltatás használatával, amelynek alkalmazásához nem szükségesek előzetes gépi tanulás és mesterséges intelligencia ismeretek. A dolgozatban bemutatásra kerül a teljes folyamat, ami az előrejelzések elkészítéséhez szükséges, a hozzá tartozó logikával együtt, valamint bemutatásra kerül a hozzá készített asztali alkalmazás, amely a szolgáltatással való kommunikációval képes a Forecast kezelésére. Az előrejelzések eredményei pedig megjelenítésre és értelmezésre kerülnek.

# Rezumat

Previziunea cererii este o activitate extrem de importantă și valoroasă în multe industrii și medii de afaceri. Previziunile exacte permit companiilor să gestioneze eficient stocurile, să își optimizeze procesele de proiectare și de producție, să dezvolte strategii de vânzări și de marketing, să reducă costurile și să crească eficiența. Unul dintre principalele beneficii ale previziunii cererii este capacitatea de a gestiona eficient stocurile, deoarece permite companiilor să reducă la minimum deficitul și surplusul de stocuri, să optimizeze nivelurile stocurilor și să reducă costurile de depozitare și de stocare. Previziunea cererii ajută, de asemenea, la optimizarea proceselor de planificare și de producție. Pe baza previziunilor, companiile își pot planifica capacitatea de producție, pot ajusta programele de producție și pot utiliza mai eficient resursele disponibile. În plus, prognoza cererii permite companiilor să înțeleagă mai bine nevoile și comportamentul clientilor lor. Acest lucru permite dezvoltarea de oferte de produse personalizate, optimizarea strategiilor de vânzări și de marketing și furnizarea unei experiențe mai bune pentru clienți. Satisfacția și loialitatea clientilor pot fi sporite prin faptul că le permite companiilor să atingă și să satisfacă cu exactitate nevoile clientilor pe baza previziunilor.

O gestionare eficientă a stocurilor se bazează în parte pe o bună prognoză a cererii. Nu numai că un stoc insuficient provoacă nemulțumirea clientilor și pierderea de venituri, dar dacă se întâmplă destul de des sau într-un moment destul de important pentru un client, poate duce și la pierderea unor afaceri viitoare. Cu toate acestea, aprovisionarea excesivă este costisitoare din punct de vedere al depozitariei și al logisticii, iar unele stocuri pot rămâne nevândute pentru o perioadă lungă de timp - sau chiar pentru totdeauna - ceea ce duce la o pierdere totală a investiției. Găsirea echilibrului corect al stocurilor este, prin urmare, un aspect esențial al unei bune programe de prognoză a cererii.

De-a lungul anilor, au existat evoluții semnificative în domeniul prognozei cererii. Dezvoltarea continuă a metodelor și tehniciilor de prognoză, precum și creșterea cantitatii și calitatii datelor disponibile au avut un impact semnificativ asupra acurateței și fiabilității prognozelor. În plus, îmbunătățirile în gestionarea și prelucrarea datelor au condus, de asemenea, la progrese semnificative. Progresele în domeniul mineritului de date și al algoritmilor de învățare automată permit analiza seturilor complexe de date și identificarea modelelor ascunse, contribuind astfel la realizarea unor prognoze mai precise și mai fiabile.

Scopul acestei teze este de a prezice distribuția viitoare a cererii unor produse prin demonstrarea procesului de creare a unei serii de prognoze temporale folosind serviciul web Amazon Forecast, care nu necesită cunoștințe prealabile de învățare automată și inteligență artificială. Lucrarea prezintă procesul complet necesar pentru a produce previziunile, inclusiv logica asociată, și aplicația desktop construită pentru a gestiona Forecast prin comunicarea cu serviciul. Rezultatele previziunilor sunt afișate și interpretate.

# Abstract

Demand forecasting is an extremely important and valuable activity in many industries and business environments. Accurate forecasting allows companies to effectively manage inventory, optimise their design and manufacturing processes, develop sales and marketing strategies, reduce costs and increase efficiency. One of the main benefits of demand forecasting is the ability to manage inventory efficiently, as it allows companies to minimise stock shortages and overstocks, optimise stock levels and reduce storage and warehousing costs. Demand forecasting also helps to optimise design and production processes. Forecasting allows companies to plan their production capacity, adjust production schedules and make more efficient use of available resources. It also enables companies to better understand their customers' needs and behaviour. This allows the development of personalised product offerings, the optimisation of sales and marketing strategies and the delivery of a better customer experience. It helps companies to improve customer satisfaction and loyalty by enabling companies to accurately reach and meet customer needs based on predictions.

Effective inventory management is partly based on good demand forecasting. Not only does insufficient inventory cause customer dissatisfaction and loss of revenue, but if it happens often enough, or at an important enough time for a customer, it can also result in the loss of future business. However, oversupply is costly in terms of storage and logistics, and some stock can remain unsold for a long time - or even forever - resulting in a total loss of investment. Finding the right stock balance is therefore an essential aspect of good demand forecasting.

Over the years, there have been significant developments in the field of demand forecasting. The continuous development of forecasting methods and techniques, as well as the increase in the quantity and quality of data available, has had a significant impact on the accuracy and reliability of forecasting. In addition, improvements in data management and processing have also led to significant progress. Developments in data mining and machine learning algorithms allow the analysis of complex data sets and the identification of hidden patterns, thus helping to produce more accurate and reliable forecasts.

The aim of this paper is to predict the future demand distribution of some products by demonstrating the process of creating a time series forecast using the Amazon Forecast web service, which does not require prior knowledge of machine learning and artificial intelligence. The paper presents the complete process required to produce the forecasts, including the associated logic, and the desktop application designed to manage the Forecast by communicating with the service. The results of the forecasts are displayed and interpreted.

# Tartalomjegyzék

<b>1. Bevezető</b>	<b>3</b>
1.1. A kereslet előrejelzés . . . . .	3
1.2. Időszorok előrejelzése . . . . .	3
1.3. Meglevő szolgáltatások használata . . . . .	4
1.4. Dolgozat szerkezete . . . . .	4
<b>2. Probléma leírása</b>	<b>6</b>
2.1. Az adathalmaz . . . . .	6
2.2. Eloszlás előrejelzése . . . . .	7
2.3. Célok . . . . .	7
2.4. Az Amazon Forecast szolgáltatásról . . . . .	8
2.4.1. Adathalmazok importálása . . . . .	8
2.4.2. Előrejelzők tanítása . . . . .	9
2.4.3. Előrejelzések Generálása . . . . .	15
<b>3. Rendszer bemutatása</b>	<b>17</b>
3.1. Kezelő szoftver . . . . .	17
3.1.1. Szoftver funkciói . . . . .	18
3.1.2. Adatbetöltés . . . . .	19
3.1.3. Előrejelzők (Predictors) . . . . .	21
3.1.4. Előrejelzések (Forecasts) . . . . .	22
3.1.5. Előrejelzések lekérdezése és exportálása . . . . .	23
3.1.6. Erőforrások listázása . . . . .	24
3.1.7. Erőforrások törlése . . . . .	25
3.1.8. Naplózás . . . . .	25
3.2. Python Scriptek . . . . .	25
<b>4. Technológiai háttér</b>	<b>27</b>
4.1. Áttekintő architektúra . . . . .	27
4.2. Az Amazon Forecast szolgáltatás működése . . . . .	28
4.3. Frontend . . . . .	30
<b>5. Kiértékelés</b>	<b>34</b>
5.1. Az Amazon Forecast által generált előrejelzések kiértékelése . . . . .	34
5.2. Költségek . . . . .	44
5.3. A szoftver eredményei . . . . .	44

<b>6. Összefoglaló</b>	<b>46</b>
<b>Összefoglaló</b>	<b>46</b>
6.1. Továbbfejlesztési lehetőségek . . . . .	46
<b>Köszönetnyilvánítás</b>	<b>47</b>
<b>Ábrák jegyzéke</b>	<b>49</b>
<b>Irodalomjegyzék</b>	<b>50</b>
<b>Függelék</b>	<b>51</b>

# **1. fejezet**

## **Bevezető**

Az előrejelzési technikák és módszerek az üzleti döntéshozatal és a kereslet tervezésének fontos eszközei. Az egyre növekvő adatmennyiséggel és az üzleti szféra dinamikájával összhangban vált az idősor előrejelzés egyre elterjedtebbé és fontosabbá. Az idősor előrejelzés olyan módszer, amelynek célja a jövőbeni értékek becslése a korábbi megfigyelések alapján. Ez a képesség kulcsfontosságú lehet a kereslet tervezésében, raktárkészletekkel kezelésben, termelési ütemezésben, és számos más üzleti folyamatban.

### **1.1. A kereslet előrejelzés**

Az egyik ilyen előrejelzési technika a kereslet előrejelzés, amely lehetővé teszi különböző termékek és szolgáltatások jövőbeli keresletének becslését rendelkezésre álló adatok és különböző analitikai módszerek alapján. Ez a folyamat kiemelkedően fontos szerepet játszik olyan üzleti döntésekben, mint az anyagbeszerzés, gyártási ütemezés, raktárkészletezés, ármeghatározás és értékesítési stratégiák kidolgozása, lehetővé téve a hatékony tervezést és az erőforrások optimalizálását. Ez a folyamat rendkívül komplex, amely számos tényezőt tartalmaz, mint például elérhető régebbi adatokat, trendeket, szezonális változásokat és más befolyásoló tényezőket. A kereslet előrejelzés során különböző módszereket alkalmazhatunk, például statisztikai és matematikai módszereket, adatbányászati technikákat vagy akár mesterséges intelligenciát.

A kereslet előrejelzés fő célja a bizonytalanság csökkentése és a megfelelő döntéshozatal elősegítése. Ez a folyamat nem egyszeri, hanem folyamatos művelet, amelyet rendszeresen végrehajtanak friss adatok alapján, a minél pontosabb és aktuálisabb előrejelzések érdekében.

### **1.2. Idősorok előrejelzése**

Az idősorok előrejelzése olyan módszer, amely időbeli adatsorok és múltbeli mintázatok alapján becsül meg jövőbeli értékeket. Az idősorok olyan adatok, amelyek időrendi sorrendben követik egymást és különböző időegységekben vannak megadva, például órákban, napokban stb.. Az idősorok előrejelzése múltbeli adatokra támaszkodik, hogy felismerje a jövőbeli trendeket és mintázatokat, azzal az alapvető feltevéssel, hogy a korábbi adatok és tendenciák meghatározó hatással lehetnek a jövőbeli események alakulására.

Az idősorok előrejelzésének eredményei általában olyan adatok, amelyek a jövőbeli értékeket tartalmazzák, azonban ezek az eredmények bizonytalanságokat is hordozhatnak, amelyek különböző módokon fejezhetők ki. A kvantilisek például olyan mértékegységek, amelyek a valószínűségi eloszlást reprezentálják, és segítenek értelmezni az előrejelzés bizonytalanságát. A kvantilisek megmutatját, hogy milyen tartományban valószínűíthető a jövőbeli értékek elhelyezkedése, és használatukkal lehetőségünk van felmérni az előrejelzés bizonytalanságát és az értékek valószínűségi eloszlását, ezzel segítve a döntéshozókat abban, hogy a kockázatot és a vállalható bizonytalanságot figyelembe véve tervezzenek és cselekedjenek.

### 1.3. Meglevő szolgáltatások használata

Az idősorok előrejelzéséhez számos szolgáltatás és szoftver áll rendelkezésre, amelyek különböző módszereket és technológiákat kínálnak. A fejlődés folyamatosan zajlik, és az elmúlt évtizedekben jelentős változások történtek a szolgáltatások és módszerek elérhetőségében.

Kezdetben a statisztikai módszerek dominálták az idősor előrejelzést, olyen módszerek például, mint az egyszerű mozgó átlag (SMA), exponenciális simítás (ETS), autoregresszív mozgó átlag (ARMA). Ezek a módszerek alapvetően a múltbeli adatok alapján modellezik a jövőbeli trendeket és mintázatokat [1].

Napjainkban az MI (mesterséges intelligencia) és a gépi tanulás fejlődése új perspektívákat nyitott az idősorok előrejelzése terén. Egy ilyen példa az SPSS Modeler, amely egy erőteljes és sokoldalú adat- és szövegelemző munkaállomás, amely segít a felhasználóknak gyorsan és intuitívan előrejelző modellekkel építeni programozás nélkül, és vizuális felületének segítségével könnyedén felfedezhetők a minták és trendek a strukturált vagy strukturálatlan adatokban.

További szolgáltatások és platformok is rendelkezésre állnak az idősor előrejelzés területén, például a Microsoft Azure Machine Learning, Google Cloud AI Platform, Meta (Facebook) Prophet, vagy az Amazon Forecast.

Az Amazon Forecast egy felhőalapú szolgáltatás, amely gépi tanulást (ML) használ, nagyon pontos előrejelzések létrehozásához [2]. Az Amazon Forecast széles körben alkalmazható különböző területeken, mint például a termékigény becslése, az ellátási lánc optimalizálása, az energiaigény előrejelzése vagy a forgalomigény becslése. Az Amazon Forecast használatához csak a korábbi idősor adatokat kell megadni, és opcionális bármi-lyen adatot, amely befolyásolhatja az előrejelzéseket. A szolgáltatás automatikus elkészíti és frissíti a modelleket, így a felhasználóknak nem szükséges előzetes gépi tanulás tapasztalattal rendelkezniük.

### 1.4. Dolgozat szerkezete

Dolgozatom témaja, az Amazon Forecast előrejelző szolgáltatás, valamint az ezzel való kereslet előrejelzés folyamatának részletes bemutatása, és az általa generált előrejelzések vizsgálata.

A 1. fejezetben röviden ismertettem a kereslet előrejelzés jelentőségét és az időszorok előrejelzésének feladatát. Ezt követően bemutattam néhány meglevő szolgáltatást. amelyeket kereslet előrejelzésre lehet használni, kiemelve az Amazon Forecast szolgáltatást.

A 2. fejezetben részletesen leírom, hogy milyen adatok álltak rendelkezésre és szemléltetem az eloszlás előrejelzés nehézségeit és technikáit. Emelett bemutatom a dolgozat célját, és hogy mit kell tudni az Amazon Forecast szolgáltatásról, annak folyamatairól és működéséről.

A 3. fejezetben bemutatom az Amazon Forecast szolgáltatás kezeléséhez készített szoftvert és az előrejelzések elemzésehez írt Python szkripteket.

A 4. fejezet áttekintést nyújt az Amazon Forecast szolgáltatás technológiai hátteréről és működéséről, továbbá a kezelő szoftver frontendjének fejlesztéséről.

Az 5 fejezetben bemutatásra kerülnek az Amazon Forecast szolgáltatás által generált előrejelzések, figyelembe véve az eredményeket és a szolgáltatás időbeli és díjbeli költségeit. Emellett összefoglalásra kerülnek a szoftver eredményei és a használata közbeni tapasztalatok.

## **2. fejezet**

# **Probléma leírása**

A szakdolgozatom során azt fogom körbejárni, hogy hogyan lehet meglevő adatokból előrejelzéseket készíteni az Amazon Forecast szolgáltatás segítségével, majd az eredmények alapján eloszlást kinyerni a kigenerált előrejelzések ből. Ennek érdekében részletesen bemutatom a szolgáltatás használatához és az előrejelzések készítéséhez szükséges lépések, valamint bemutatom az elkészült előrejelzések eredményeit.

### **2.1. Az adathalmaz**

Az előrejelzések készítéséhez elengedhetetlenül fontosak a megfelelő adatok rendelkezésre állása, az adathalmaz az előrejelzés alapját képzi, és meghatározza a készítendő modell pontosságát és megbízhatóságát. Ebben az alfejezetben ismertetem az előrejelzésekhez használt adatokat, valamint azok struktúráját és szervezését.

Az általam felhasznált adatok egy vállalkozás által pár évre visszamenőleg árult vízvezeték szerelési termékeinek és különböző háztartási eszközeinek a rendelési feljegyzéseit tartalmazzák. Az adathalmazban szereplő termékek osztályozása egy CSV (comma separated values) fájlban található, amely tartalmazza minden termék azonosító számát, nevét, főcsoportját és alcsoportját. A termékek eladásainak feljegyzései külön csv fájlokban találhatóak, amelyek a termékek azonosító számának megfelelő nevű fájlokban rögzítettek, például, az 1011006 azonosítójú termék eladásainak feljegyzései az 10110006.csv fájlban találhatóak.

09/30/2017	1512	11087.38
10/31/2017	976	6703.85
11/30/2017	582	3861.5
12/31/2017	104	668.76
01/31/2018	401	2698.98
02/28/2018	100	665

**2.1. ábra.** Egy termék feljegyzései

Az adathalmazban összesen több, mint 22 ezer termékről vannak feljegyzések, egy adott termék feljegyzései hónapokra vannak lebontva, ahol minden hónap külön sorokban található. minden sor tartalmazza a hónapot, az adott termékből történt rendelések mennyiségét és a termék árát az adott hónapban. A termékek feljegyzései különböző időintervallumokat tartalmaznak, a legkorábbi feljegyzés az egyik termékről 2013 januárjában történt, azonban a legtöbb termék feljegyzéseinek kezdete 2016-ra tehető. Másfelől akad néhány termék, amelyről csak néhány hónapnyi adat áll rendelkezésre.

## 2.2. Eloszlás előrejelzése

Az eloszlás előrejelzés olyan módszer, amelynek segítségével egy adott jelenség vagy változó jövőbeli eloszlását próbáljuk megbecsülni. Az eloszlás az adatok eloszlása, vagyis az, hogy az adott változó milyen gyakorisággal vesz fel különböző értékeket. Az eloszlás előrejelzése lehetővé teszi számunkra, hogy becsléseket és prognózisokat készítsünk jövőbeli értékekről és a velük kapcsolatos valószínűségekről. Az eloszlás előrejelzés megpróbálja megbecsülni az előrejelzett értékek eloszlását, egy egyedüli érték helyett, és egy bevett gyakorlat olyan területeken, mint az időjárás vagy a fertőző betegségek előrejelzése, mert az előrejelzett sűrűség becslésével magában foglalja a kereslet bizonytalanságát. A különbség a becsült kvantiilisek és a becsült átlag között felfogható úgy, mint a biztonsági készlet szintje a hibás kereslet megfigyelésekből kifolyólag.

## 2.3. Célok

A cél az, hogy az Amazon Forecast által készült előrejelzésekkel képesek legyünk eloszlást kinyerni, de mivel a szolgáltatás egyik megkötlese miatt, egy előrejelzésben maximum csak 5 darab percentilis értéket képes megadni, több előrejelzést kell véghezvinni. A kvantiiliseket a 0.1-es értéktől a 0.99-es percentilisig kell lekérdezni ötösével, majd miután az összes előrejelzés elkészült, ezeket összesíteni és eloszlást kinyerni. Mivel több elemről áll adat rendelkezésre, és a adathalmazok létrehozása során több elemet is meg lehet adni egyszerre, azt is megvizsgálom, miként befolyásol egy előrejelzést, ha több

elemre illeszkedik egy előrejelző modell. A cél elérésének érdekében szükség volt egy alkalmazás létrehozására is, amely leegyszerűsíti a Forecast szolgáltatásának kezelését az előrejelzések elkészítéséhez, majd az előrejelzések kivizsgálására és szemléltetésére más eszközök is felhasználásra kerültek.

## 2.4. Az Amazon Forecast szolgáltatásról

Az Amazon Forecast egy teljesen kezelt szolgáltatás, amely statisztikai és gépi tanulás algoritmusokat használ nagy pontosságú, idősorok előrejelzésének biztosítására. Az [Amazon.com](#)-on használt idősorok előrejelzésére szolgáló technológián alapuló Forecast, a legkorszerűbb algoritmusokat kínálja jövőbeli idősorok adatainak előrejelzéséhez előzményadatok alapján, és nem igényel előzetes gépi tanulási tapasztalatokat.

Az idősor előrejelzés többek között hasznos a kiskereskedelemben, a pénzügyekben, logisztikában és az egészségügyben. A Forecast segítségével olyan mutatókat lehet előrejelzni, mint a raktárkészlet, munkaerő, webforgalom, szerverkapacitás vagy akár pénzügy specifikus metrikák. A szolgáltatás automatizálja az idősorok előrejelzési folyamatának nagy részét, így a felhasználóknak csak az adatkészletek előkészítésére és az előrejelzések értelmezésére kell összpontosítanuk.

Az Amazon Forecast szolgáltatás használatához három folyamatot és fogalmat kell alapvetően megismerni: adathalmazok importálása, előrejelzők tanítása és előrejelzések generálása. Az adathalmazok importálása az első lépés, tulajdonképpen ezek azok a strukturált adatok, amelyek tartalmazzák a vizsgálni kívánt jelenség idősorát, illetve egyéb, más kapcsolódó releváns attribútumokat is tartalmazhatnak. A következő lépés az előrejelzők tanítása, ezek olyan modellek vagy algoritmusok, amelyek képesek előrejelzéseket készíteni az adathalmazok alapján. A Forecast szolgáltatás különböző előrejelző modelleket kínál, amelyeket tanítani lehet a már importált adatok felhasználásával. Miután az előrejelző modell sikeresen tanult előrejelzéseket kell generálni belőlük, majd ezeket a előrejelzéseket lehet lekérdezni. Az Amazon Forecast szolgáltatásban ezen erőforrások menedzselésével és felhasználásával lehetünk képesek előrejelzéseket készíteni [2].

### 2.4.1. Adathalmazok importálása

Adataink előrejelzésekhez való felhasználásához a Forecast szolgáltatásnak valamilyen módon meg kell szerveznie őket, ez a folyamat az adathalmazok importálása, amelynek több erőforrásra is szüksége van.

Az adathalmazok (datasets) olyan erőforrások, amelyek az előrejelzők (predictors) tanításához szükséges adatokat és azok jellemzéseit tartalmazzák, vagyis olyan információkat az adatokról, mint például az adatok frekvenciája vagy másnéven intervalluma, amelyben az adatok rögzítve lettek, ez lehet óra, nap, hónap, stb.. Az adathalmaznak le kell írnia az előrejelzés formátumát (tartományát) és az adathalmaz típusát is (a tartományon belül). A tartomány azt mondja meg, hogy az előrejelzést milyen területen szeretnénk elvégezni (kereskedelmi, munkaerő, készletezés, egyéni, stb.), míg az adathalmaz típusa segít a megszervezni az tanuláshoz szükséges adatokat előrejelzés-barát kategóriákká. Egy másik információ amit biztosítani kell az adathalmaz számára az a séma, ami az idősorok adatainak oszlop fejlékeit, valamint ezen oszlopok adattípusait. Mindezek mellett opcionálisan meg lehet adni geolokációs és időzóna információkat is,

amelyek segíthetnek az előrejelzések pontosságában, bizonyos kategoriák eső cél értékek előrejelzéseinél.

Adathalmaz létrehozásakor ki kell választani a tartományt és a típust, ennek érdekében a Forecast biztosít néhány használati esetet, mint például a kereskedelmi kereslet vagy web forgalom előrejelzése. minden tartományon belül, a következő adathalmaz típusokat lehet megadni:

- **Cél idősor (target time series dataset)** - ezt az adathalmazt kötelezően meg kell adni, egy előrejelzés elkészítéséhez, ez a típusú adathalmaz tartalmazza azt a cél mezőt, amelyre az előrejelzést akarjuk generálni.
- **Kapcsolódó idősor (related time series dataset)** - ez a típus akkor használatos, ha az adataink között vannak olyan idősorok, amelyek nem tartalmazzák a cél mezőt, de értékeik akár befolyásolhatják azt az előrejelzés során.
- **Elem metaadat (item metadata dataset)** - ezt akkor érdemes használni, hogy ha vannak olyan adataink is amelyek nem idősorok, hanem statikus adatok, értékes információkat tartalmazhatnak a cél és kapcsolódó idősorainkhoz.

Az adataink által rendelkezésre álló információk és az előrejelezni kívánt feladattól függően, akár több adathalmazt is készíteni kell. minden adathalmaznak rendelkeznie kell egy sémával, amely az adatok mezőt írja le. Ebben listázzuk mind a szükséges és opcionális dimenzióit és jellemzőit az adathalmazba szánt adatoknak.

Miután a szükségesnek vélt adathalmazok elkészültek, ezeket egy másik erőforrás-hoz kell rendelni. Ennek az erőforrásnak a neve az adathalmaz csoport (dataset group), amely az előbb felsorolt három, egymást kiegészítő adathalmaz típusból tartalmazhat 1-1 et. Egy adathalmaz csoportnak kötelezően tartalmaznia kell egy cél idősort, és opcionálisan a másik kettőt, az elkészített adathalmazokat hozzá kell rendelni egy adathalmaz csoporthoz, majd magukat az adatakat egy Amazon S3 Bucket-ből importálni kell az adathalmazokba, erről a folyamatról részletesebben [4.2](#) fejezetben.

## 2.4.2. Előrejelzők tanítása

Miután az adathalmazok importálása megtörtént, adathalmaz csoportok felhasználásával rendelkezésre állnak az adataink egy előrejelző (predictor) tanításához. Egy előrejelző nem más, mint egy Amazon Forecast modell, amely azokból az adathalmazokból tanul, amelyeket a hozzárendelt adathalmaz csoportba importáltunk. Ezeket a prediktorokat a tanulás végeztében arra használhatjuk, hogy előrejelzéseket generálunk belőlük az idősoraink alapján.

Egy előrejelző elkészítéséhez a következő információkra és bemenetekre van szükség:

- **Adathalmaz csoport** - A adathalmaz csoportnak kötelezően tartalmaznia kell egy cél idősort. A cél idősor pedig tartalmazz egy azonosítót (item\_id) a cél tulajdonságot (target\_value), egy időbeli mezőt, valamit valamennyi szükséges más dimenziót. A kapcsolódó idősor és az elem metaadatok opcionálisak.
- **Előrejelzés gyakoriságát** - Ez az előrejelzések tagoltságára vonatkozik (órák, napok, hetek, stb.), tehát arra, hogy milyen időegységekben készüljön el az előrejelzés.

Ezt akár pontosítani is lehet frekvencia egységek és értékek megadásával. Ha például minden két hétre szeretnénk előrejelzést, ez úgy tehetjük meg, ha a frekvencia egység a hetente, az érték pedig 2. Abban az esetben, hogyha az adatok nagyobb gyakorisággal lettek rögzítve, mint az előrejelzéshez megadott gyakoriság, akkor a meglevő adatok összesítésre kerülnek a megadott gyakoriság szerint.

- **Előrejelzés horizont** - Az előrejelzett lépések száma, vagyis az, hogy mekkora időszakot jelezzen előre a gyakoriság szerint.

Opcionálisan még meg lehet adni a következő bemeneteket:

- **Időigazítási határ** - Ennek megadásával pontosíthatjuk azt, hogy az adatok összesítése ha szükséges, miként történjen.
- **Előrejelzés dimenziók** - Ezek a dimenziók opcionális jellemzők a cél idősorban, amelyek kombinálva használhatóak a cél értékkel különálló idősorok létrehozásához.
- **Előrejelzés típusok** - A kvantilisek az előrejelző kiértékeléséhez.
- **Optimalizálási metrikák** - A pontossági metrika az előrejelző optimalizálásához.
- **További adathalmazok** - Egyéb, a szolgáltatásba beépített adathalmazok, mint például időjárás index vagy ünnepnapok.

Az Amazon Forecast alapértelmezett módon egy *AutoPredictor* hoz létre, ahol a Forecast az optimális algoritmus kombinációkat alkalmazza az adathalmazainkban található idősorokra. Általában ezek az AutoPredictor-ok pontosabbak az AutoML-el vagy kézileg kiválasztott algoritmusú előrejelzőknél, és a szolgáltatás egyes funkciói csak AutoPredictor-ok esetén használhatóak.

A prediktorok kiértékeléséhez az Amazon Forecast pontossági metrikákat alkalmaz, annak érdekében, hogy az adataink számára legmegfelelőbb prediktorokat válasszuk az előrejelzések generálásához. A Forecast a prediktorokat olyan mérőszámokkal értékeli ki, mint a Root Mean Square Error (RMSE), Weighted Quantile Loss (wQL), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE) és Weighted Absolute Percentage Error (WAPE).

### **Root Mean Square Error (RMSE)**

A Root Mean Square Error a négyzetes hibák átlagának a négyzetgyöke, ennek következtében sokkal érzékenyebb a kilogó értékekre mint más pontossági metrikák. Minél kisebb az értéke, annál jobb a modell pontossága.

$$RMSE = \sqrt{\frac{1}{nT} \sum_{i,t} (\hat{y}_{i,t} - y_{i,t})^2}$$

Ahol:

$y_{i,t}$  - a vizsgált érték az  $(i, t)$  pontban

$\hat{y}_{i,t}$  - az előrejelzett érték az  $(i, t)$  pontban

$nT$  - az adat pontok száma a teszt halmazban

Előrejelzett értékként,  $\hat{y}_{i,t}$ , a Forecast az átlag (mean) értéket használja, a prediktor metrikáinak számításakor pedig  $nT$  az adat pontok száma a backtest ablakban.

Az RMSE a hibák értékeinek négyzeteit használja, ami felerősíti a kiugró értékek hatását, így ennek a metrikának a használata olyan esetekben relevánsabb, amikor minden össze néhány téves előrejelzés is súlyos költségekkel járhat.

### Weighted Quantile Loss (wQL)

A Weighted Quantile Loss, avagy súlyozott kvantilis veszteség egy olyan metrika, amely a modell pontosságát méri meg egy megadott kvantilisnél. Ez különösképpen hasznos olyan esetekben, amikor külön költségekkel jár mind az alulbecslés és a fölébecslés, a wQL függvény súlyának ( $\tau$ ) meghatározásával beépíthetők különböző büntetések az alul és fölébecslésre.

A veszteség függvény a következő képpen néz ki:

$$wQL[\tau] = 2 \frac{\sum_{i,t} [\tau \max(y_{i,t} - q_{i,t}^{(\tau)}, 0) + (1-\tau) \max(q_{i,t}^{(\tau)} - y_{i,t}, 0)]}{\sum_{i,t} |y_{i,t}|}$$

Ahol:

$\tau$  - egy kvantilis a  $\{0.01, 0.02, \dots, 0.99\}$  halmazból

$q_{i,t}^{(\tau)}$  - a  $\tau$  - kvantilis, amelyet a modell előrejelzett

$y_{i,t}$  - a vizsgált érték az  $(i, t)$  pontban

A kvantilisek ( $\tau$ ) a wQL-hez 0.01-től (p1) 0.99-ig (p99) mozoghatnak, és ezt a metrikát nem lehet az átlag előrejelzéshez kiszámítani.

Alapértelmezett módon, a Forecast a 0.1 (p10), 0.5 (p50) és a 0.9 (p90) percentilis értékekre számolja ki a wQL függvény értékét. Ezek a percentilis értékek a következő jelentésekkel bírnak:

- **p10 (0.1)** - A valós érték várhatóan kisebb lesz az előrejelzett értéknél az esetek 10%-ában.
- **p10 (0.5)** - A valós érték várhatóan kisebb lesz az előrejelzett értéknél az esetek 50%-ában. Ezt a percentilist úgy is ismerik, mint a medián előrejelzés.
- **p10 (0.9)** - A valós érték várhatóan kisebb lesz az előrejelzett értéknél az esetek 90%-ában.

### Mean Absolute Percentage Error (MAPE)

A Mean Absolute Percentage Error a százalékos hibák abszolút értékét átlagolja a figyel és az előrejelzett értékek között, minél kisebb ez az érték, annál pontosabb a modell.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

Ahol:

$A_t$  - a vizsgált érték a  $t$  pontban

$F_t$  - az előrejelzett érték a  $t$  pontban

$n$  - az adat pontok száma az idősorban

Az előrejelzett értékként,  $F_t$ , a Forecast az átlag előrejelzést használja. A MAPE olyan esetekben hasznos, amikor az értékek az időpontok között nagy mértékben változnak és a kiugró értékeknek jelentős hatása van.

### Mean Absolute Scaled Error (MASE)

A Mean Absolute Scaled Error az átlag hiba egy skálázási tényezővel való elosztása által kerül kiszámításra, ez a factor egy szezonálitási változótól,  $m$ , függ, amely az előrejelzés gyakorisága alapján választ meg. Itt is igaz, hogy minél kisebb ennek a metrikának az értéke, annál pontosabb a modell.

$$MASE = \frac{\frac{1}{J} \sum_j |e_j|}{\frac{1}{T-m} \sum_{t=m+1}^T |Y_t - Y_{t-m}|}$$

Ahol:

$Y_t$  - a vizsgált érték a  $t$  pontban

$y_{t-m}$  - az előrejelzett érték a  $t - m$  pontban

$e_j$  - az hiba a  $j$  pontban (vizsgált érték - előrejelzett érték)

$m$  - szezonálitási változó

Előrejelzett értékként a Forecast az átlag előrejelzést használja. A MASE olyan adathalmazoknál ideális amelyek ciklikusak vagy egyéb szezonális tulajdonságokkal rendelkeznek, például olyan termékek esetén, amelyekre nyáron magasabb, illetve télen alacsonyabb az igény, és ennek érdekében előny származhat abból, ha számításba vesszük ezt a szezonális hatást.

## Weighted Absolute Percentage Error (WAPE)

A súlyozott abszolút százalékos hiba vagy WAPE az előrejelzett értékek megfigyelt értékektől való eltérését méri, kiszámításához vessziük a megfigyelt értékek összegét és az előrejelzett értékek összegét majd kiszámoljuk e két érték közötti hibát, minél kisebb ez az érték, annál pontosabb a modell. Abban az esetben, ha egy backtesting ablakban a figyelt értékek összege minden adat pontra és elemre megközelítőleg 0, a WAPE értéke meghatározatlan. Az ilyen esetekben a Forecast a súlyozatlan abszolút hiba összeget adja vissza, amely a WAPE kifejezés számlálójában található.

$$WAPE = \frac{\sum_{i,t} |y_{i,t} - \hat{y}_{i,t}|}{\sum_{i,t} |y_{i,t}|}$$

Ahol:

$y_{i,t}$  - a vizsgált érték az  $(i, t)$  pontban

$\hat{y}_{i,t}$  - az előrejelzett érték az  $(i, t)$  pontban

Az előrejelzett értékként,  $\hat{y}_{i,t}$ , a Forecast az átlag előrejelzést használja. A WAPE sokkal ellenállóbb a kiugró értékekkel szemben az RMSE-nél, mert ez az abszolút hibát használja a négyzetes hiba helyett.

Az előbbiekben felsorolt metrikákat a Forecast önállóan kezeli, de megengedi hogy exportáljuk és ezek szerint az értékek szerint módosítsuk a prediktorunk paramétereinek.

Ezen metrikák produkálásához és a tanuláshoz szükséges paraméterek hangolásához, másként fogalmazva, a prediktorok tanításához a Forecast *backtesting*-et használ, amelynek alkalmazásához automatikusan szétválasztja az idősorainkat tanító és tesztelő halmazra. A tanító halmazból tanult modell előrejelzéseket generál minden adat pontra a teszt halmazból, majd ezeket az előrejelzett értékeket hasonlítja össze a vizsgált értékekkel a teszthalmazból, és ez alapján értékeli ki a modell pontosságát. A Forecast megengedi, hogy ezt a kiértékelést különböző előrejelzés típusok (kvantilisek vagy az átlag érték) szerint befolyásoljuk, az átlag előrejelzés (mean forecast) egy pontbecslést ad, valamint a kvantilis előrejelzés általában számos lehetséges kimenetelt biztosít.

## Beépített előrejelző algoritmusok

Bármely Amazon Forecast prediktornak szüksége van egy algoritmusra a modell tanításának érdekében, majd a tanult modell által generálódnak a metrikák és a becslések. A szolgáltatás hat darab beépített algoritmust biztosít a felhasználók számára, amelyek a gyakran használt statisztikai algoritmusoktól egészen a komplex neurális hálózatokig terjednek.

Az egyik legegyszerűbb ilyen algoritmus az *autoregresszív integrált mozgóátlag (ARIMA)*, amely egy gyakran használt statisztikai algoritmus idősorok előrejelzéséhez, amely kifejezetten hasznos lehet, olyan adathalmazok esetén, amelyek kevesebb mint 100 idősorral rendelkeznek és leképezhetőek stacionárius idősorokra. A stacionárius idősorok olyan statisztikai tulajdonságokkal rendelkeznek, mint az autokorreláció, vagy az időtől való függetlenség, és az ilyen idősorokkal rendelkező adathalmazok általában jelek és zajok kombinációt tartalmazzák. A jel szinuszos oszcillációs mintázatot mutathat, vagy szezonális összetevő is lehet. Az ARIMA egy szűrőként viselkedik a jelnek a zajtól való elválasztására, majd a jel extrapolálásával jövőbeli becsléseket tesz.

Egy másik gyakran használt algoritmus idősorok előrejelzéséhez az *exponenciális simítás (ETS)*, leginkább ez is olyan esetekben hasznos, amikor az adathalmaz egyszerűbb, azaz kevés dimenzióval rendelkezik, és fellelhető benne egy időszakos mintázat. Az ETS kiszámítja az idősor megfigyeléseinek súlyozott átlagát predikcióként, az idő múlásával exponenciálisan csökkenő súlyok felhasználásával. Az egyszerű mozgó átlagos módszerekkel szemben ahol a súlyok állandóak, a súlyok egy úgynevezett simítási tényezőtől függenek, és ezek a súlyok idővel arányosan exponenciálisan csökkenek.

A nem paraméterezett idősorok (*NPTS*) algoritmus egy skálázható, valószínűségi alapszintű előrejelző, amely kimondottan haszon lehet hiányos vagy ritka idősorok esetén. A Forecast négy algoritmus változatot is biztosít: sztenderd NPTS, időszakos NPTS, klímatológiai és időszakos klímatológiai előrejelző. Ez az algoritmus egy adott idősor jövőbeli érték eloszlását jelzi elő a korábbi megfigyelésekkel vett mintavételezéssel. Ezek az előrejelzések a figyelt értékek által vannak behatárolva. Az NPTS kifejezetten hasznos olyan esetekben, amikor az idősorok hiányosak vagy ritkák (sok nullás értékek tartalmaznak), példaként olyan elemek jövőbeli keresletének előrejelzsénél, ahol az idősoroknak sok alacsony értéket tartalmaznak. A szolgáltatás által biztosított variánsok a múltbeli értékek mintázásának módjában különböznek.

A *Prophet* egy olyan additív modellen alapuló idősor előrejelző algoritmus, ahol a nemlineáris trendeket éves, heti és napi szezonálitással illesztik be, ez erős szezonálitással és számos időszakot jegyző idősorok esetén működik a legjobban.

Az Amazon Forecast *CNN-QR, konvoluciós neurális háló - kvantilis regresszió*, algoritmusa egy saját gépi tanulási algoritmus idősorok előrejelzésére ok-okozati konvoluciós neurális hálózatok (CNN) használatával, olyan nagyméretű adathalmazokra amelyek akár többszáz idősort is tartalmaznak. Ez az algoritmus elfogadja az elem meta-adatokat, és ez az egyetlen algoritmus, amely képes jövőbeli értékek nélküli kapcsolódó idősorok kezelésére. A CNN-QR egy sequence-to-sequence (szekvenciáról szekvenciára, Seq2Seq) modell valószínűségi előrejelzésekre, amely azt vizsgálja, hogy egy előrejelzés mennyire jól rekonstruálja a dekódolási szekvenciát a kódolási szekvencia függvényében. A CNN-QR kvantilis regressziót hajt végre egy hierarchikus kauzális CNN-nel, amely tanulható jellemző-kivonóként szolgál.

Legvégül egy másik, az Amazon Forecast által szabadalmaztatott, gépi tanulásos algoritmust is rendelezésre áll, ez a *DeepAR+*, amely egy rekurrens neurális hálózat idősorok előrejelzésére. A klasszikus előrejelzési módszerek, mint például az autoregresszív integrált mozgóátlag (ARIMA) vagy az exponenciális simítás (ETS), egyetlen modellt illesztenek minden egyes idősorhoz, majd ezt a modellt használják az idősor jövőbe történő extrapolálására. Megtörténhet azonban, hogy sok egymáshoz hasonló idősor áll rendelkezésre. Ezek az idősor-csoportosítások különböző termékeket, szerverterhelést és webol-

dalkéréseket igényelnek. Ebben az esetben előnyös lehet egyetlen modell közös képzése az összes idősorron. A DeepAR+ ezt a megközelítést alkalmazza, ha az adatkészlet több száz jellemző idősorból áll, a DeepAR+ algoritmus felülmúlja a sztenderd ARIMA és ETS módszereket. A betanított modellt arra is felhasználható, hogy előrejelzéseket generálunk olyan új idősorokra, amelyek hasonlóak azokhoz, amelyekre betanították.

Az ismertetett algoritmusok képességeinek összehasonlítására az Amazon Forecast a következő táblázatot nyújtja:

	Neurális Hálózatok		Rugalmas helyi algoritmusok	Alapvető algoritmusok		
	CNN-QR	DeepAR+	Prophet	NPTS	ARIMA	ETS
Komputacionálisan intenzív tanulási folyamat	Magas	Magas	Közepes	Alacsony	Alacsony	Alacsony
Elfogad hisztorikus kapcsolódó idősorokat	✓	✗	✗	✗	✗	✗
Elfogad előre-tekintő kapcsolódó idősorokat	✓	✓	✓	✗	✗	✗
Elfogad elem metaadatokat (termék színe, márkája, stb.)	✓	✓	✗	✗	✗	✗
Elfogadja a beépített idójárás index funkciót	✓	✓	✓	✗	✗	✗
Alkalmas ritka adathalmazokhoz	✓	✓	✗	✓	✗	✗
Végez hiperparaméter optimalizációt	✓	✓	✗	✗	✗	✗
Megengedi az alapértelmezett hiperparaméterek felülírását	✓	✓	✗	✓	✗	✗

**2.2. ábra.** Az Amazon Forecast algoritmusainak összehasonlítása

#### 2.4.3. Előrejelzések Generálása

A előrejelző algoritmusának kiválasztása és tanulási folyamatának befejeződése után, készen állunk egy előrejelzést készíteni. Alapértelmezett módon az előrejelzés az adathalmaz csoportban levő összes elemre, amelyekre a prediktor modellje tanult, tartalmaz előrejelzéseket, de ennek ellenére megadhatunk egy részhalmazt is, amelynek elemeire az előrejelzést szeretnénk generálni.

Miután az előrejelzés elkészült, a jóslott eredmények exportálhatóak egy Amazon S3 bucket-be, amely tárolóként tekinthető (mellesleg egy ilyen bucket-ből kell az adatainkat is importálni), vagy pedig lekérdezhetők akár elemenként is meghatározott követelmények alapján. Az előrejelzést alapértelmezetten CSV vagy Parquet fájl formátumban mentődnek exportálás esetén. Az exportált előrejelzések gyakorisága a prediktor elkeszítéskor megadott előrejelzés gyakorisága alapján kerülnek meghatározásra. Az egyes sorokban megtalálható az előrejelzet elemre vonatkozó azonosító (item\_id), a dátum amelyre az előrejelzés készült, és az előrejelzett percentilis értékek vagy az átlag érték.

Itemd_id	Date	P5	P25	Mean	P75	P95
10150014	2021-07-01T00:00:00	64.66	835.38	1481.52	2108.62	3279.34
10150014	2021-08-01T00:00:00	-1.82	938.79	1654.32	2411.7	3619.59
10150014	2021-09-01T00:00:00	-89.1	782.82	1440.71	2184.83	3153.67

**2.3. ábra.** Példa egy CSV fájlba exportált, majd formázott előrejelzésre

**Megjegyzés:** a fenti ábrán a dátum a hónap elejét adta vissza, de ez a teljes hónapra vonatkozik.

Miután sikeresen elkészítettük a kívánt előrejelzéseket, a felhasznált erőforrásokat, mint az adathalmazok (datasets), adathalmaz csoportok (dataset groups), az adathalmaz importáló munkafolyamat (dataset import job), előrejelző (predictor), előrejelzés (forecast) és az előrejelzés importáló munkafolyamat (forecast export job) fel kell szabadítani a felmerülő és szükségtelen költségek elkerülése érdekében. Mivel a folyamatok és az erőforrások egymásra épülnek, ezért a törlésnek nagyjából az erőforrások létrehozásának fordított sorrendjében kell megtörténjen, például mivel minden előrejelzés egy prediktorból jön létre (egy prediktorból pedig akár több előrejelzés is származhat), nem lehetséges egy prediktor törlése addig, amíg legalább egy belőle készült előrejelzés is létezik.

## 3. fejezet

# Rendszer bemutatása

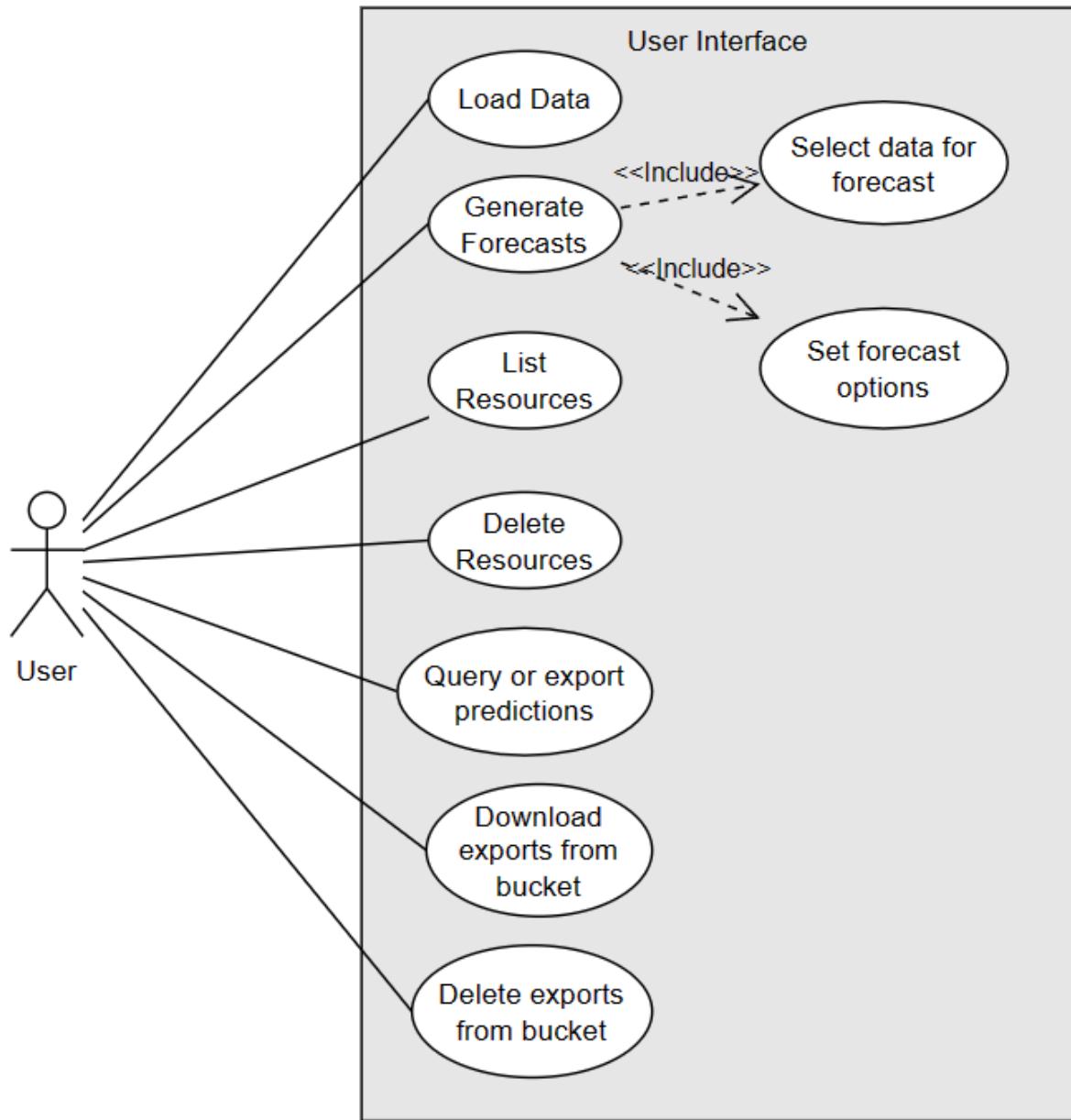
Az általam használt rendszer több alkotórészből áll. Először is szükség volt egy szoftverre, amely képes az nyers adatok beolvasására és megfelelő formájú idősorokká való átformálására, majd ezeket az idősorokat képes feltölteni egy AWS bucket-be, hogy ezekből az adatokból a megfelelő Forecast erőforrások kezelésével tegye lehetővé a előrejelzések elkészítését, illetve a lekérdezésüket. Ugyanakkor az előrejelzések eredményeit valamilyen módon szemléltetni is szükséges, és ennek érdekében készítettem egy különálló szkriptet, amelynek futtatásával képes vagyok a lekért előrejelzések elemzésére.

### 3.1. Kezelő szoftver

Az Amazon Forecast szolgáltatás használatához több lehetőség is rendelkezésre áll, használható az Amazon Web Services (AWS) weboldalán levő Amazon Forecast Konzol, az AWS Command Line Interface (CLI), ezek leginkább parancsokat és utasításokat fogadnak, egyszeri műveletekhez, tehát sok nehézséggel és ismétléssel járna a használatuk. Éppen ezért kellett egy szoftver, amely képes a folyamatok automatizálására, szerencsére az Amazon biztosít API-okat és ezek használatával felépített SDK-kat a szolgáltatásainak használatához.

Viszont ahhoz, hogy az API vagy az SDK-k [3] által kommunikálni tudjon a szoftver elengedhetetlen követelmény egy azonosító fájl *credentials* néven, amely egy Amazonos felhasználóhoz van kötve, és a kérések és utasítások ezen felhasználó nevében lesznek végrehajtva. Ennek működéséről részletesebben a [4.2](#) fejezetben.

### 3.1.1. Szoftver funkciói



**3.1. ábra.** Use case diagram - Kezelő szoftver

Az elkészített szoftver egy felhasználói felület az Amazon S3 (Simple Storage Service) és a Forecast szolgáltatások néhány, az előrejelzések elkészítéséhez szükséges funkciójának kihasználásához. Mivel az előrejelzések elkészítéséhez szükséges folyamat elég hosszú, több lépésből áll, a szoftver ezeket a lépéseket egybevonja, automatizálja a részfolyamatok egymás után való futtatásával, így ezt sokáig tartó műveletet az adatok előkészítésétől az előrejelzések eléréséig három lépésre egyszerűsíti le, hozzáadva még pár funkciót a felhasznált erőforrások menedzseléséhez.

Fontosnak tartom megjegyezni, hogy habár a folyamat az adatok importálásától az előrejelzés generálásáig teljeséggel automatikus, de igen hosszú. Éppen ezért a program majdnem bármikor leállítható, nem veszik el a folyamat állapota, mivel szoftver csak az utasítások kiadását végzi, a háttérben a szolgáltatás dolgozik. Ha egy erőforrás már elkészült, több különböző ráépülő erőforrás is eltud készülni úgy, hogy nem jön létre több ugyanolyan erőforrás, másként fogalmazva, jól meghatározott tulajdonságok alapján egy erőforrás csak egyszer jöhet létre. Példaként vegyük egy meglevő adathalmazt, egy adathalmazt importáló munkafolyamatot és egy most elkészülő prediktort. Ugyan ez az alkalmazáson belül nem látszik (vagy nem minden esetben) a szoftver felméri a kiválasztott adatok alapján, hogy ezekből az adatokból létezik-e már adathalmaz amit a szolgáltatás használni tud, és ha létezik, akkor nem szükséges az importáló folyamat, egyből elkezdheti a prediktor létrehozását a már készen levő adathalmaz alapján.

### 3.1.2. Adatbetöltés

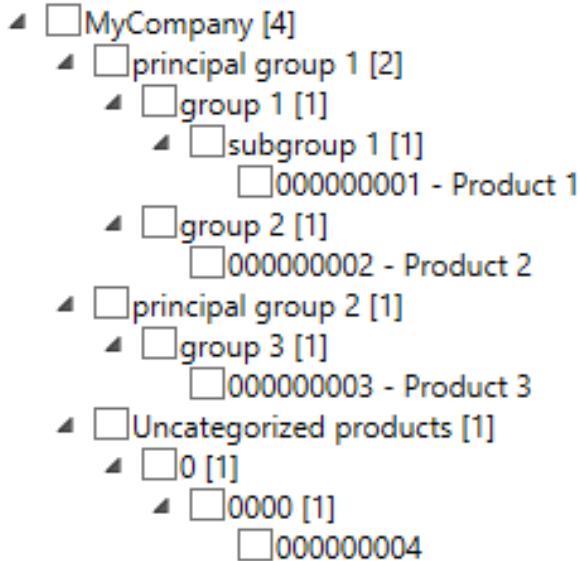
Az első szükséges lépés az előrejelzések elkészítéséhez, azon adatok betöltése, amelykből a predikciókat szeretnénk készíteni. Annak érdekében, hogy a szoftver adatbetöltése bárki által használható legyen, bizonyos konvenciókat és formai követelményeket kellett bevezetni. Ez leginkább a programba betölthető könyvtár strukturájára vonatkozik, amelyben az adatok találhatóak. Ennek a könyvtárnak tartalmaznia kell egy szöveges állományt, amelynek neve *company.txt*, és ebben meg kell adni, hogy kihez tartoznak az adatok. A fájl tartalma egy egyszerű szöveg, ami nem tartalmazhat speciális karaktereket, csak az angol ábécé betűi megengedettek. Erre azért van szükség, mert az S3 bucketben (amiről egyelőre annyit, hogy egy felhőalapú tároló) valahogyan kell különíteni az adatokat egymástól, erről részletesebben a [4.2 fejezetben](#).

A betöltésre szánt adatok főkönyvtárában lennie kell még két másik alkönyvtárnak is, az egyik ezek közül a *classification* könyvtár, amelyben szükséges egy *classification.csv* állomány. Ez a fájl az adatokat csoportokba rendezheti és név szerint azonosíthatja. A fájlban minden sorban öt darab értéknek kell lenni a következők szerint: az első oszlopban a termékek azonosítója vagy kódja, majd a neve, ez a két érték fontos a termékek megnevezése érdekében, mert például ha egy bizonyos termékről szeretnénk előrejelezni, az azonosítóját nem biztos, hogy tudjuk, de a nevére könnyebb emlékezni. A hátra maradt három oszlop a termékek bekategorizálásáért felel, az oszlopok sorrendje a főcsoport, csoport és alcsoport, ám ha az egyik érték nem ismert, akkor a sorok megfelelő oszlopaiban a *Nedefinit* jelöléssel nem veszi figyelembe az adott kategóriát. A fájlnak rendelkeznie kell fejéccel, de a fejléc tartalma nem lényeges az adatbetöltés szempontjából.

<b>Id</b>	<b>Product</b>	<b>Principal group</b>	<b>Group</b>	<b>Subgroup</b>
1	000000001 - Product 1	principal group 1	group 1	subgroup 1
2	000000002 - Product 2	principal group 1	group 2	Nedefinit
3	000000003 - Product 3	principal group 2	group 3	Nedefinit

**3.2. ábra.** Osztályozó fájl tartalma példa

A második könyvtár neve *data*, ebben találhatóak a feljegyzések az adott azonosítójú elemekről, és minden fájl az azonosítójával jegyzett CSV állomány. Az osztályozó fájlban (classification.csv) esetlegesen nem szereplő termékek is beolvasásra kerülnek *Uncategorized products* csoport alatt.



**3.3. ábra.** Fa struktúra a betöltött adatok alapján

Az adatokat ezen követelmények betartásával lehet betölteni a rendszerbe.

### Adathalmaz importálása

Az adatok importálása az a folyamat, amikor a rendelkezésre álló idősorokat adathalmazokba importáljuk, de mivel az alkalmazás szempontjából még csak nyers adataink vannak, valahogyan kell alakítani az idősorokat a kiválasztott adatokból, majd ezeket az idősorokat fel kell tölteni egy S3 bucketbe valamilyen rendszerező struktúra szerint, hogy adathalmazokba importálhassuk őket. Röviden fogalmazva meg kell oldani a nyers adatok formázását, feltöltését, egy az adatokat leíró adathalmaz létrehozását, majd a feltöltött adatok adathalmazba való importálását.

Miután az adatok sikeresen bekerültek a rendszerbe, megtörténhet, hogy rengeteg elem adatai állnak rendelkezésre, de nem mindegyikból szeretnénk az előrejelzést elkészíteni. Éppen ezért a felhasználni kívánt tételek kiválaszthatóak egy fa struktúrából. Ha a megfelelő elemeket kiválasztottuk még két beállítást kell eszközölni, ki kell választani, hogy a megjelölt adatok közül melyik időintervallumban rögzített adatokat szeretnénk felhasználni. Ennek a beállításnak főleg több kiválasztott elem esetén van értelme, amikor is az adatok rögzítése különböző időpontokban kezdődik, vagy végződik. A program több elem kiválasztásakor összeveti az adat rögzítések időpontjait, és a legszűkebb időintervallumot állítja be alapértékül, egy elem kiválasztásakor pedig az adott elem bejegyzései közül az első, illetve az utolsó dátumot.

Habár az adatok kiválasztása a megfelelő időintervallum beállításával megtörtént, az adatok még nem kerülnek feltöltésre és importálásra. Az idősorok csak azután fognak

létrejönni és feltöltésre kerülni, miután a tanulást elindítottuk, ami tulajdonképpen nem csak a tanulást, hanem a teljes folyamatot indítja el. A tanulás elindításakor először is az előrejelzéshez szükséges beállítások érvényessége kiértékelődik, majd a rendszer ellenőrzi, hogy a kiválasztott elemek kombinációja előzőleg már szerepel az S3 bucketben, ha nem, akkor kialakítja az idősort és feltölти a felhőbe, egy neki megfelelő *metadata.txt* fájlal együtt, amely alapján a későbbiekben vizsgálni lehet az idősor tartalmát. Ez a fájl lényegében az idősorban szereplő elemek azonosítóját, valamint a kezdeti és a végső dátumot tartalmazza. A kialakított idősor három oszlopot fog tartalmazni: a dátumot amikor bejegyzésre került, az értéket amit az előrejelzéssel kívánunk megbecsülni, illetve az azonosítót arról az elemről amelyhez a bejegyzés tartozik.

2020-03-31	20150	10110027
2020-04-30	7414.5	10110027
2020-05-31	27497.75	10110027
2020-06-30	42710.5	10110027
2016-01-31	39157.67	10110046
2016-02-29	52833	10110046

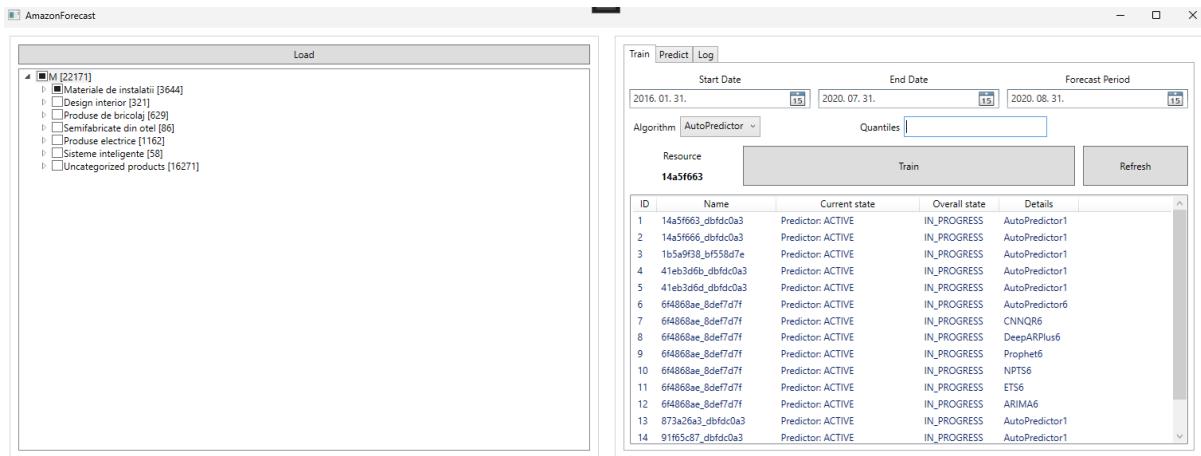
**3.4. ábra.** A kialakított idősor

Miután az idősor megtalálható az S3 bucketben, létrejöhét egy adathalmaz és az adatok bele is importálódnak. Az importáló folyamat ideje, az adatok mennyiségétől függően pár perctől, akár egy órába is beletelhet nagyon sok bejegyzés esetében. Az adathalmaz létrehozása után egy adathalmaz csoport is létrejön és az adathalmaz hozzáadódik, ugyan jelenleg csak a cél idősort tartalmazó adathalmaz fog szerepelni a csoportban, de ez egy megkötés a szolgáltatás részéről, amit nem tudunk kikerülni. Az adathalmaz importálása után következhet az előrejelző létrehozása.

### 3.1.3. Előrejelzők (Predictors)

Egy kicsit visszaugorva a folyamat elindításához, az idősor intervallumának beállításán kívül, az elindítás előtt még meg kell határozni néhány beállítást. Meg kell határozunk az előrejelzés hosszát, azaz, hogy a megadott gyakoriság szerint, hány egységet (pl. hónap) szeretnénk előrejelezni. Ezt a beállítást kiválasztott idősor intervallum alapján a rendszer, a szolgáltatás megkötései szerinti, lehető leghosszabb időtartamot állítja be, ami a kisebbik érték az 500 és az adatok kiválasztott időbeli gyakorisága szerinti hosszának az egyharmada között. Tehát ha egy elemről van 50 hónapnyi feljegyzés, akkor az előrejelzés hossza maximum  $50/3$  vagyis 16 hónap lehet. Egy másik beállítás az előrejelző, vagy prediktor, algoritmusának kiválasztása, ezeket részletesen bemutattam a 2.4.2 fejezetben. Ha kiválasztottuk a prediktor algoritmusát is, már csak a kvantiliseket kell megadnunk, legalább egy és legfennebb öt darab percentilis értéket 0.01 és 0.99 között, amely érté-

kekét pontosvesszővel választunk el. Ezek azok a kvantilis értékek lesznek, melyeket az előrejelzés vissza fog adni.



**3.5. ábra.** Az alkalmazás előrejelzés készítő ablaka

Ha a beállítások megtörténtek, és a folyamat elindult, az előbb említett módon az adatok feltöltésre majd importálásra kerülnek és az adathalmaz csoporthoz alapján létrejön egy prediktor. Az előrejelzők tanításának folyamat szintén több időt vehet igénybe, ez az adatok mennyisége és az előrejelzés hosszának fügvényében akár órákat is igénybe vehet. Ha egy prediktor elkészült, az adott modell alapján bármennyi előrejelzést készíthetünk más kvantilis értékek megadásával.

### 3.1.4. Előrejelzések (Forecasts)

A folyamat legvégén az előrejelzések generálódnak ki, ezek is külön erőforrások, amelyek a betanított prediktorok alapján készülnek el. Ha a folyamat nem volt megszakítva az előrejelzés a prediktor elkészülte után egyből létrehozódik, ami szintén egy hosszabb művelet. Úgy is letrehozhatunk egy új előrejelzést, ha éppen már van meglevő prediktorunk, ehhez a megjelenő listából ki kell választani egy prediktort. Ennek következtében a listában kijelölt sor alapján, a fa struktúrában automatikus kijelölődnek az előrejelzéshez használni kívánt elemek, és a kvantilisek beállítása után a gomb megnyomásával már elindítottuk az előrejelzés létrehozását. Ebben az esetben a már létrejött erőforrások felhasználódnak, és csak az új előrejelzés elkészülését kell megvárni.

Ha már vannak előrejelzésein, a *Predict* fülön megtekinthetjük listájukat, amelyen fellelhető az adott előrejelzés neve (ami arról adhat információt, hogy mely adatokból és melyik intervallumból készültek), a prediktor algoritmusa, az előrejelzés hossza, és a becsült kvantilis értékek. A lista egy elemének kiválasztásával lekérdezhetjük vagy exportálhatjuk az előrejelzést.

ID	Name	Algorithm	Forecast Length	Quantiles
1	41eb3d6d_dbfdc0a3	AutoPredict	1	0.26, 0.27, 0.28, 0.29, 0.3
2	41eb3d6d_dbfdc0a3	AutoPredict	1	0.96, 0.97, 0.98, 0.99
3	41eb3d6d_dbfdc0a3	AutoPredict	1	0.91, 0.92, 0.93, 0.94, 0.95
4	41eb3d6d_dbfdc0a3	AutoPredict	1	0.86, 0.87, 0.88, 0.89, 0.9
5	41eb3d6d_dbfdc0a3	AutoPredict	1	0.81, 0.82, 0.83, 0.84, 0.85
6	41eb3d6d_dbfdc0a3	AutoPredict	1	0.76, 0.77, 0.78, 0.79, 0.8
7	41eb3d6d_dbfdc0a3	AutoPredict	1	0.71, 0.72, 0.73, 0.74, 0.75
8	41eb3d6d_dbfdc0a3	AutoPredict	1	0.66, 0.67, 0.68, 0.69, 0.7
9	41eb3d6d_dbfdc0a3	AutoPredict	1	0.61, 0.62, 0.63, 0.64, 0.65
10	41eb3d6d_dbfdc0a3	AutoPredict	1	0.56, 0.57, 0.58, 0.59, 0.6
11	41eb3d6d_dbfdc0a3	AutoPredict	1	0.51, 0.52, 0.53, 0.54, 0.55
12	41eb3d6d_dbfdc0a3	AutoPredict	1	0.46, 0.47, 0.48, 0.49, 0.5
13	41eb3d6d_dbfdc0a3	AutoPredict	1	0.41, 0.42, 0.43, 0.44, 0.45
14	41eb3d6d_dbfdc0a3	AutoPredict	1	0.36, 0.37, 0.38, 0.39, 0.4
15	41eb3d6d_dbfdc0a3	AutoPredict	1	0.31, 0.32, 0.33, 0.34, 0.35
16	41eb3d6d_dbfdc0a3	AutoPredict	1	0.27, 0.28, 0.29, 0.3
17	41eb3d6d_dbfdc0a3	AutoPredict	1	0.21, 0.22, 0.23, 0.24, 0.25

2020. 07. 01. 15 2020. 07. 01. 15

Refresh      Predict      Show Exports  
Delete Export Jobs

**3.6. ábra.** Elkészült előrejelzések listája

Az ábrán látható még két dátumkiválasztó is. Ezek abban az esetben állíthatóak, ha az előrejelzés hossza nagyobb mint egy, használatukkal tudjuk lekérni az időpontokat amelyekre az előrejelzések szeretnénk látni. Például ha egy előrejelzés hossza tíz, de csak az első három hónapot szeretnénk valamilyen okból felhasználni, ezeknek a használatával megtehetjük.

### 3.1.5. Előrejelzések lekérdezése és exportálása

Az elkészült előrejelzéseket termékek szerint lekérdezhetjük, vagy az előrejelzésben levő összes elemre exportálhatjuk az S3 bucketbe. A termékek szerinti lekérdezés akkor ajánlott, ha nagyszámú elemre készült az előrejelzés, de csak párat szeretnénk belőle megnézni. Ez esetben egy hasonló fa struktúra jelenik meg a választható elemekkel, amelyek az előrejelzésben szerepelnek, és eldönthetjük az eredményeket egy vagy elemenként külön fájlokba mentésük. Ha az adatok lekérdezését választjuk akkor az eredmények egyből egy általunk kiválasztott helyre kerülnek az eszközünkön. Ha bár az összes elem kiválasztható lekérdezéskor is, nagy elemszámú előrejelzések esetén ez nem ajánlott, mert erőforrás igényes művelet és lassú lehet. A másik lehetőség az exportálás, amelynek következtében a szolgáltatás a felhőtárolóba menti ki az eredményeket, ez a művelet is eltarthat pár percig, de ez már nem a saját eszközünkön történik. Exportálás után ha látni szeretnénk az eredményeket, a 3.6 ábrán levő *Show Exports* gombra kattintva megtekinthetjük az felhőben tárolt exportokat, és letölthetünk belőlük.

Download	Resource	Details
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_001_002_003_004_005
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_006_007_008_009_01
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_011_012_013_014_015
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_016_017_018_019_02
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_021_022_023_024_025
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_026_027_028_029_03
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_031_032_033_034_035
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_036_037_038_039_04
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_041_042_043_044_045
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_046_047_048_049_05
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_051_052_053_054_055
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_056_057_058_059_06
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_061_062_063_064_065
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_066_067_068_069_07
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_071_072_073_074_075
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_076_077_078_079_08
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_081_082_083_084_085
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_086_087_088_089_09
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_091_092_093_094_095
<input type="checkbox"/>	14a5f663_dbfdc0a3	AutoPredict1_096_097_098_099
<input type="checkbox"/>	14a5f666_dbfdc0a3	AutoPredict1_001_002_003_004_005

**3.7. ábra.** Elkészült előrejelzés exportok listája

### 3.1.6. Erőforrások listázása

Az előrejelzések készítése során, mint azt már többször is említettem, több erőforrás menedzselésére is szükség van. A folyamat és az erőforrások aktuális állapotainak követése érdekében az alkalmazás rendelkezik egy lista nézettel, amelyben láthatóak az éppen létrejövő erőforrások, minden az aktuális folyamat utolsó létrejött erőforrásával a prediktorokig bezárólag. Tehát ha már egy adathalmazba bele voltak importálva az idősorok, és az adathalmaz importáló munkafolyamat lezárult, de valamilyen okból kifolyólag (például az importálás alatt bezáródott az alkalmazás) nem készült el prediktor, akkor annak a folyamatnak az aktuális utolsó elkészült erőforrása az importáló munkafolyamat, amit a későbbiekben folytatni lehet. minden erőforrásnak van egy neve, ami arra utal, hogy mely elemek kombinációjából indult ki a folyamat, valamint hogy milyen dátumokat választottak az idősorokhoz. Ezek mellett még látható, hogy éppen melyik erőforrás van készülőben, és hogy a folyamat éppen hol tart. Ha a listában az elem egy prediktor akkor a *details* mezőben az látható, hogy melyik algoritmus választásával és milyen előrejelzés hosszúsággal lett létrehozva.

Az elkészült előrejelzések egy másik oldalon, a Predict fülön ([3.6 ábra](#)) láthatóak. Az erőforrások listájához hasonlóan, itt is megtalálható az előrejelzés neve, annak a prediktornak az algoritmusá, amely alapján az előrejelzés készült, a hossza és a kvantilisek, amelyek az előrejelzésben szerepelnek. Az előrejelzések exportálásához készült exportáló folyamatok, valamint az S3 bucketbe exportált előrejelzések külön ablakokban jelennek meg ([3.7 ábra](#)), ahol az exportált előrejelzéseket letölteni és törölni, az exportáló folyamatokat pedig csak törölni lehet.

### 3.1.7. Erőforrások törlése

Az Amazon Forecast szolgáltatás erőforrásaiból egyszerre csak egy bizonyos számú megengedett. Mindegyik erőforrásnak van egy korlátja, amely meghatározza, hogy egyszerre hány darab lehet belőkük a birtokunkban (előrejelzések esetén például ez a szám 100). Ennek okán, valamint a esetlegesen felmerülő szükségtelen költségek elkerülése érdekében az erőforrásokat időnként törölni kell. Erre a célra az erőforrások listáján, ha jobb kíkkel rákattintunk egy elemre, akkor az adott elem törölhető. Prediktorok esetén egy másik opció is elérhető, amely a prediktorból elkészült előrejelzéseket fogja törölni. Egy erőforrás csak akkor törölhető, ha belőle nem származik egy vagy több más erőforrás, ugyan így az S3 bucketbe feltöltött adatok csak akkor törölhetőek, ha minden belőlük létrejött erőforrás már töröldött. Az exportált előrejelzéseket és az ezeket exportáló folyamatokat külön a nekik megfelelő ablakokban lehet törölni.

### 3.1.8. Naplózás

Az alkalmazásban még megtalálható egy *Logs* fül, amely főleg a fejlesztés során volt hasznos az esetleges hibák és kivételek kijelzésére, de ez képes olyan információkat is nyújtani, mint például ha egy erőforrás elkészült, törlődött, letöltődött vagy éppen ha a szolgáltatás megkötései miatt nem végezhető el egy új művelet (példaként említve, hogy egyszerre csak három prediktor tanítható párhuzamosan).

## 3.2. Python Scriptek

Az elkészült alkalmazás használatával képes voltam az előrejelzések készítésére, de az eredmények vizsgálatához kellett egy másik eszköz, amely által könnyedén el tudtam készíteni az ábrákat az eredmények szemléltetéséhez. A választásom a Pythonra esett, mert az olyan erős és kiforrott csomagjainak használata, mint a *pandas* [4] vagy a *matplotlib* [5] nagyon leegyszerűsítette a szükséges munkát.

A Pandas egy szoftverkönyvtár, amelyet kifejezetten a Python programozási nyelvhez fejlesztettek adatmanipuláció és analízis céljából, amely lehetővé teszi numerikus táblázatok és időszorok reprezentálását, valamint ezeken való műveletek végrehajtását. A Matplotlib pedig egy adatvizualizációs könyvtár, amely lehetővé teszi a különböző típusú grafikonok és ábrák létrehozását, rugalmas és sokoldalú eszközök szerepének felhasználásával.

A szkriptben le van implementálva az eredmények összesítése, mivel az előrejelzéseket egyszerre csak öt darab percentilis értékre lehet lekérni, ezért minden elem esetében az eredményeket összesíteni a felhasználásuk előtt. Az összesített adatokból olyan diagrammok készültek el, amelyek az eredményeket és tulajdonságaikat szemléltetik, például az előrejelzések kvantilis értékeinek kirajzolása, vagy az jövőbeli adatok eloszlásának közelítése.

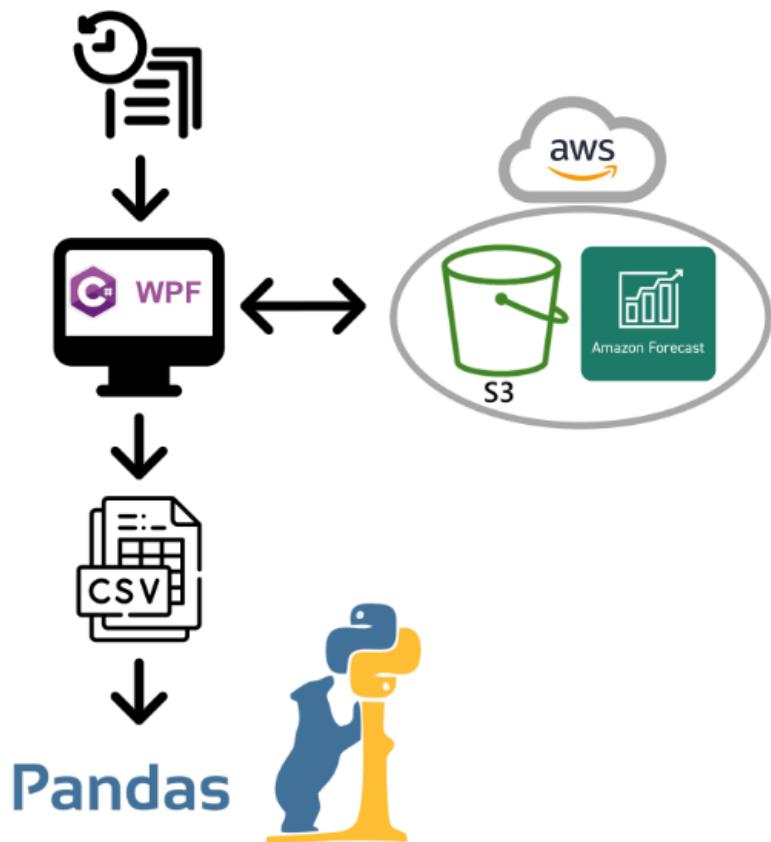
A szkirpt mellett megtalálható négy könyvtár, az egyik az előrejelzésekben felhasznált termékek feljegyzéseit tartalmazza, a másik három pedig az előrejelzéseket tartalmazzák különböző előrejelzés fajtákkal. Az előrejelzésekhez tíz darab termék adatai lettek kiválasztva, és három fajta előrejelzés típus készült belőlük, olyan amelyben az előrejelzés az összes adatra készült el növekvő percentilis értékekkel, olyanok amelyben a termékek

külön külön lettek előrejelezve ugyancsak növekvő percentilis értékekkel, és egy olyan amelyben az összes termék került előrejelzésre, de a percentilis értékek nem p01-től p99-ig növekedtek, hanem véletlenszerűen lett kiválasztva pár kvantilis és azokra készítettem előrejelzést.

## 4. fejezet

# Technológiai háttér

### 4.1. Áttekintő architektúra



4.1. ábra. Rendszer absztrakt architektúrája

Az végső rendszer három alkotó eleme között van egy WPF Frontend aplikáció, amely C# nyelvben készült az adatok előrejelzéshez való felkészítéséhez és az Amazon szolgáltatásaival való kommunikációhoz, az S3 bucket és a Forecast szolgáltatás, amelyek az előrejelzésekhez szükséges adatok és azok eredményeinek tárolását, és az előrejelzések el-

készítését biztosítják, valamit egy Python-ban írt műveletsorozat, amely az előrejelzések eredményeinek szemléltetésére alkalmas.

## 4.2. Az Amazon Forecast szolgáltatás működése

A rendszer teljes egészében a Forecast szolgáltatásra épül, használata az előrejelzések elkészítésének lebonyolításához kell, de a szolgáltatás igénybevételéhez és az API-k használatához számos előfeltétel szükséges. Az alábbiakban ismertetem a szükséges lépéseket a szolgáltatás használatához, illetve bemutatom, azt a logikát, amely szerint a szolgáltatás működik.

A Forecast és a Simple Storage Service az Amazon Web Services részét képzik, ami azt jelenti, hogy használatukhoz AWS felhasználói fiókkal kell rendelkezni. Ha már van egy AWS fiókunk, ez alapesetben root, vagyis gyökér felhasználóként létezik, amelynek az AWS szolgáltatásain belül joga van mindenhez. Viszont egy teljesen privilegizált felhasználóval dolgozni nem a legjobb ötlet a legtöbb esetben, ezért érdemes használni az AWS Identitás és Hozzáférés Kezelését (*Identity and Access Management, IAM*), amely az AWS erőforrásainak biztonságos és ellenőrzött hozzáférhetőségét irányítja. Használatával a fő fiókhoz, úgynevezett IAM felhasználókat adhatunk, amiket szerepkörökkel *role* láthatunk el. A szerepkörök különböző engedélyekből *policy* állnak az AWS szolgáltatásaihoz való hozzáférés biztosításához. Az előrejelzések készítéséhez két szolgáltatáshoz kell hozzáféréssel rendelkezni, az Amazon Forecast-hez és az Amazon Simple Storage Service-hez.

Egy IAM felhasználó a megfelelő szerepkörrel képes használni az engedélyezett szolgáltatásokat az Amazon weboldalán, de egy külső alkalmazás elkészítésekor a szolgáltatások használatához biztonsági hitelesítő adatokra van szükség. Ezek az adatok minden IAM felhasználó esetén elérhetők és ezek alapján dönti el az IAM, hogy a kérést küldő entitás rendelkezik-e megfelelő joggal a művelet végrehajtásához vagy sem. Ezek a hitelesítő adatok egy profil névből, egy hozzáférési kulcs azonosítóból, egy titkos hozzáférési kulcsból állnak, amihez tartozhat egy régió is, ami a megfelelő AWS szerverekhez irányít. A hitelesítő adatok bejelentkezés után letölthetők az AWS IAM oldaláról. Ha megvannak az hitelesítő adatok, a *credentials* fájlt el kell helyezni a fájlrendszerben a *Users||<felhasználó>||.aws||* könyvtár alatt, vagy ha Visual Studio-t használunk, akkor az AWS Toolkit kiterjesztéssel, a VS elindítása után, az Extensions fül alatti AWS Toolkit > Getting Started elemre kattintva, a megjelenő oldalon beírhatjuk a hitelesítő adatokat.

A jogok és a hitelesítő adatok elrendezése után, használhatjuk az Amazon által biztosított API-kat (Application Programming Interface) és SDK-kat (Software Development Kit), amelyeken keresztül az szolgáltatásokat kezeljük. Az én projektemhez a következő SDK-k voltak szükségesek C# ban, amelyeket a Visual Studio NuGet csomag kezelőjével szereztem meg: AWSSDK.ForecastService, AWSSDK.ForecastQueryService, AWSSDK.S3. A kommunikáció kliens objektumokon keresztül történik, ezek az objektumok hozzák létére a kapcsolatot az alkalmazás és az Amazon szerverei között. Ahhoz, hogy egy műveletet elvégezzünk a legtöbb esetben szükség van egy kérés (request) objektumra, ez információkat tartalmaz az erőforrás(ok)-ról, amelyhez a kérés kapcsolódik. Ha a request objektum kész van, át kell adni a kliens objektum megfelelő metódusának, így elindítva a műveletet.

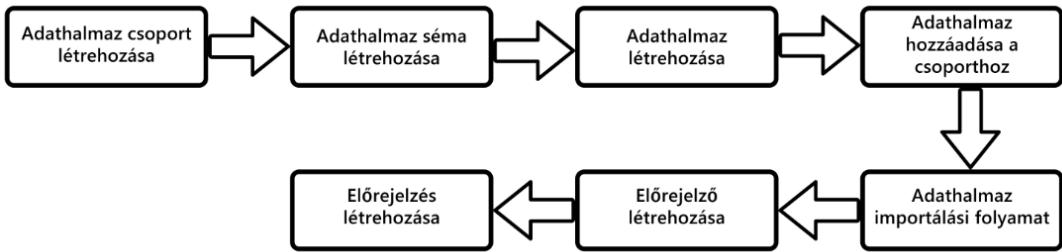
```

1 public async Task DescribeDatasetAsync(string datasetARN)
2 {
3     IAmazonForecastService forecastClient = new
4         AmazonForecastServiceClient();
5
6     var request = new DescribeDatasetRequest()
7     {
8         DatasetArn = datasetARN
9     }
10
11    DescribeDatasetResponse response;
12
13    try {
14        var response = await forecastClient.
15            DescribeDatasetAsync(request);
16        Console.WriteLine($"Status of the described dataset:
17            {response.Status}");
18    }
19    catch (AmazonForecastServiceException ex)
20    {
21        Console.WriteLine($"Unable to describe dataset '{
22            datasetARN}': {ex.Message}");
23    }
24}

```

#### 4.1. kódrészlet. Egy egyszerű kérés küldése

Az erőforrások kezeléséhez legtöbb esetben leginkább négy műveletet kell használni, a létrehozást (create), leírást (describe), listázást (list) és törlést (delete), esetenként pár kiegészítő utasítással. A create utasítások hozzák létre az erőforrásokat és egy erőforrás létrejötte után, az öt létrehozó metódus visszatér az amazonos erőforrás nevével (*Amazon Resource Name, ARN*), amely egyedileg azonosítja az erőforrást. minden erőforrás létrehozásánál, biztosítani kell az erőforrásnak egy nevet és a beállításokat, az ARN az erőforrás nevéből fog automatikusan generálódni. Ha egy művelet szeretnénk elvégezni bármely erőforrás használatával, szükség lesz az öt azonosító ARN-ra, ez által tudunk hivatkozni a felhasználni kívánt erőforrásra. Felhasználási szempontból viszont egy erőforrás csak akkor jöhét létre, ha még nem létezik egy vele azonos erőforrás ugyanazokkal a beállításokkal. Például semmi szükség két adathalmazra ugyanazokból az adatokból, hanem ehelyett a folyamat megszakadása vagy abbamaradásakor, a rendszernek észre kell vennie, hogy egy másik a létrehozni kívánt erőforrással megegyező erőforrás már használatra kész, nem szükséges újat létrehozni. Az előrejelzések készítéséhez szükséges teljes folyamat a háttérben, a következő erőforrások létrehozását foglalja magába:



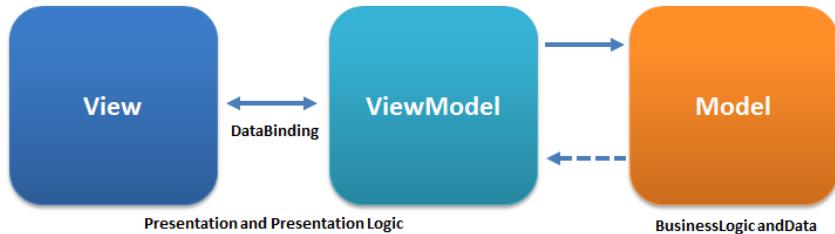
**4.2. ábra.** Előrejelzéshez szükséges lépések

A folyamat két, a 4.2 ábrán jelzett lépés között állhat le, ilyen esetekben folytatni lehet a már meglevő erőforrásokból a következő lépést. Ahhoz, hogy a program felismerje a már létező erőforrásokat, elnevezési konvenciókat alkalmaztam, a következő képpen: *<Társaság neve>\_<kiválasztott adatok azonosítóinak hashelt értéke>\_<intervallum hashelt értéké>*. Prediktorok esetén még ehhez hozzáartozik egy *<algoritmus és az előrejelzés hossza>*, az előrejelzések esetén pedig *<algoritmus és előrejelzés hossza>\_<kvantilis értékek>*. Ezek a név konvenciók biztosítják, hogy ne jöhessen létre két azonos erőforrás. Néhány erőforrás elkészülése több időbe telik, ezért a folyamatban való létrehozásukkor ügyelni kell arra, hogy ne jöjjön létre egy erőforrás, amíg az előtte levő még nincs kész. A folyamat úgy lett megvalósítva, hogy amikor a rendszer kap egy hívást egy új erőforrás létrehozásához a szükséges információkkal, ellenőrzi, hogy létezik-e már egy vele azonos erőforrás. Ha létezik egyből visszaadja az erőforrás ARN-ját, de ha nem, akkor elküldi a Forecast szolgáltatásnak a kérést az erőforrás létrehozásához. Bár a kliens objektumon keresztül egyből vissza adódik az létrejövő erőforrás ARN-ja, ez még nem használható más erőforrások létrehozására mindaddig, amíg a státusz nem aktív. Éppen ezért a rendszer nem adja egyből vissza az készülő erőforrás ARN-ját, hanem addig vár, amíg a státusz aktívvá nem változik, ezt követően folytatódhat a folyamat.

Az előrejelzéshez szükséges adatokat az Amazon S3 egy bucket nevezetű tárolójában kell tartani. A bucket ahogyan a névében is benne van, egy vödörként képzelhető el, amiben bármilyen fájl tárolható, és ahol minden adat egy helyen van, bármiféle könyvtár struktúra nélkül. Viszont a fájlok nevei alapján létrehozhatunk egy látszólagos könyvtár struktúrát a jobb átláthatóság érdekében. Ezt a strukturáltságot a '/' jelek használatával lehet elérni, ahol minden '/' lentebbi szintet jelezhet. Az előrejelzések készítéséhez szükséges idősorokat a bucketben a *trainin\_data/* előtaggal láttam el, majd ezt követte a *hashelt név/* és *hashelt dátum/*. Ezzel azt tudtam elérni, hogyha két előrejelzés ugyanazokból az adatokból történne de más intervallumból indulna ki, a strukturáltságok tekintve egy hashelt név alá kerülnének, de más hashelt dátumokba. Hasonlóan jártam el az előrejelzések exportjainál is, amiket a *forecasted\_data/* jelzés előz meg, majd a *hashelt név\_hashelt dátum* után megtalálhatóak az adatokból kinyert előrejelzések.

### 4.3. Frontend

Az Amazon Forecast kezeléséhez készült asztali alkalmazás a Windows Presentation Foundation (WPF) grafikus keretrendszer felhasználásával készült, betartva az Model-View-ViewModel (MVVM) tervezési minta megkötéseit.



**4.3. ábra.** MVVM Architektúra

A WPF egy grafikus keretrendszer a Microsoft .NET platformhoz, amely felhasználi felületek készítését és Windows alkalmazások fejlesztését teszi lehetővé. Az alapvető jellemzői közé tartozik a XAML (Extensible Application Markup Language) használata, ami egy deklaratív jelölőnyelv a felhasználói felület elemeinek leírására. Ugyanakkor a WPF támogatja a nagyobb méretű alkalmazások moduláris felépítését [6].

Az MVVM [7] egy tervezési minta, kifejezetten felhasználói felületekkel rendelkező alkalmazások fejlesztésére. Az MVVM különválasztja a felhasználói felülelet (View) az alkalmazás logikájától (Model) és közvetítő elemként használ egy ViewModel réteget, amely a kapcsolat a View és a Model között. Az MVVM előnyei közé tartozik a fejlesztési folyamat jobb felosztása, a könnyű tesztelhetőség, a felületi elemek egyszerű újrafelhasználása és a fejlesztői csapatmunka támogatása. Az MVVM architektúrában a Model tartalmazza az alkalmazás üzleti logikáját és adatelérési réteget, a View a felhasználói felületet jelenít meg, míg a ViewModel kapcsolatot tart a View és a Model között. A ViewModel felelős a felhasználói felület állapotának és az adatoknak a kezeléséért, és támogatja a parancsokat és eseményeket, amelyek interakciót tesznek lehetővé a felhasználó és az alkalmazás között. A WPF és az MVVM együttműködése lehetővé teszi a hatékony és jól strukturált alkalmazások fejlesztését. A WPF használatával könnyedén építhetők felhasználóbarát felületek, míg az MVVM a fejlesztőknek segít a kód strukturálásában és tesztelhetőségében.

Az elkészült frontend alkalmazásban a View-ok (nézetek) a háttéri adatokkal ViewModelek által vannak összekötve az alábbiak példához hasonlóan.

```

1  <UserControl x:Class="UserInterface.Views.ForecastListView"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6      xmlns:local="clr-namespace:UserInterface.Views"
7      xmlns:viewmodels="clr-namespace:UserInterface.ViewModels"
8      xmlns:behaviours="http://schemas.microsoft.com/xaml/behaviors"
9      mc:Ignorable="d"
10     d:DesignHeight="450" d:DesignWidth="800"
11     d:DataContext="{d:DesignInstance Type=viewmodels:ForecastListViewModel}">
12     <Grid>
13         <ListView ItemsSource="{Binding ListSource}" HorizontalContentAlignment="Center" SelectedItem="{Binding SelectedItem}">
14             <ListView.View>
15                 <GridView>
16                     <GridViewColumn Width="30" Header="ID" DisplayMemberBinding="{Binding ID}"/>
17                     <GridViewColumn Header="Name" DisplayMemberBinding="{Binding Name}"/>
18                     <GridViewColumn Header="Algorithm" DisplayMemberBinding="{Binding Algorithm}"/>
19                     <GridViewColumn Header="Forecast Length" DisplayMemberBinding="{Binding ForecastLength}"/>
20                     <GridViewColumn Header="Quantiles" DisplayMemberBinding="{Binding ForecastType}"/>
21                 </GridView>
22             </ListView.View>
23             <behaviours:Interaction.Triggers>
24                 <behaviours:EventTrigger EventName="SelectionChanged">
25                     <behaviours:InvokeCommandAction Command="{Binding ListSelectionChanged}"/>
26                 </behaviours:EventTrigger>
27             </behaviours:Interaction.Triggers>
28         </ListView>
29     </Grid>
30 </UserControl>

```

4.4. ábra. Egy View szerkezete

A 4.4. ábrán levő View-t egy *UserControl* komponens hozza létre és a 11. sorban levő *d:DataContext* által van összekötve egy ViewModel-el. A UserControlban egy lista nézet van, amit *binding*-al köt rá a ViewModel *ListSource* nevű adattagjára, amiben a lista elemei találhatóak. A listában a tárolt Model-ek lesznek láthatóak, amelyek közül kilehet választani bármelyik elemet, és annak adatai a *SelectedItem* adattagban lesznek tárolva egy *Command*, vagy másnéven parancs, által. Az ilyen parancsok *RelayCommand* típusú objektumok, amelyek a ViewModel-ekben vannak implementálva, és egy View-on látható elem aktiválásakor kiváltódnak és végrehajtják a hozzájük kötött eseményt. Attól viszont, hogy egy esemény végrehajtódik és esetlegesen megváltozik néhány vagy valahány objektum és modell állapota, a változások nem fognak automatikusan tükröződni a View-on. Ennek érdekében a ViewModel objektumok kiterjesztenek egy *ViewModelBase* típusú absztrakt osztályt, amely implementálja az *INotifyPropertyChanged* interfészt, és az interfészben levő *RaisePropertyChanged* függvényt, amely egy elem módosításakor a meghívása által szól a View-nak, hogy frissítse az elemhez tartozó adatokat.

```

12 references
public abstract class ViewModelBase : INotifyPropertyChanged
{
    #region Propchanged
    public event PropertyChangedEventHandler? PropertyChanged;
    29 references
    protected void RaisePropertyChanged([CallerMemberName] string propertyName = null!)
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }
    #endregion
}

```

4.5. ábra. A *ViewModelBase* absztrakt osztály

```
private ObservableCollection<ExportJobModel> _exportJobsList;
5 references
public ObservableCollection<ExportJobModel> ExportJobsList
{
    get { return _exportJobsList; }
    set
    {
        _exportJobsList = value;
        this.RaisePropertyChanged();
    }
}
```

4.6. ábra. A RaisePropertyChanged működése

## 5. fejezet

### Kiértékelés

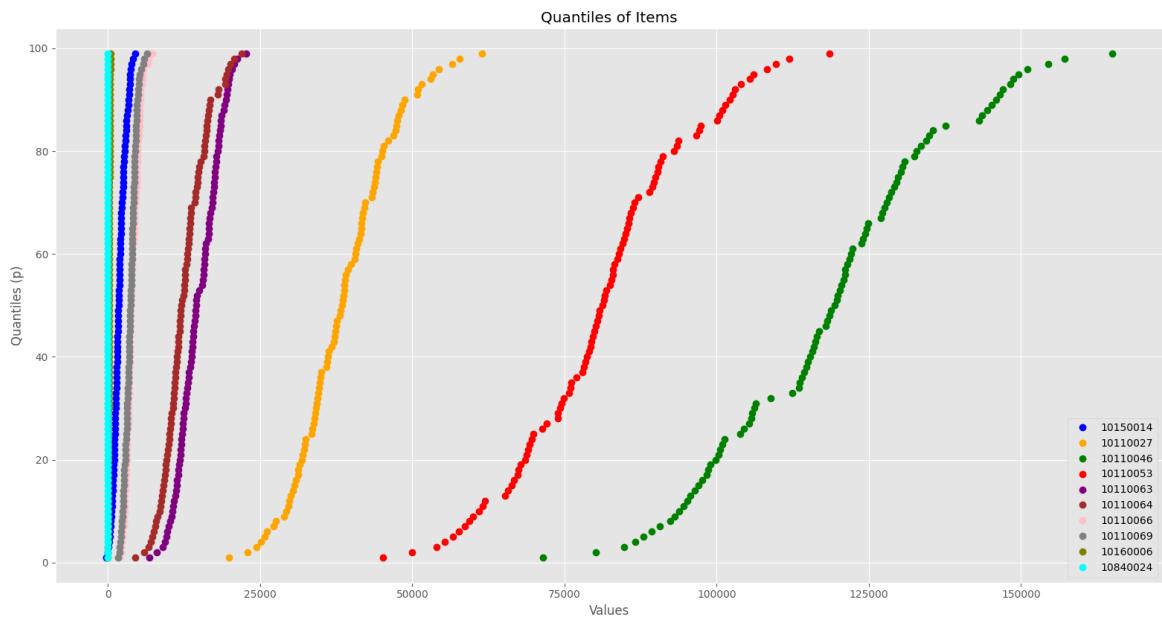
#### 5.1. Az Amazon Forecast által generált előrejelzések kiértékelése

Ahogy azt már a dolgozat során említettem az Amazon Forecast előrejelzései kvantilis értékeket képesek visszaadni és egyszerr csak öt darabot. A kvantilisek olyan statisztikai mutatók, amelyek segítenek az adatok eloszlásának és relatív pozíciójának megértésében. Ezek százalékos értékek amelyek azt mutatják meg, hogy az adatok hány százaléka helyezkedik el alattuk, például a 75. percentilis (p75) azt jelenti, hogy az adatok 75%-a ez alatt az érték alatt van. Az előrejelzések szempontjából viszont úgy kell értelmezni, hogy a p75 azt adja meg, hogy az esetek 75%-ban a valós érték várhatóan kisebb lesz az előrejelzett percentilis értéknél. Minél nagyobb a percentilis annál nagyobb a hozzá tartozó konfidencia szint.

Az előrejelzéseket egy hónapra előre készítettem, és elkészítésükhoz tíz darab termék adatai lettek kiválasztva 2016-01-31-től 2020-06-30-ig és három féle előrejelzés típus készült el. Egy amelyben a modell tanításakor minden termék adatai összesítve kerültek az adathalmazba, egy másik amelyekben a modellek a termékek adataira külön-külön lett betanítva és lekérdezve. Ebben a két típusban az előrejelzések a p01-es percentilistől a p99-ig, ötösével kerültek kigenerálásra. Az utolsó típus, amelyből csak egy készült, az első típushoz hasonlóan az összes termék adataiból készült, viszont a percentilis értékek véletlenszerűen lettek kiválasztva. Az előrejelzések tehát a következő formákban készültek el minden percentilis értékre: mind a tíz elem egyszerre (E10), egyszerre egy elem tíz alkalommal (E1).

Az összes szükséges előrejelzés kigenerálása után a Python szkript használatával különböző diagrammokat készítettem el az eredmények vizualizálásához. A diagrammokon látható görbék a kvantilis görbék. A görbe alakja és iránya a kvantilisek eloszlását és a mintaadatok jellemzőit tükrözi. Például, ha a kvantilisek között kis távolság van, akkor a görbe inkább meredek és szűk lesz, ami arra utal, hogy az előrejelzett értékek nagyobb valószínűséggel esnek a közelükben lévő tartományokba. Ha viszont a kvantilisek között nagy távolság van, akkor a görbe laposabb és szélesebb lesz, ami azt jelenti, hogy az előrejelzett értékek nagyobb szórásban helyezkednek el.

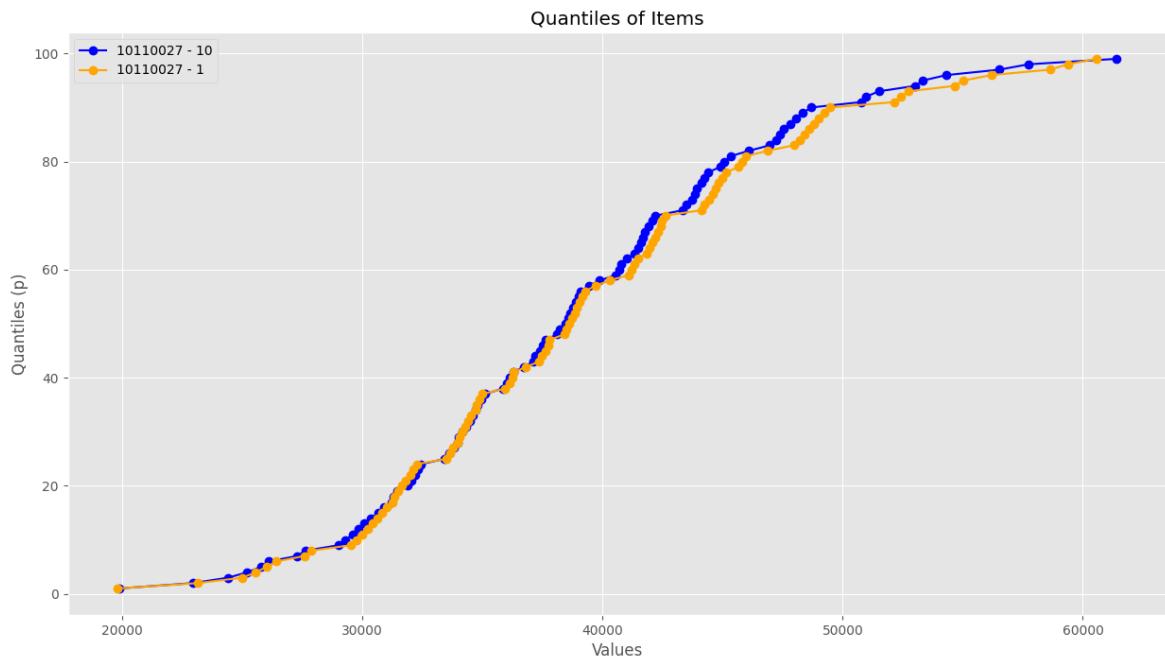
Az alábbi dia-grammon az egyszerre tíz termék adatai alapján készült előrejelzések percentilis értékei vannak egymáshoz viszonyítva elhelyezve.



**5.1. ábra.** Termékek kvantilis előrejelzései

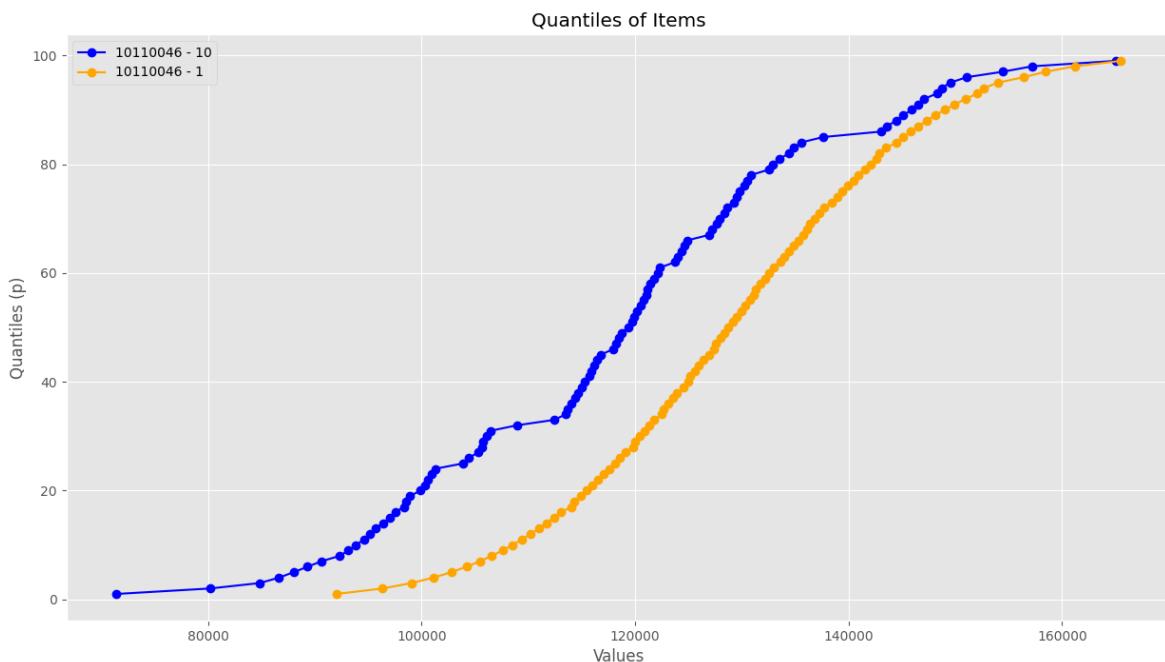
A fenti (5.1) ábra szemlélteti, hogy a termékekhez tartozó kvantilis előrejelzések növekvőek, ha ez nem így lenne akkor az összeférhetetlenséget jelentene az előrejelzések között (egy kvantilis értéke kisebb valószínűséggel rendelkezik egy nála kisebb kvantilis értékénél, vagy fordítva).

Az előrejelzések közül összevetettem az E1-ben lekérdezett termékek eredményeit az E10-ben kapott megfelelő termék eredményeivel. Az alábbi ábrákon láthatóak a különbségek a két eset között, az E10 eredményei kékkel, az E1 eredményei sárgával vannak jelölve.



**5.2. ábra.** Előrejelzések összehasonlítása az 10110027-es azonosítójú termék esetén

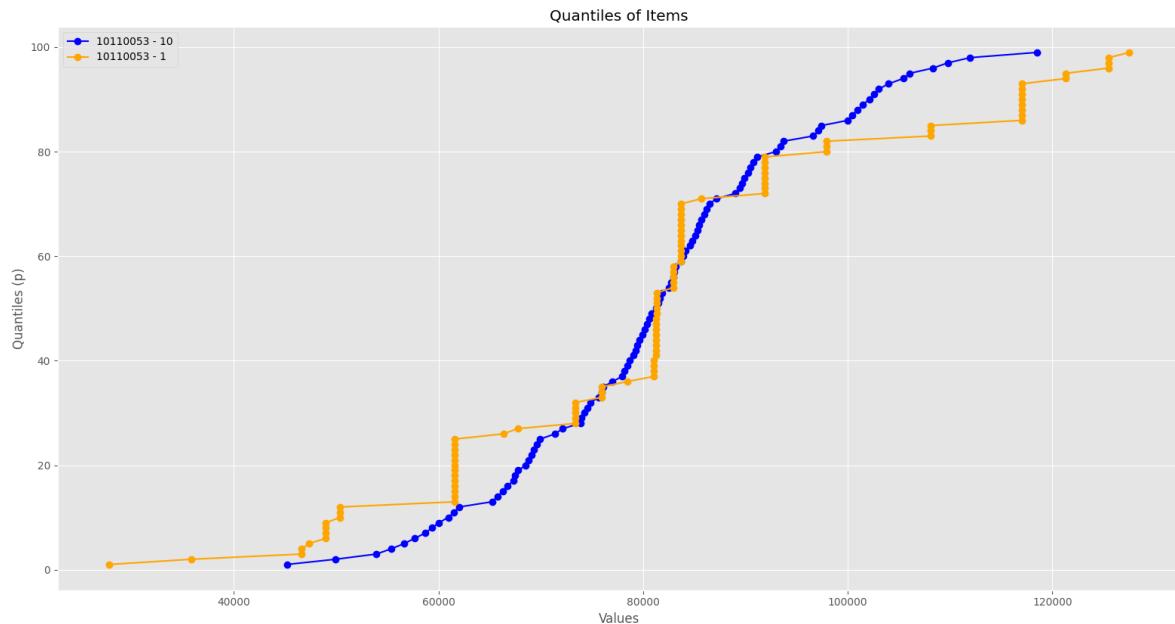
Az 10110027-es azonosítójú termék esetén (5.2. ábra) a két előrejelzési típus között nagyságrendileg nincs hatalmas különbség, a pont görbék látszólag elég jól fedik egymást, a p55-p56-os értékekig az előrejelzett percentilisek majdnem megegyzenek, elég közeli értékek.



**5.3. ábra.** Előrejelzések összehasonlítása az 10110046-os azonosítójú termék esetén

Az 5.3-es ábrán viszont már más a helyzet, a két pont görbénak csak a p99-es percentilis értéke hasonló, az E1-es esetében a percentilis értékek jóval nagyobbak az E10-énél és egy szűkebb intervallumot fednek le.

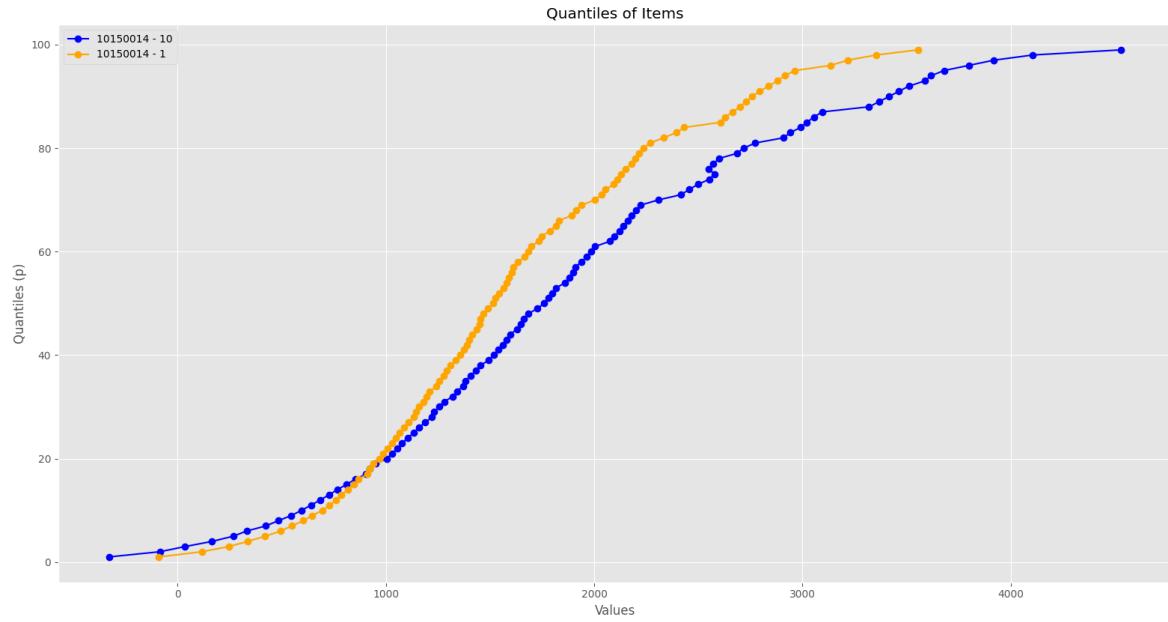
Egy másik érdekes eset az 10110053-as azonosítójú termék E1-es előrejelzéseinél keletkezett az alábbi ábra szerint.



**5.4. ábra.** Előrejelzések összehasonlítása az 10110053-as azonosítójú termék esetén

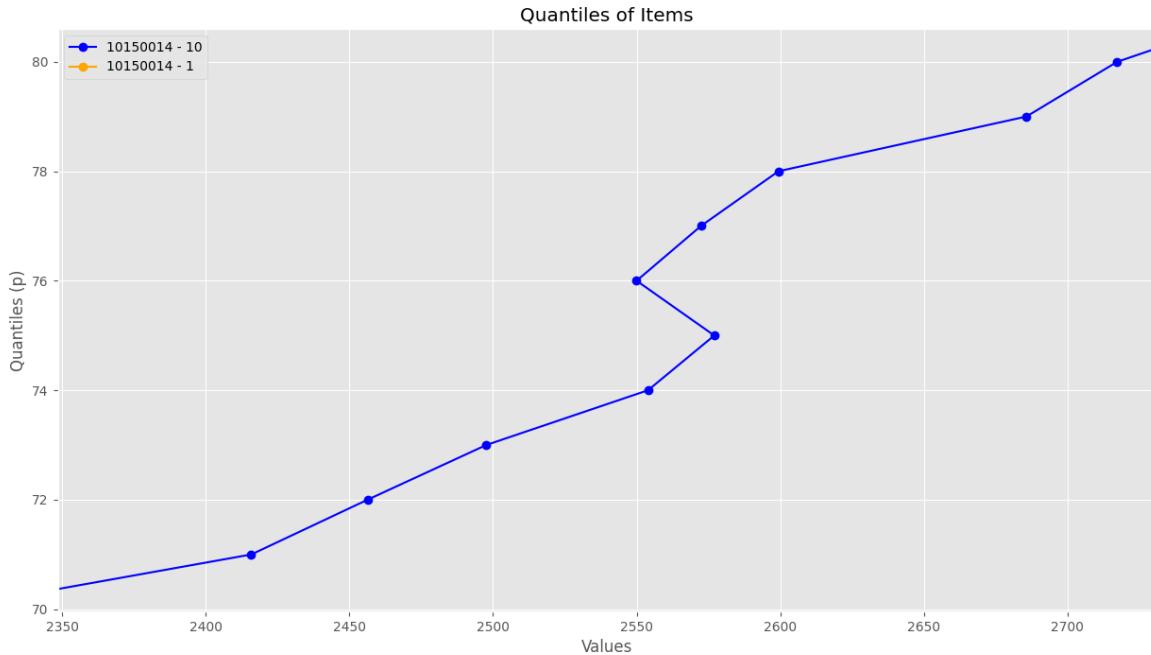
Az 5.4-es ábrán egyből szembetűnik, hogy az E1-es előrejelzésnél a pont görbe lépcsőzetes, bizonyos egymást követő percentilis értékek megegyeznek. Ez jelentheti azt, hogy az adott időpontra vonatkozóan az eloszlás meglehetősen stabil vagy hasonló marad az adott kvantilis értékeken.

Mint látható volt eddig, az Forecast által generált előrejelzések következetesek voltak nem volt ellentmondás a percentilis értékek között. Viszont a következő előrejelzésnél egy eddig szokatlan dolgot vehetünk észre.



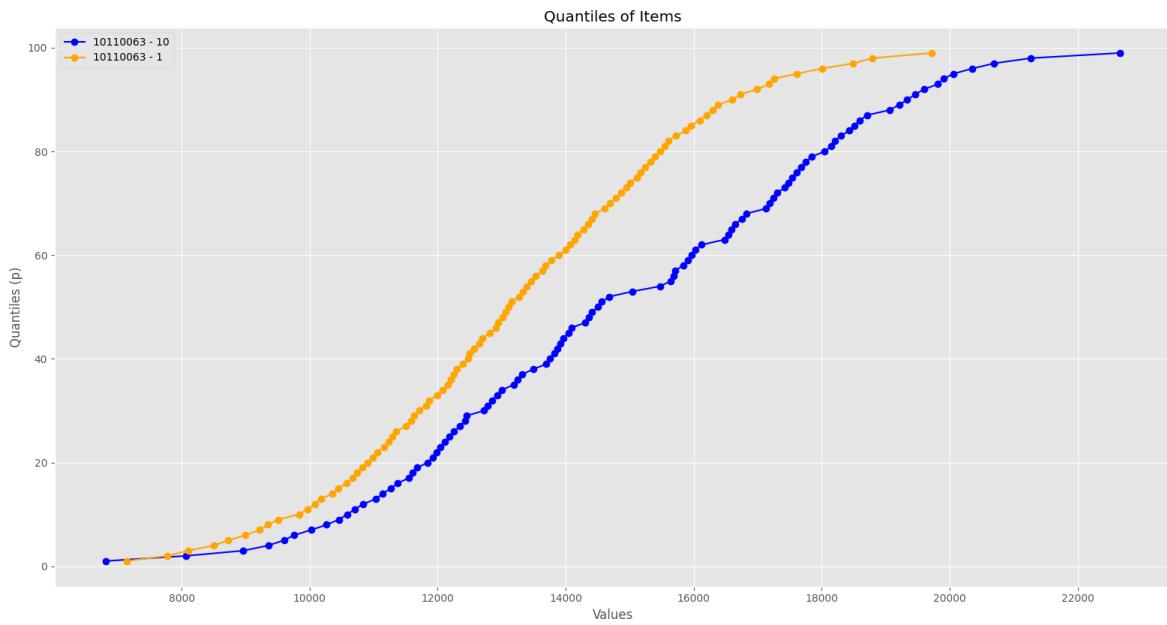
**5.5. ábra.** Előrejelzések összehasonlítása az 10150014-as azonosítójú termék esetén

A 5.4-as ábrán észrevehető, hogy a p76-os érték kisebb a p75-ös értéknél, már pedig a valóságban ilyen eset nem létezhet. Ez azt jelenti, hogy a kvantilisok nem monoton növekvő sorrendben vannak, ami ellentmond a kvantilisok természetes tulajdonságának.

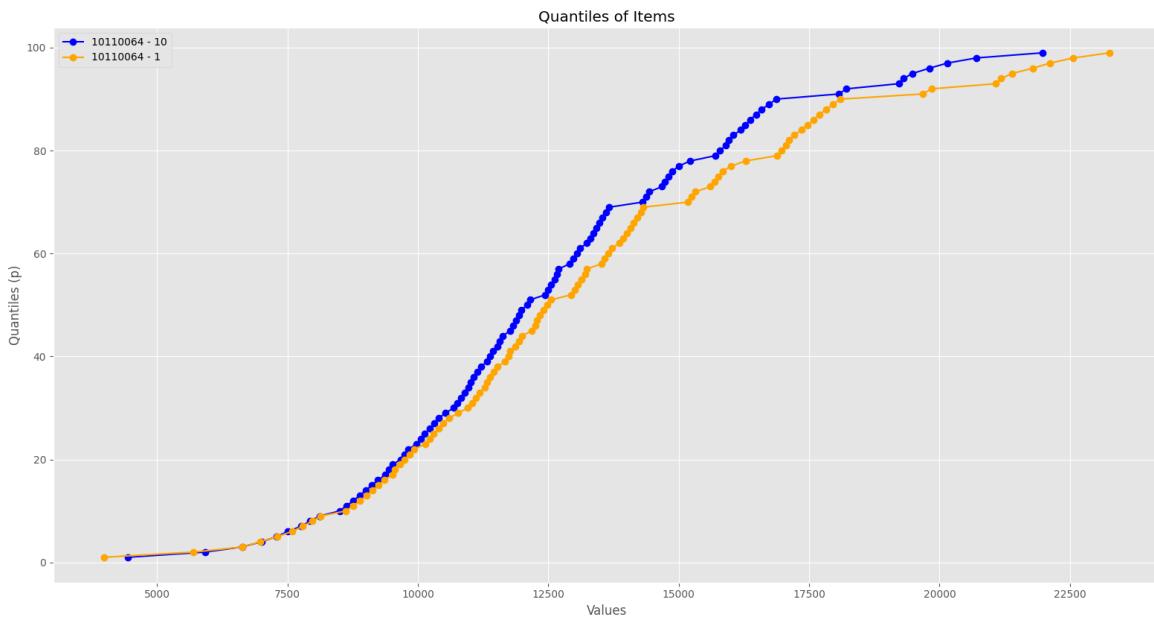


**5.6. ábra.** Kvantilis kereszteződés

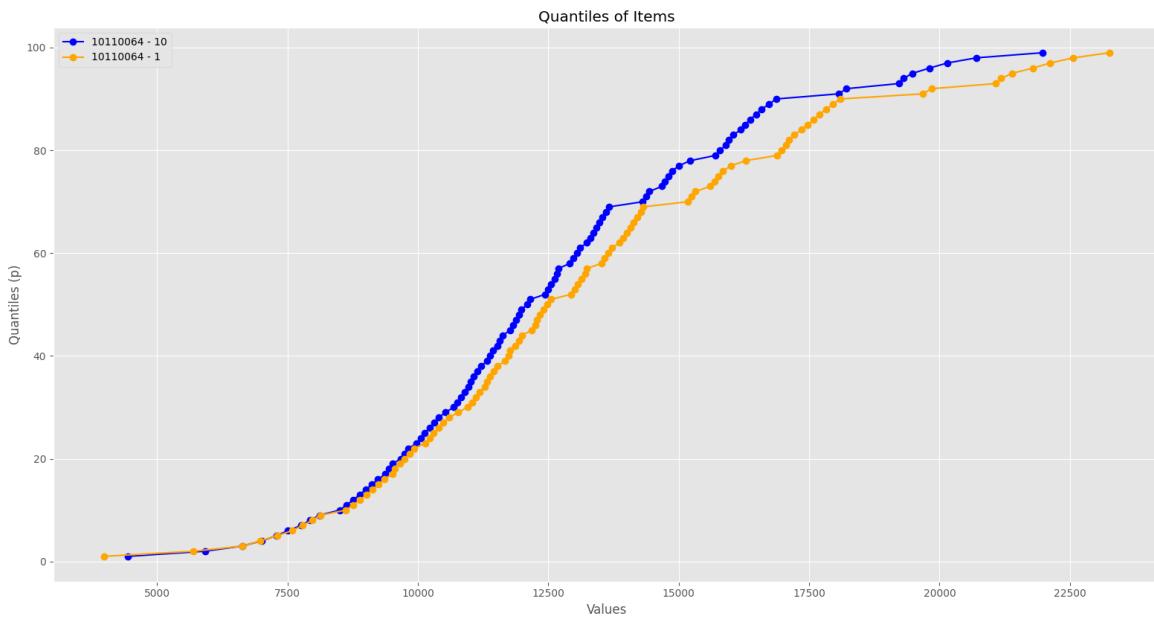
A továbbiakban még megmutatok pár összehasonlítást a többi termék előrejelzéseire, de ezeknél nincs különösebb érdekesség.



**5.7. ábra.** Előrejelzések összehasonlítása az 10110063-as azonosítójú termék esetén



**5.8. ábra.** Előrejelzések összehasonlítása az 10110064-as azonosítójú termék esetén

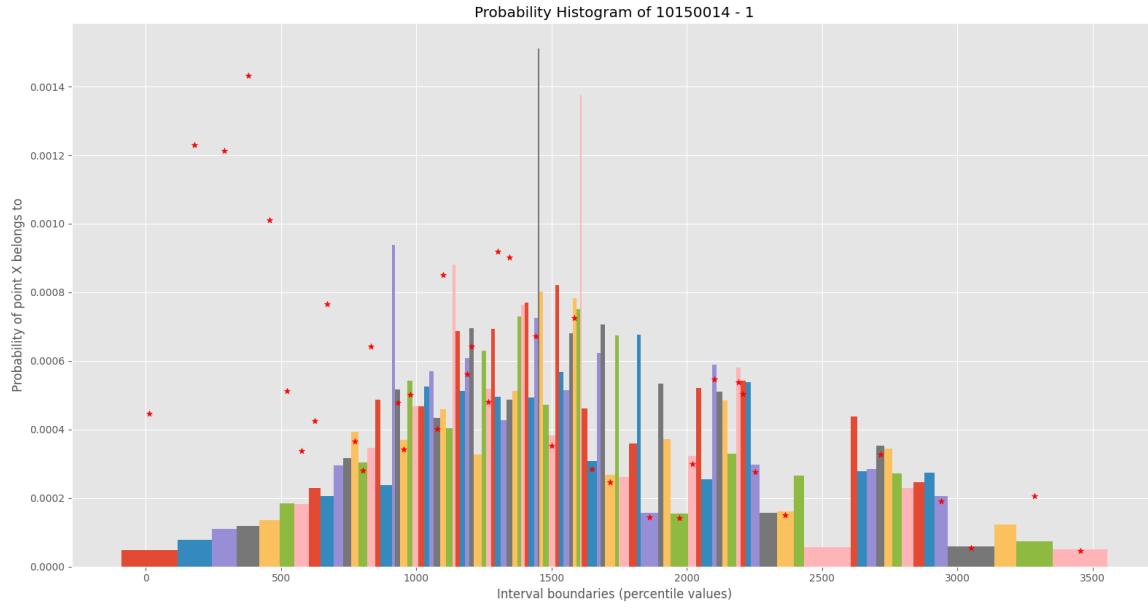


**5.9. ábra.** Előrejelzések összehasonlítása az 10110066-as azonosítójú termék esetén

A fenti ábrákon, amikor a kvantilis értékek közel vannak egymáshoz, az azt jelenti, hogy az előrejelzett értékek között kisebb változatosság van, ami utalhat arra, hogy a modell meglehetősen biztos és összehangolt előrejelzést adott. Ha viszont a kvantilis értékek távolabb vannak egymástól, az azt jelenti, hogy az előrejelzett értékek között nagyobb változatosság van. Ez arra utalhat, hogy a modell bizonytalanabb vagy kevésbé konzisztens előrejelzést adott. Egy szélesebb kvantilis tartomány jelzi, hogy nagyobb a bizonytalanság az előrejelzett értékek körül, és a tényleges érték valószínűleg eltérhet az előrejelzett tartománytól.

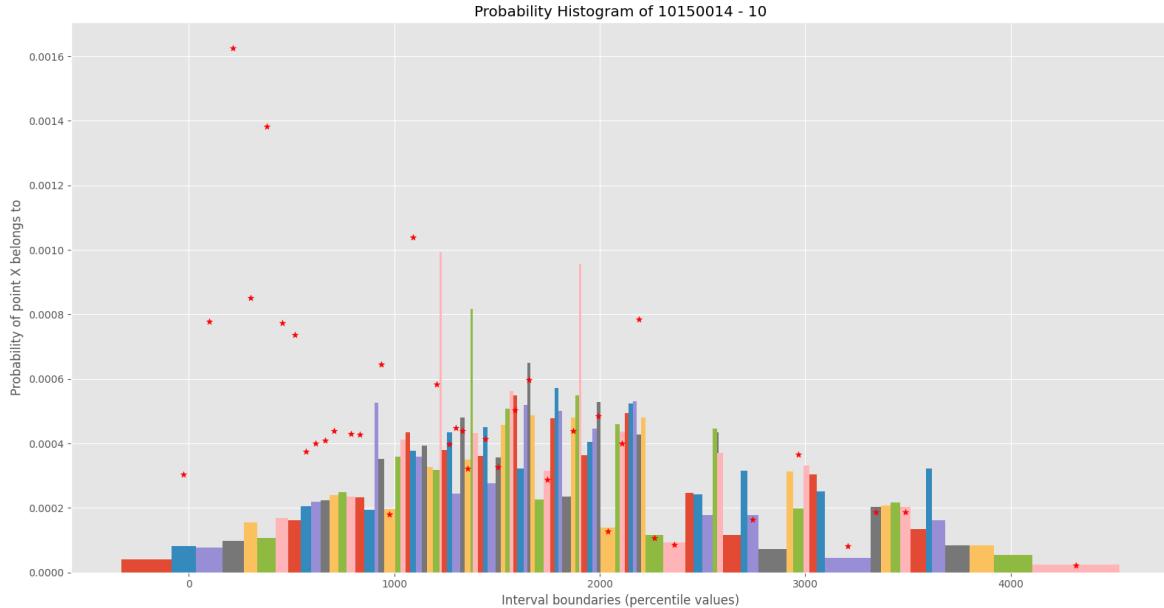
A kvantilisok közötti távolság tehát az előrejelzés bizonytalanságára utalhat. Ha kisebb bizonytalanságra van szükségünk, akkor a közelebbi kvantilisek jelenthetnek pontosabb előrejelzést. Ha viszont nagyobb bizonytalanságra van szükségünk, vagy a valósághoz való hűséget szeretnénk hangsúlyozni, akkor a távolabb eső kvantilisek jelenthetnek hasznos információt.

A kvantilisok és az idősorban levő adatok ismeretében ábrázoltam a valószínűségi eloszlás becslését a különböző termékekre. Ezt úgy oldottam meg, hogy az alsó és felső kvantilis értékek között 0.01-es lépéssel ( $[p01, p02 \dots, p99]$ ) két egymást követő kvantilis érték különbségét vettettem egy tartomány szélességének, a magasságának pedig az területet, ami egyenlő a lépésközzel, elosztottam a szélességgel. Így kaptam egy hisztogram becslést, amely intervallumaiban levő oszlopok magassága megmondja, hogy az adott intervallumba mennyi eséllyel eshet bele egy véletlen pont (X) az eladások közül. Ehhez a hisztogramhoz, igazolás képpen hozzátemtem az idősorok adatai közül a valós értékeket, amelyek beleesnek a p01 és p99 közé az adott intervallumba, az X tengelyen való pocízójuk az intervallumok közepe, a magasságuk pedig az intervallumon belüli pontok száma elosztva az adott intervallum szélességének és az összes pont mennyiségek szorzatával. Az eredmények az alábbihoz hasonlóak.



**5.10. ábra.** Valószínűségi hisztogram 10150014 - 1 esetén

A legjobb eset az lenne, ha a csillagok, amik egy véletlenszerű értéknek az adott intervallumba való kerülésének valószínűségét mutatják meg a tényleges adatok szerint, a hisztogram oszlopainak a tetején helyezkednének el, ez azt jelentené, hogy az adatok tényleges eloszlása megegyezik a valószínűségi eloszlással.

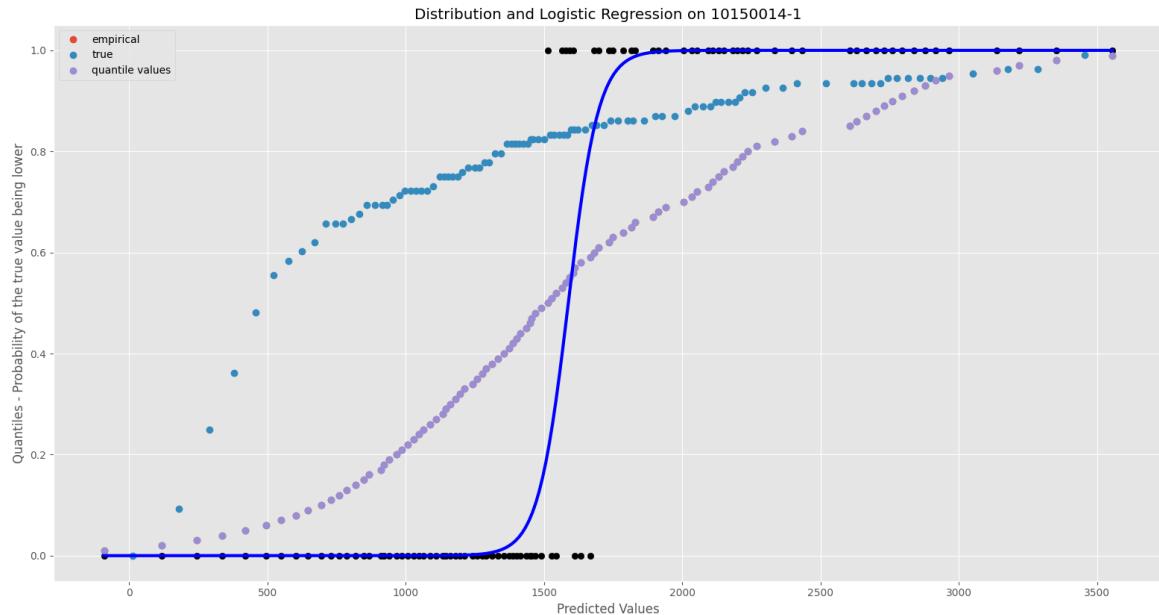


**5.11. ábra.** Valószínűségi hisztogram 10150014 - 10 esetén

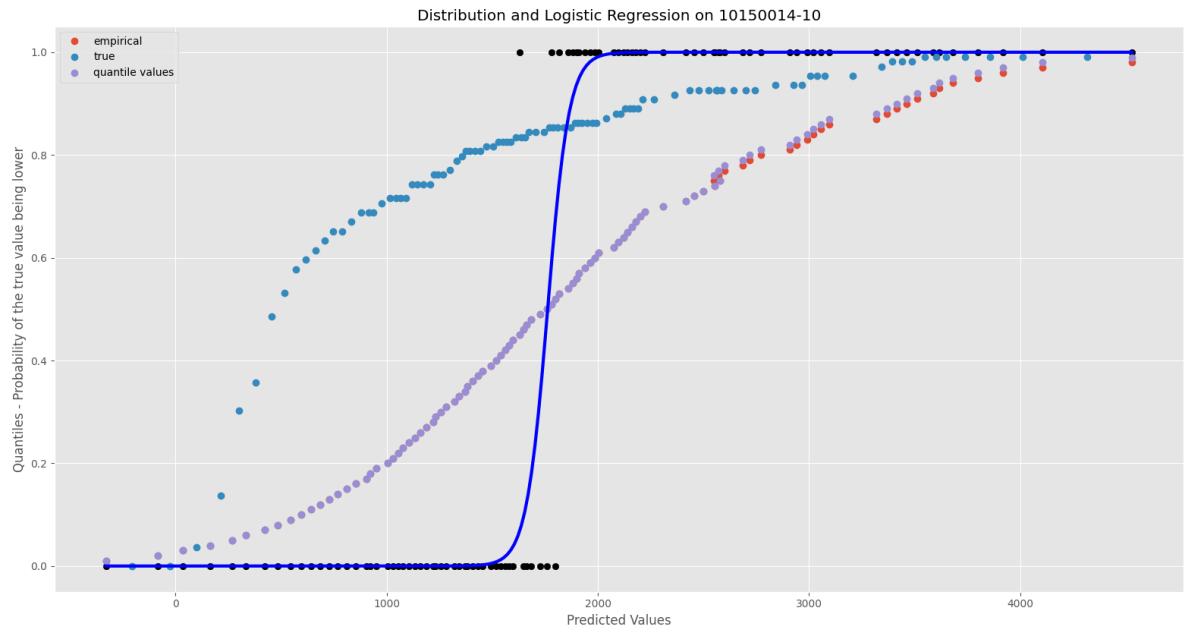
A további termékek hisztogramjai megtalálhatóak a függelékben.

Legvégül összehasonlítottam az előrejelzések kvantilis görbéjét a valós adatok kumulatív eloszlás görbéjével [8] és, majd logisztikus regressziót végeztem az előrejelzett

kvantils értékek szerint, és a regressziós görbét is ábrázoltam . Az ábrákon négy féle görbe van megjelölve, a tapasztalati görbét (piros) a valószínűségi hisztogrammok alapján ábrázoltam, úgy, hogy a bizonyos pontok y koordinátái az azt megelőző intervallumok kumulált összege, x koordinátáknak, pedig a kvantiliis értékeket vettetem. Mivel elvileg minden oszlop területe 0.01, ezért a legtöbb esetben a tapasztalati görbe egybe esik a kvantilis görbével [9]. A kék görbe a valós adatok kumulatív eloszlás görbéje, amely pontjainak y koordinátái az valószínűségi hisztogrammokon levő csillagok magasságának kumulált összege, x koordinátájuk pedig megegyezik a csillagok x koordinátáival.



**5.12. ábra.** Eloszlás és Logisztikus görbe 10150014 - 1 esetén



**5.13. ábra.** Eloszlás és Logisztikus görbe 10150014 - 10 esetén

A kvantilis értékekre elvégeztem egy logisztikus regressziót is. Mivel a logisztikus regressziót osztályozási feladatokra használják, a kvantilis értékeket osztályokba kellett sorolni, még pedig úgy, hogy minden kvantilisre generáltam egy véletlenszerű számot, amely ha kisebb lett a kvantilisnél akkor az osztálya 1 különben 0 lett, és ezt minden kvantilisra megismételtem ötvenszer, a végső osztály pedig az a szám lett, amely többször került ki a 0 és 1 között. Miután a kvantilis értékek rendelkeztek osztályokkal, a python sickitlearn-linear\_model csomagjából meghívtam a LogisticRegression() osztályt, amely előállított és betanított egy modellt az osztályozásra. Mivel a logisztikus függvény ábrázolására volt szükség, a betanult modelltől elkértem az *intercept* és az *coef* értékét, amelyek használatával ábrázolni tudtam a függvényt [10].

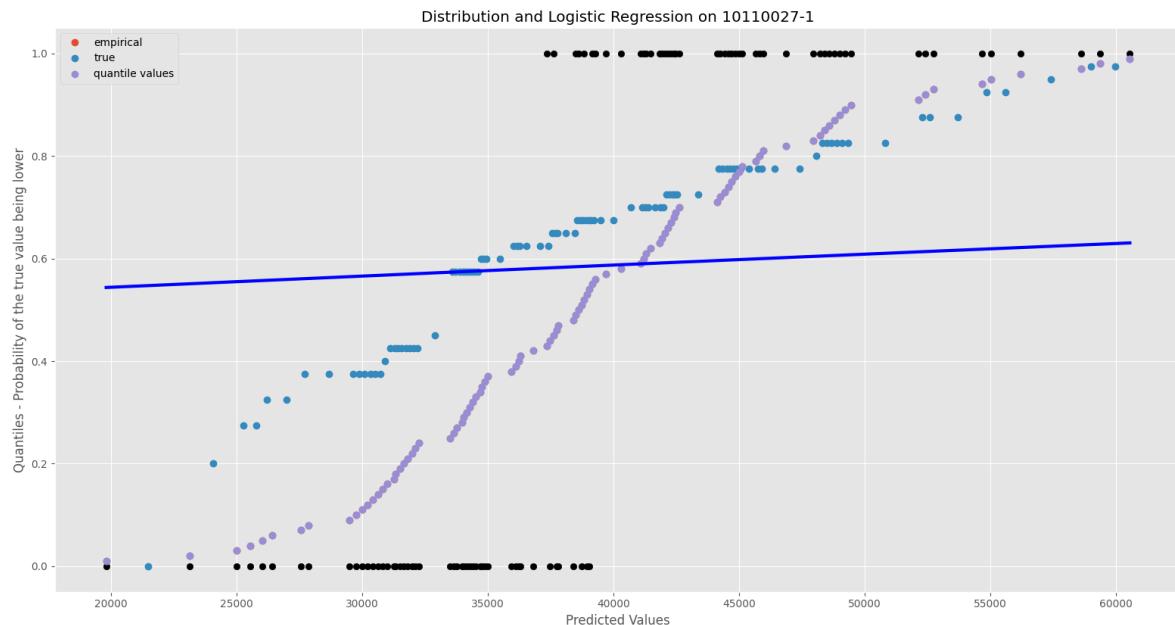
$$p(x) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x)}}$$

Ahol:

$\beta_0$  - intercept vagy eltolás

$\beta_1$  - coefficient vagy együttható

Nem minden alkalommal tanult megfelelően a regressziós modell, mint például a 5.14 ábra esetén.



**5.14. ábra.** Eloszlás és Logisztikus görbe 10110027 - 1 esetén

## 5.2. Költségek

Az Amazon Forecast szolgáltatás használata és a általa készített előrejelzések mind időbeli, minden pénzbeli erőforrásokat emészt fel. A szolgáltatás használata pénzügyi költségek szempontjából jól meghatározott, minden gigabajtnyi importált adatért \$0.088 dollárt kér, minden tanítási óráért \$0.24 dollárt, az első százezer előrejelzett érték esetén pedig ezer értékenként \$2.00 dollár az ára. A Forecast a prediktorok tanítását párhuzamosan is végezheti, emiatt a valós tanulással töltött idő nagyobb lehet az érzékelt időnél.

Az én esetemben az fájlokban tárolt adatok mérete megközelítőleg két kilóbájt körűli volt egyedi termékek előrejelzései esetén, és tizenöt kilóbájt a tíz termékes előrejelzés esetén. Összesen kilenc prediktort tanítottam az előrejelzések elkészítéséhez, amelyek elkészítéséhez általában 40 perc és 1 óra közötti időre volt szükség, ami kevesebb lehet a ténylegesen tanulással töltött időnél a párhuzamosítás miatt, az előrejelzések kigenerálásának elkészüléséhez pedig nagyjából 20 percet kellett várni. Ezeken a kívül még az adat importálásra kellett várni körülbelül fél órát egy termék esetén és egy órát a tíz termék esetén, valamint az előrejelzések bucketbe való exportálása 10 percben telt előrejelzéseként.

## 5.3. A szoftver eredményei

Mivel az alkalmazás nem végez túl sok, nagy számítási kapacitást igénylő feladatot, hanem leginkább csak vár a következő lépés elkezdésére, az szoftver által végzett műveletek megfelelő gyorsaságúak. Két megnevezhető művelet van amit a szoftver végez és mérni lehet az idejét, az adatbetöltés és az előrejelzés lekérdezés egyből saját gépre egyesével termékenként. Az adatbetöltés kifejezetten gyorsnak mondható, több mint 22

ezer elem betöltésére kevesebb mint két másodperces időt kell várni. Az előrejelzések egyből helyi tárolóra való kimentése sok esetén kivárhatatlanul sok időbe telhet.

## **6. fejezet**

### **Összefoglaló**

Dolgozatomban betekintést nyújtottam a kereslet előrejelzés fontosságába és módszereibe, és bemutattam az Amazon Forecast szolgáltatást amely gépi tanulás módszereket használva képes előrejelzéseket készíteni. A használata nem igényli a különböző mesterséges intelligencia vagy gépi tanulási módszerek ismeretét, és segítségével könnyedén el lehet készíteni előrejelzéseket különféle területeken.

Részletesen demonstráltam az Amazon Forecast működésének három fő elemét az adathalmaz importálását, az előrejelzők tanítását és az előrejelzések generálását,, amelyek folyamatával bárki képes lehet előrejelzések elkészítésére. A folyamat kezelésére elkészült asztali alkalmazást is bemutattam, amely a folyamathoz szükséges lépéseket kezeli, mint az adatok idősorokba való rendezése és feltöltése az Amazon szervereire, a különféle erőforrások létrehozása és kezelése, valamint a létrejött előrejelzések letöltése, és ezeket a lépéseket egyszerűsíti le a felhasználók számára. Az alkalmazás használatához előfeltétek szükségesek, mint az Amazon Web Servicese fiók és megfelelő jogosultságokkal rendelkező IAM felhasználó hitelesítését szolgáló tokenek és kódok.

Ezenkívül prezentáltam a szolgáltatás és az alkalmazás segítségével elkészített előrejelzéseket, amelyek eredményeit különféle ábrákon vizualizáltam python szkriptek segítségével.

#### **6.1. Továbbfejlesztési lehetőségek**

Az dolgozatban bemutatott alkalmazás jelenleg csak egyszerűbb, kevesebb dimenzióval rendelkező adatokból képes előrejelzéseket készíteni, ennek fejlesztésére a felhasználók képesek lehetnének egyedi sémákat megadni az adataik alapján. Továbbá az Amazonnak még számos funkciója van a Forecast szolgáltatásán belül is, amelyek nem annyira fontosak az előrejelzések elkészítésének szempontjából, de implementálni lehetne.

# Köszönetnyilvánítás

Nagyon szépen köszönöm témavezetőmnek, dr. Kolumbán Sándornak a dolgozat témajának ötletéért, a szolgáltatás költségeinek fedezéséért, az adatok biztosításáért, valamint azért, munkájával, szakértelmével és tudásával nagymértékben hozzájárult a diplomamunkám elvégzéséhez. Továbbá meg szertném köszönni dr. clánzan David Andreinek, az egyetemtől való kapcsolattartásért, a rámfordított idejéért, és hogy ötleteivel hozzájárult a dolgozatban leírtakhoz.

# Ábrák jegyzéke

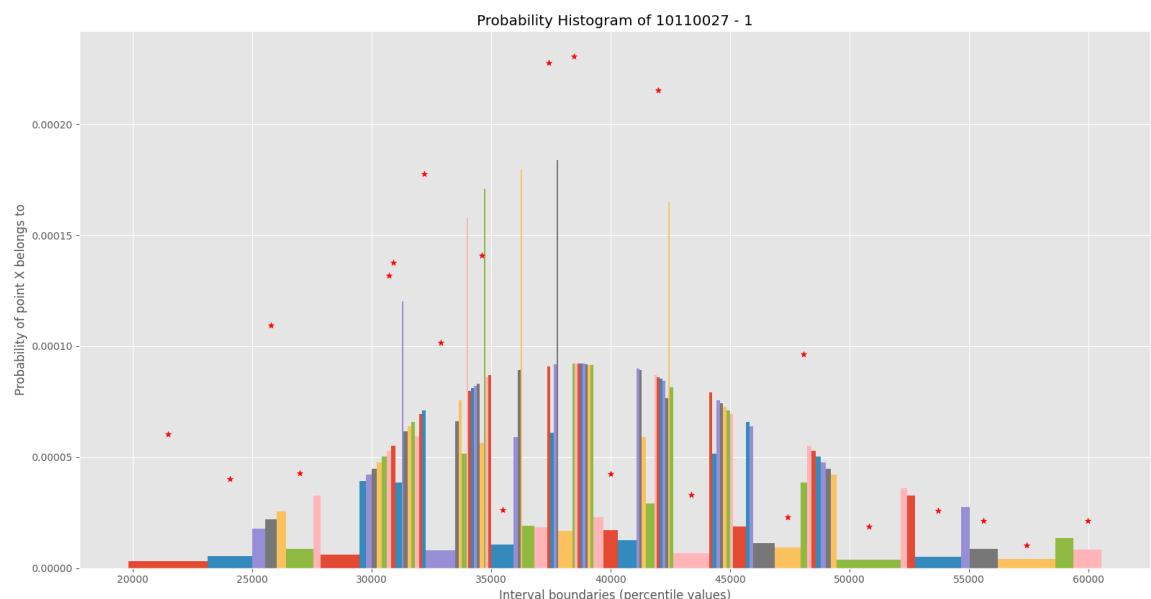
2.1. Egy termék feljegyzései . . . . .	7
2.2. Az Amazon Forecast algoritmusainak összehasonlítása . . . . .	15
2.3. Példa egy CSV fájlba exportált, majd formázott előrejelzésre . . . . .	16
3.1. Use case diagram - Kezelő szoftver . . . . .	18
3.2. Osztályozó fájl tartalma példa . . . . .	19
3.3. Fa struktúra a betöltött adatok alapján . . . . .	20
3.4. A kialakított idősor . . . . .	21
3.5. Az alkalmazás előrejelzés készítő ablaka . . . . .	22
3.6. Elkészült előrejelzések listája . . . . .	23
3.7. Elkészült előrejelzés exportok listája . . . . .	24
4.1. Rendszer absztrakt architektúrája . . . . .	27
4.2. Előrejelzéshez szükséges lépések . . . . .	30
4.3. MVVM Architektúra . . . . .	31
4.4. Egy View szerkezete . . . . .	32
4.5. A ViewModelBase absztrakt osztály . . . . .	32
4.6. A RaisePropertyChanged működése . . . . .	33
5.1. Termékek kvantilis előrejelzései . . . . .	35
5.2. Előrejelzések összehasonlítása az 10110027-es azonosítójú termék esetén .	36
5.3. Előrejelzések összehasonlítása az 10110046-os azonosítójú termék esetén .	36
5.4. Előrejelzések összehasonlítása az 10110053-as azonosítójú termék esetén .	37
5.5. Előrejelzések összehasonlítása az 10150014-as azonosítójú termék esetén .	38
5.6. Kvantilis kereszteződés . . . . .	38
5.7. Előrejelzések összehasonlítása az 10110063-as azonosítójú termék esetén .	39
5.8. Előrejelzések összehasonlítása az 10110064-as azonosítójú termék esetén .	39
5.9. Előrejelzések összehasonlítása az 10110066-as azonosítójú termék esetén .	40
5.10. Valószínűségi hisztogram 10150014 - 1 esetén . . . . .	41
5.11. Valószínűségi hisztogram 10150014 - 10 esetén . . . . .	41
5.12. Eloszlás és Logisztikus görbe 10150014 - 1 esetén . . . . .	42
5.13. Eloszlás és Logisztikus görbe 10150014 - 10 esetén . . . . .	43
5.14. Eloszlás és Logisztikus görbe 10110027 - 1 esetén . . . . .	44
F.0.1.Valószínűségi hisztogram 10110027 - 1 esetén . . . . .	51
F.0.2.Valószínűségi hisztogram 10110027 - 10 esetén . . . . .	52
F.0.3.Valószínűségi hisztogram 10110046 - 1 esetén . . . . .	52
F.0.4.Valószínűségi hisztogram 10110046 - 10 esetén . . . . .	53

F.0.5. Valószínűségi hisztogram 10110053 - 1 esetén . . . . .	53
F.0.6. Valószínűségi hisztogram 10110053 - 10 esetén . . . . .	54
F.0.7. Valószínűségi hisztogram 10110063 - 1 esetén . . . . .	54
F.0.8. Valószínűségi hisztogram 10110063 - 10 esetén . . . . .	55
F.0.9. Valószínűségi hisztogram 10110064 - 1 esetén . . . . .	55
F.0.10. Valószínűségi hisztogram 10110064 - 10 esetén . . . . .	56
F.0.11. Valószínűségi hisztogram 10110066 - 1 esetén . . . . .	56
F.0.12. Valószínűségi hisztogram 10110066 - 10 esetén . . . . .	57
F.0.13. Valószínűségi hisztogram 10110066 - 100 esetén . . . . .	57
F.0.14. Valószínűségi hisztogram 10160006 - 10 esetén . . . . .	58
F.0.15. Valószínűségi hisztogram 10840024 - 10 esetén . . . . .	58
F.0.16. Eloszlás és Logisztikus görbe 10110027-1 esetén . . . . .	59
F.0.17. Eloszlás és Logisztikus görbe 10110027-10 esetén . . . . .	59
F.0.18. Eloszlás és Logisztikus görbe 10110046-1 esetén . . . . .	60
F.0.19. Eloszlás és Logisztikus görbe 10110046-10 esetén . . . . .	60
F.0.20. Eloszlás és Logisztikus görbe 10110053-1 esetén . . . . .	61
F.0.21. Eloszlás és Logisztikus görbe 10110053-10 esetén . . . . .	61
F.0.22. Eloszlás és Logisztikus görbe 10110063-1 esetén . . . . .	62
F.0.23. Eloszlás és Logisztikus görbe 10110063-10 esetén . . . . .	62
F.0.24. Eloszlás és Logisztikus görbe 10110064-1 esetén . . . . .	63
F.0.25. Eloszlás és Logisztikus görbe 10110064-10 esetén . . . . .	63
F.0.26. Eloszlás és Logisztikus görbe 10110066-1 esetén . . . . .	64
F.0.27. Eloszlás és Logisztikus görbe 10110066-10 esetén . . . . .	64
F.0.28. Eloszlás és Logisztikus görbe 10110069-10 esetén . . . . .	65
F.0.29. Eloszlás és Logisztikus görbe 10160006-10 esetén . . . . .	65
F.0.30. Eloszlás és Logisztikus görbe 10840024-10 esetén . . . . .	66

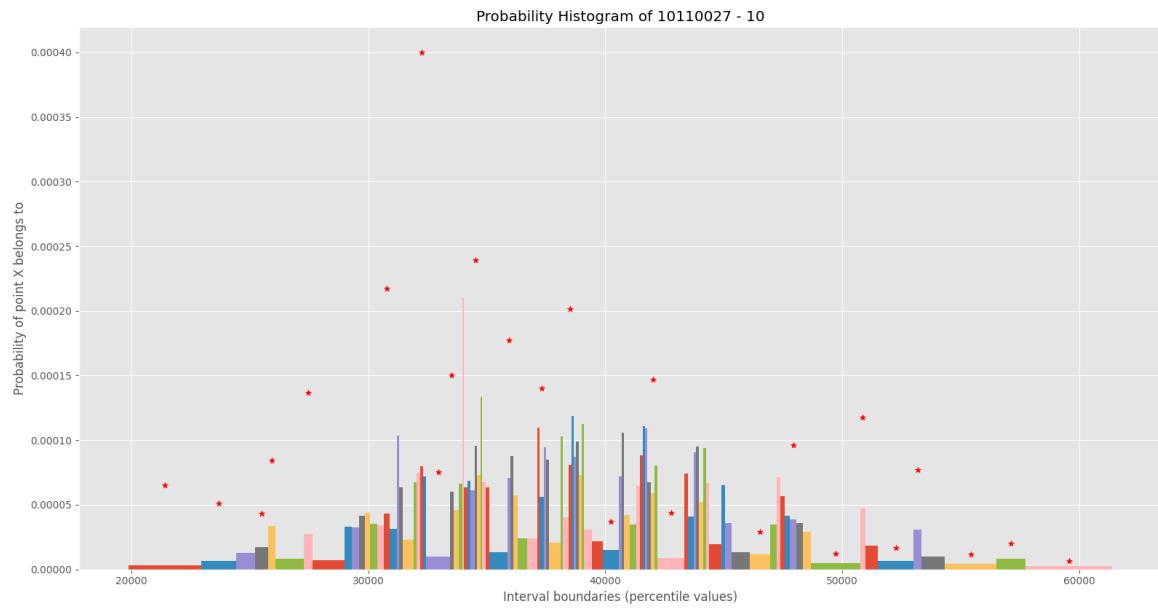
# Irodalomjegyzék

- [1] R. Hyndman and G. Athanasopoulos. (2021) Forecasting: principles and practice. Melbourne, Australia. [Online]. Available: <https://www.otexts.com/fpp3>
- [2] A. W. Services, *Amazon Forecast Developer Guide*, Amazon Web Services, 2023. [Online]. Available: [https://docs.aws.amazon.com/pdfs/forecast/latest/dg/forecast\\_dg.pdf](https://docs.aws.amazon.com/pdfs/forecast/latest/dg/forecast_dg.pdf)
- [3] „Aws sdk for .net version 3 api reference,” Online. [Online]. Available: <https://docs.aws.amazon.com/sdkfornet/v3/apidocs/Index.html>
- [4] W. McKinney *et al.*, *pandas: Powerful data analysis and manipulation library for Python*, PyData, 2020. [Online]. Available: <https://pandas.pydata.org/>
- [5] J. D. Hunter, „Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [6] Microsoft. (2023) WPF: Windows presentation foundation. Microsoft Docs. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-7.0>
- [7] Microsoft. (2016) Mvvm pattern. Microsoft Docs. [Online]. Available: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern>
- [8] A. Kujawska. (2021) Quantiles: Key to probability distributions. [Online]. Available: <https://towardsdatascience.com/quantiles-key-to-probability-distributions-ce1786d479a9>
- [9] A. B. Downey, *Think Stats: Exploratory Data Analysis in Python*, 2014.
- [10] Logistic regression on wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/Logistic\\_regression](https://en.wikipedia.org/wiki/Logistic_regression)

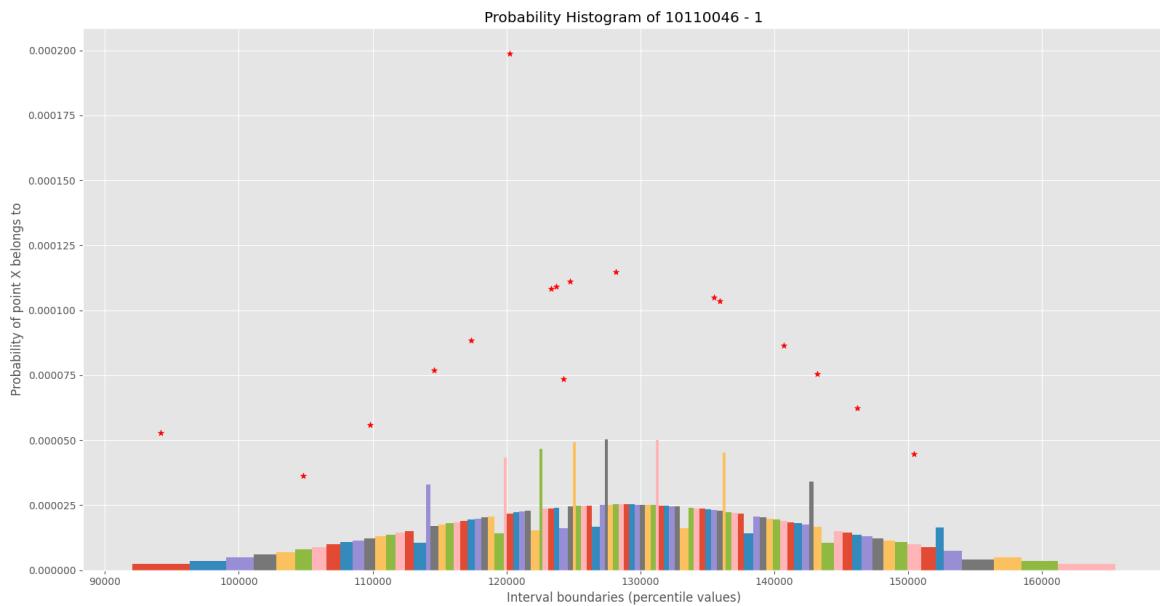
# Függelék



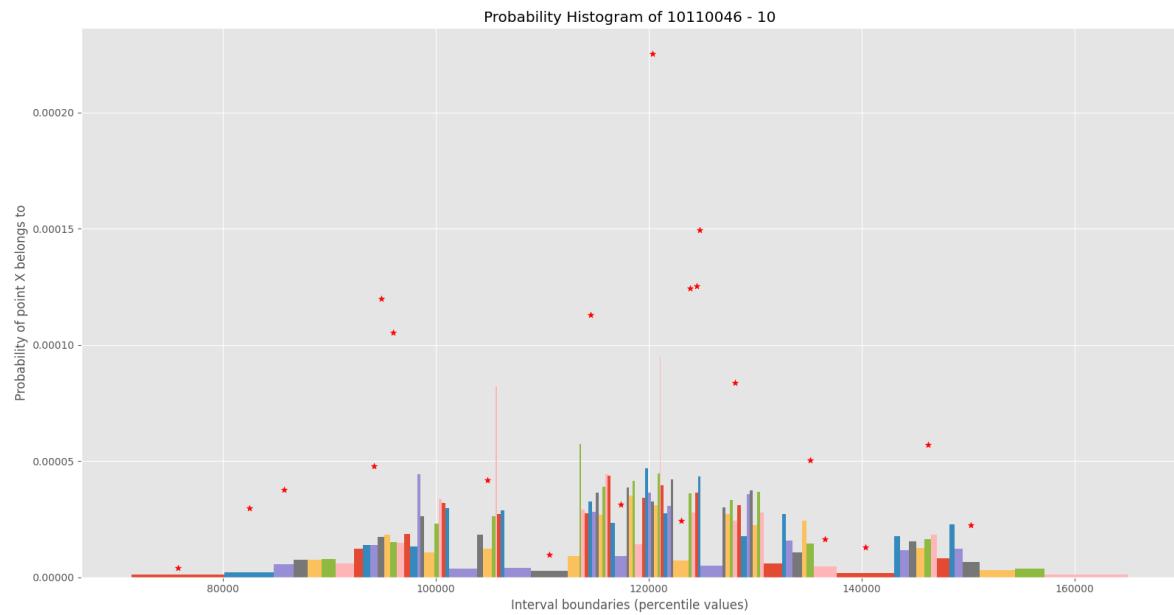
**F.0.1. ábra.** Valószínűségi hisztogram 10110027 - 1 esetén



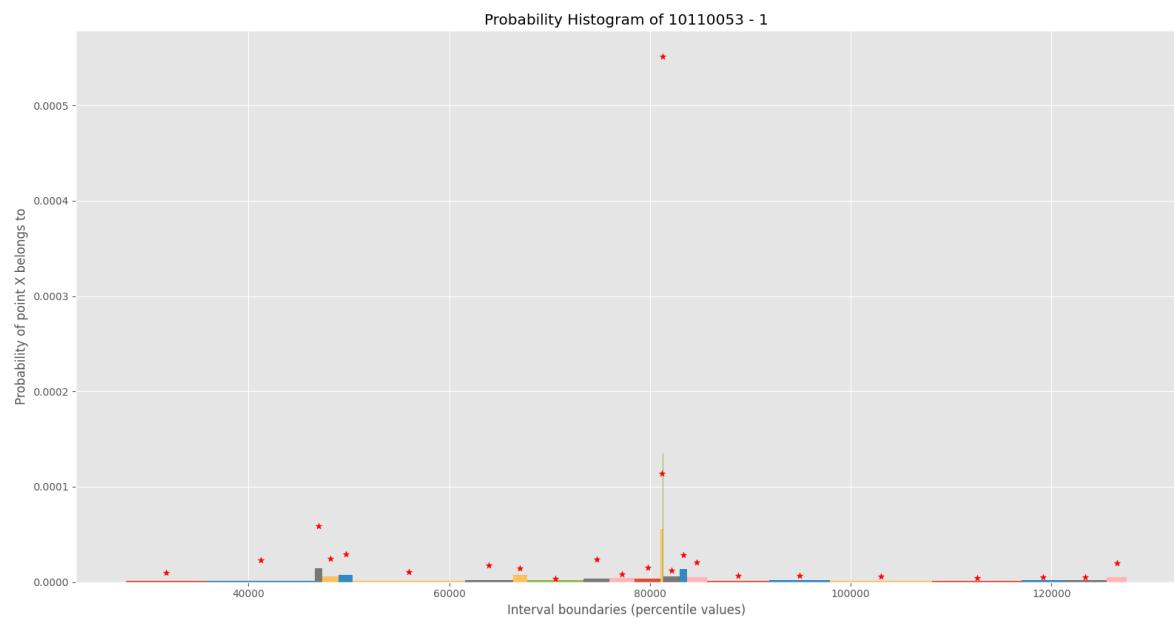
**F.0.2. ábra.** Valószínűségi hisztogram 10110027 - 10 esetén



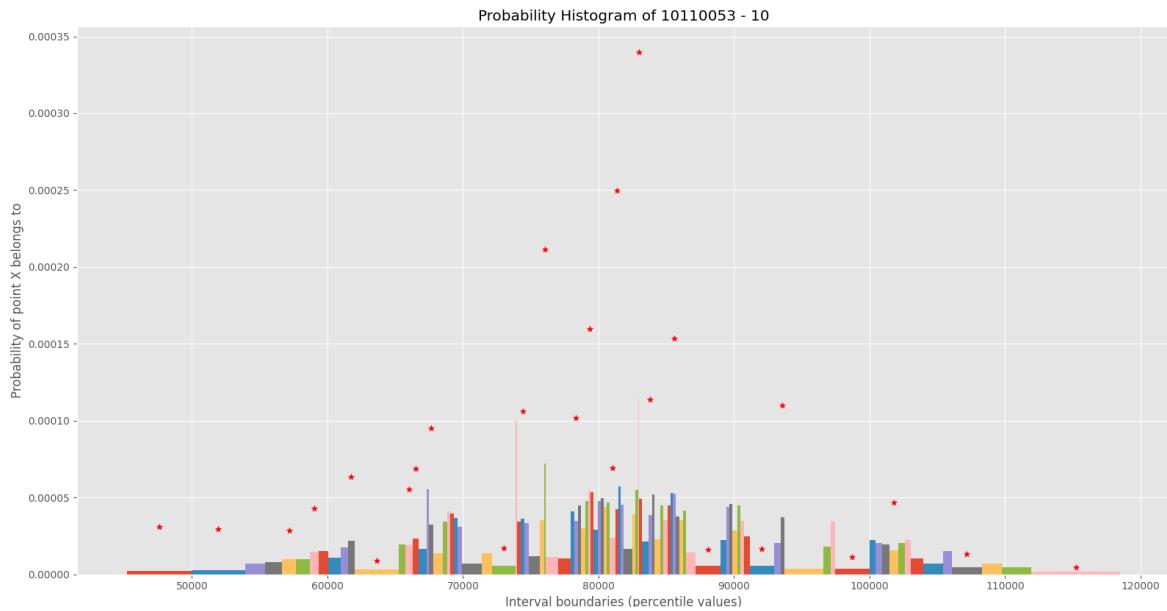
**F.0.3. ábra.** Valószínűségi hisztogram 10110046 - 1 esetén



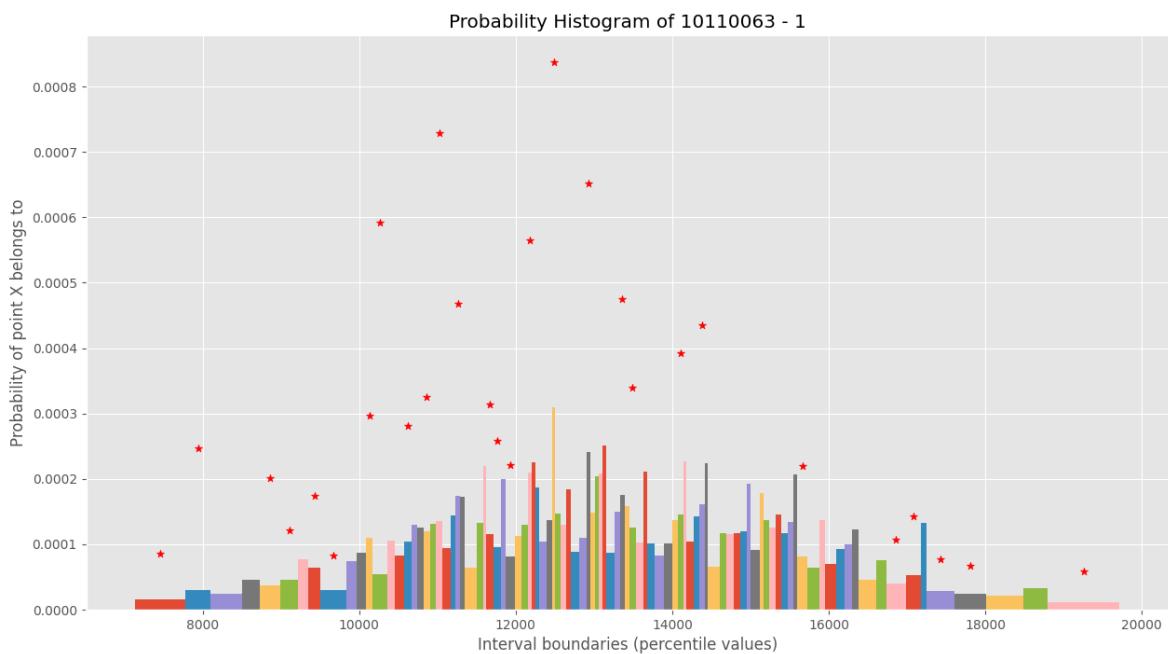
**F.0.4. ábra.** Valószínűségi hisztogram 10110046 - 10 esetén



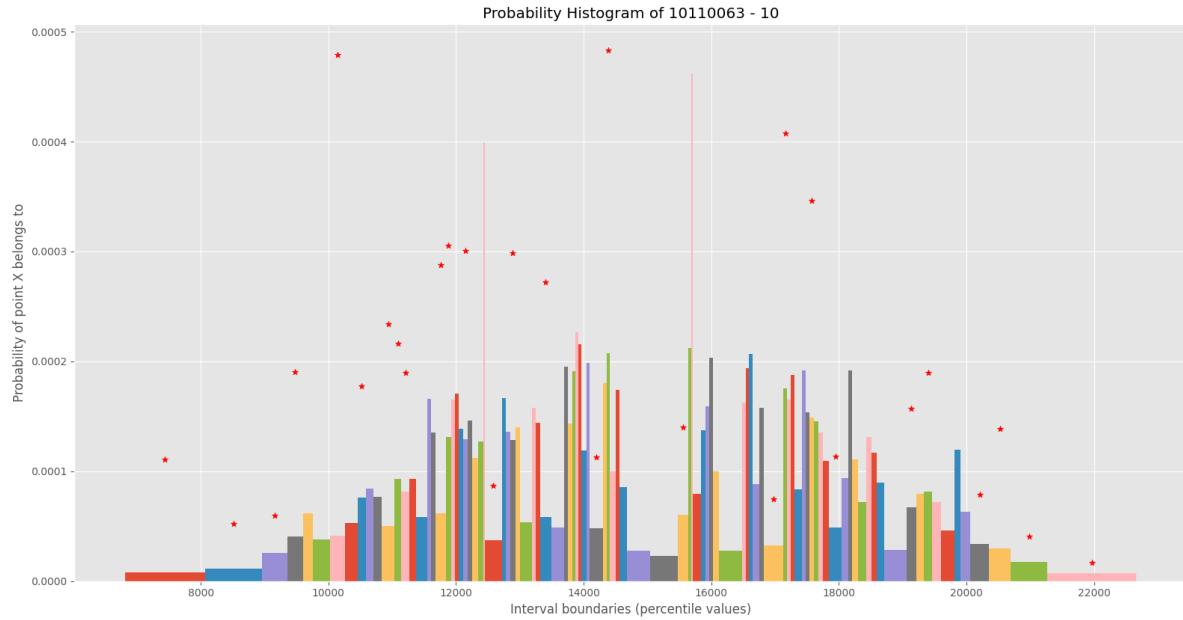
**F.0.5. ábra.** Valószínűségi hisztogram 10110053 - 1 esetén



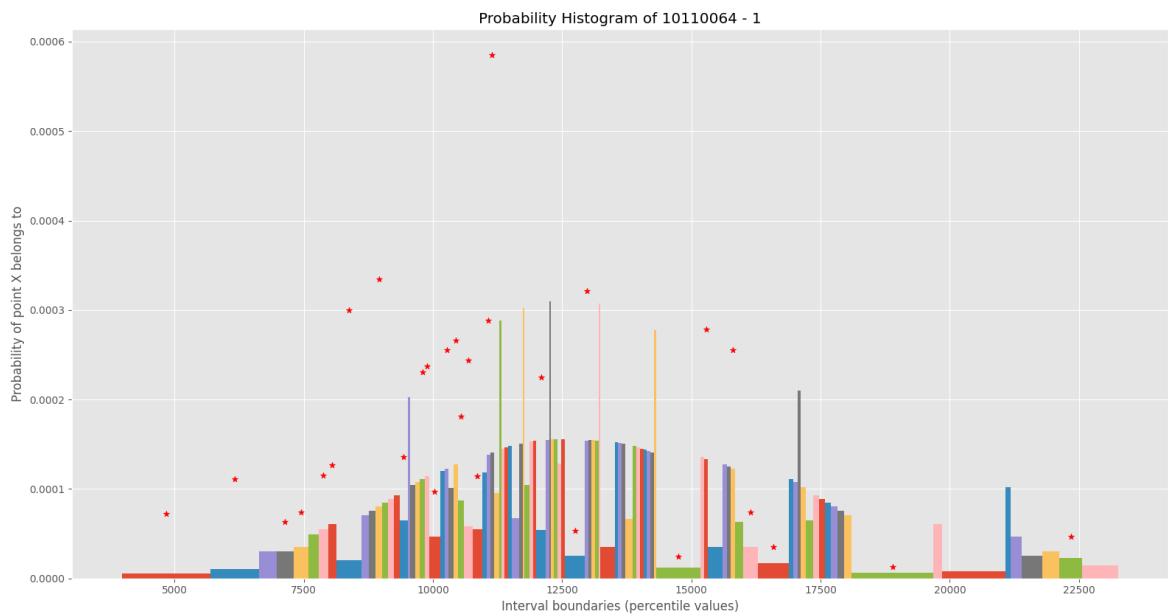
**F.0.6. ábra.** Valószínűségi hisztogram 10110053 - 10 esetén



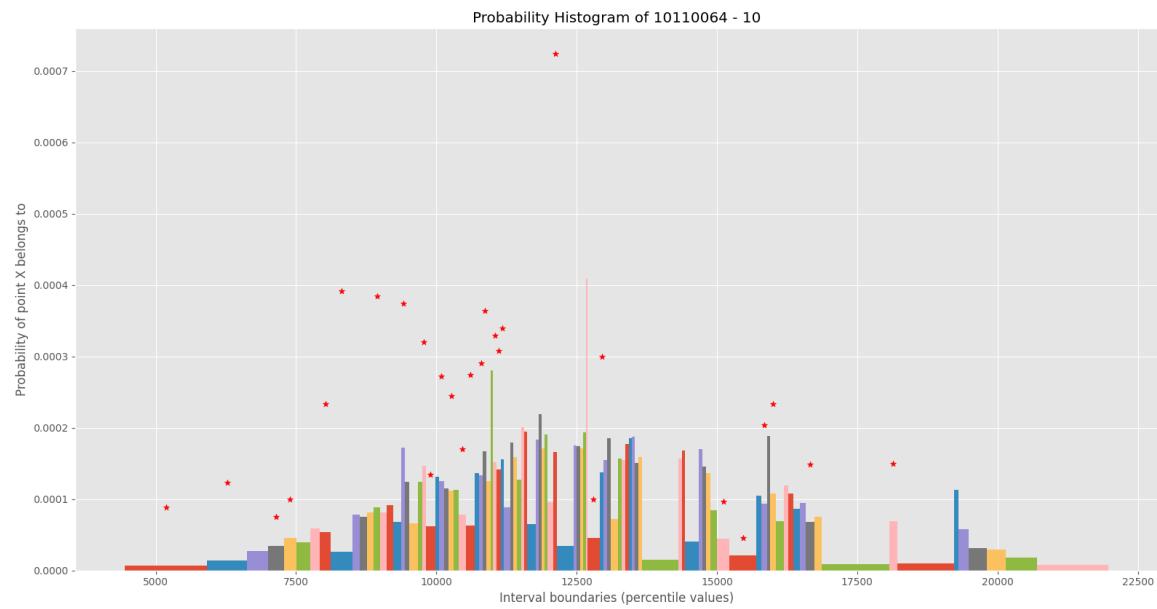
**F.0.7. ábra.** Valószínűségi hisztogram 10110063 - 1 esetén



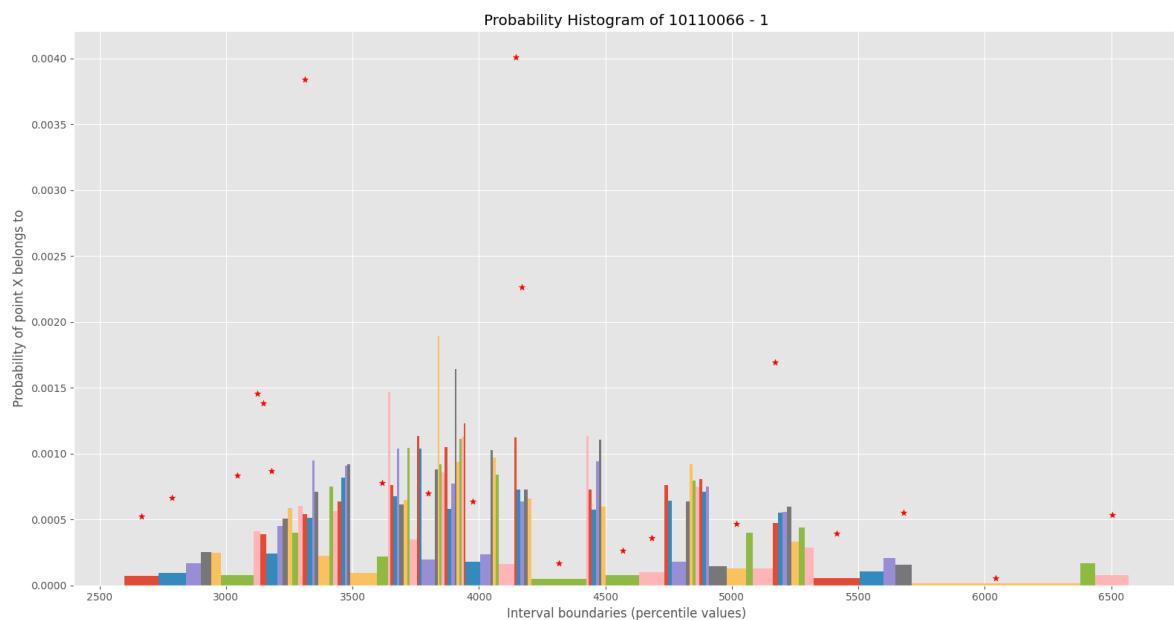
F.0.8. ábra. Valószínűségi hisztogram 10110063 - 10 esetén



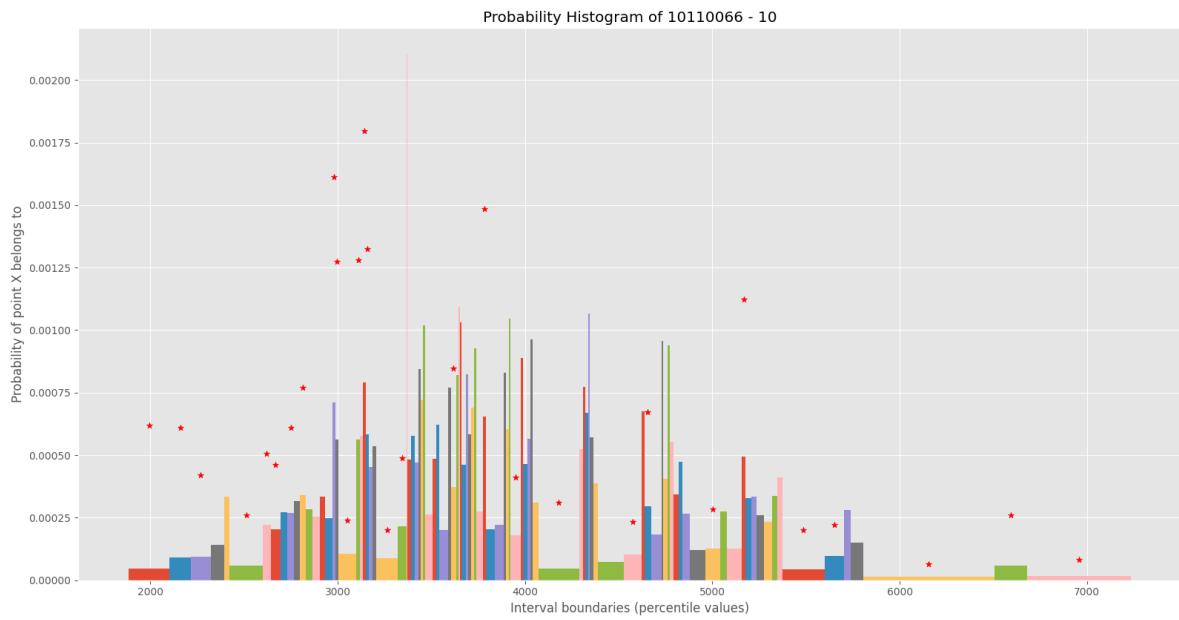
F.0.9. ábra. Valószínűségi hisztogram 10110064 - 1 esetén



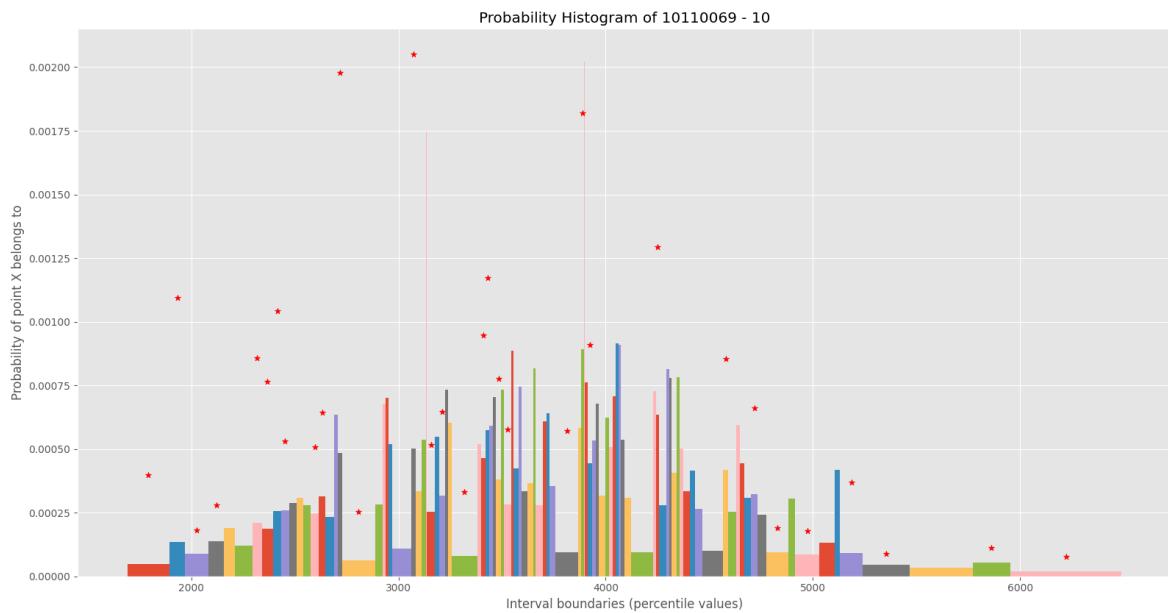
**F.0.10. ábra.** Valószínűségi hisztogram 10110064 - 10 esetén



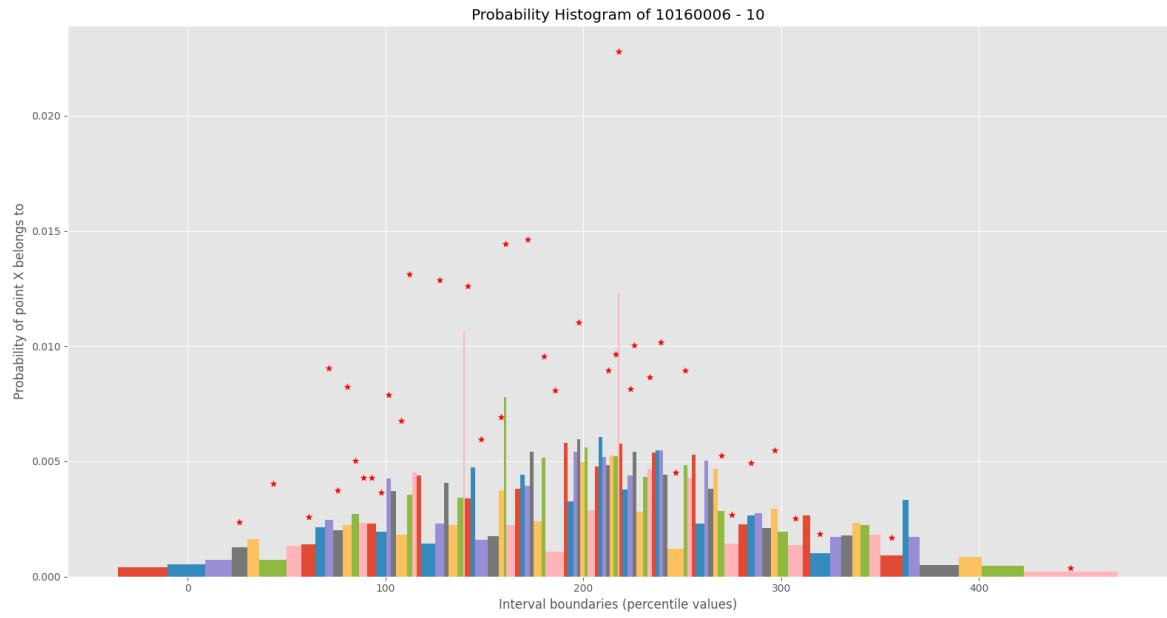
**F.0.11. ábra.** Valószínűségi hisztogram 10110066 - 1 esetén



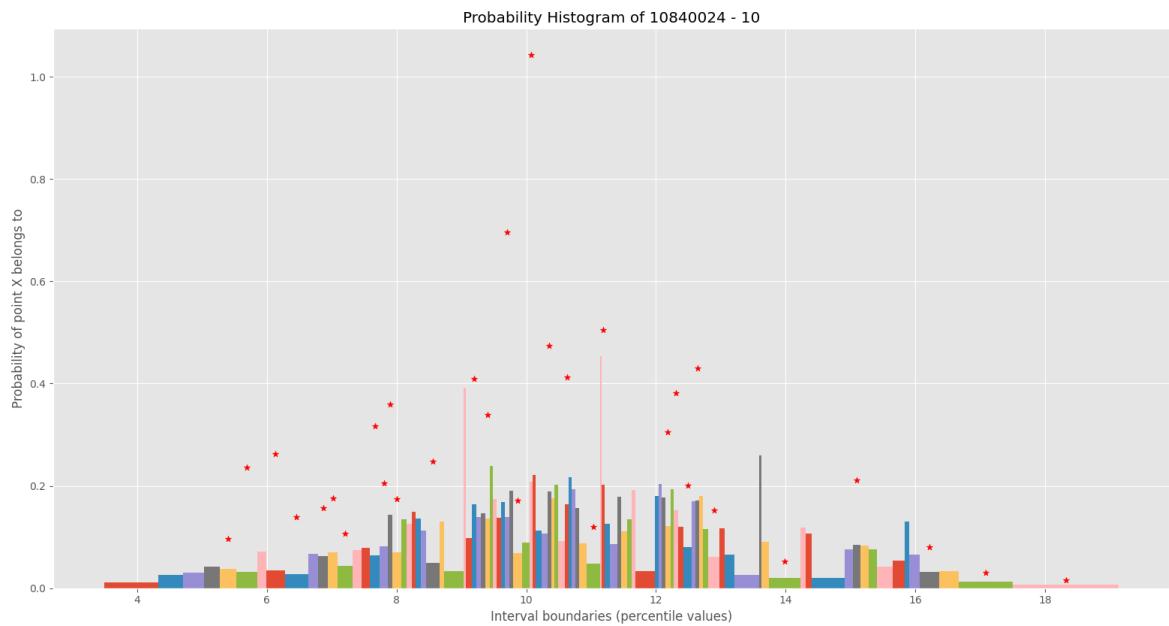
**F.0.12. ábra.** Valószínűségi hisztogram 10110066 - 10 esetén



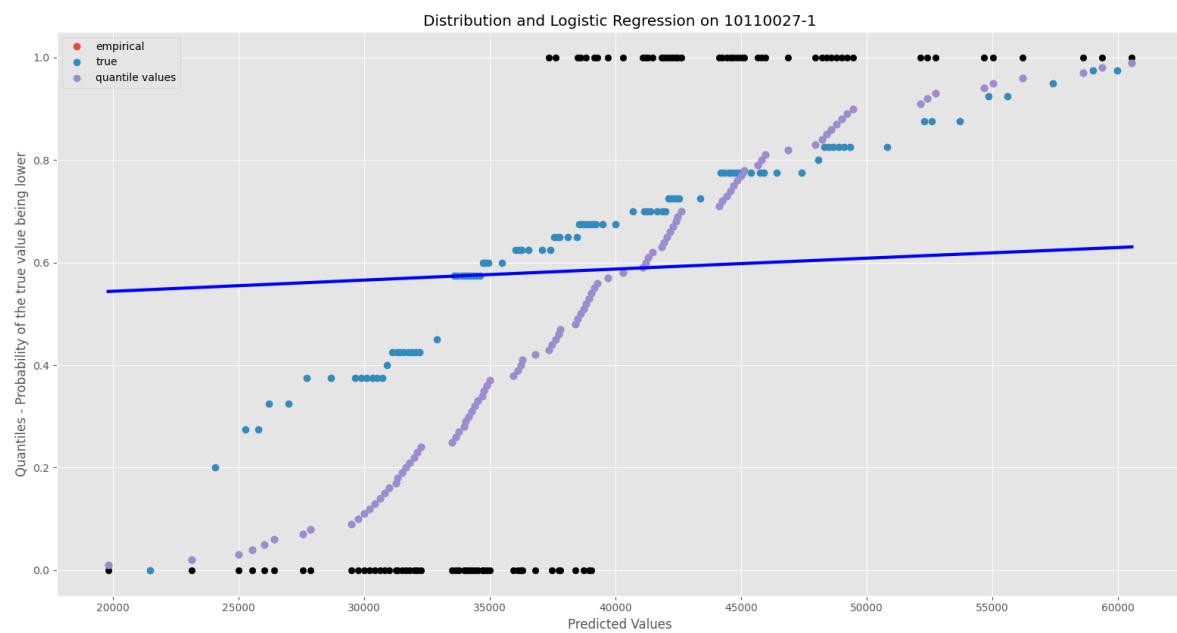
**F.0.13. ábra.** Valószínűségi hisztogram 10110066 - 10 esetén



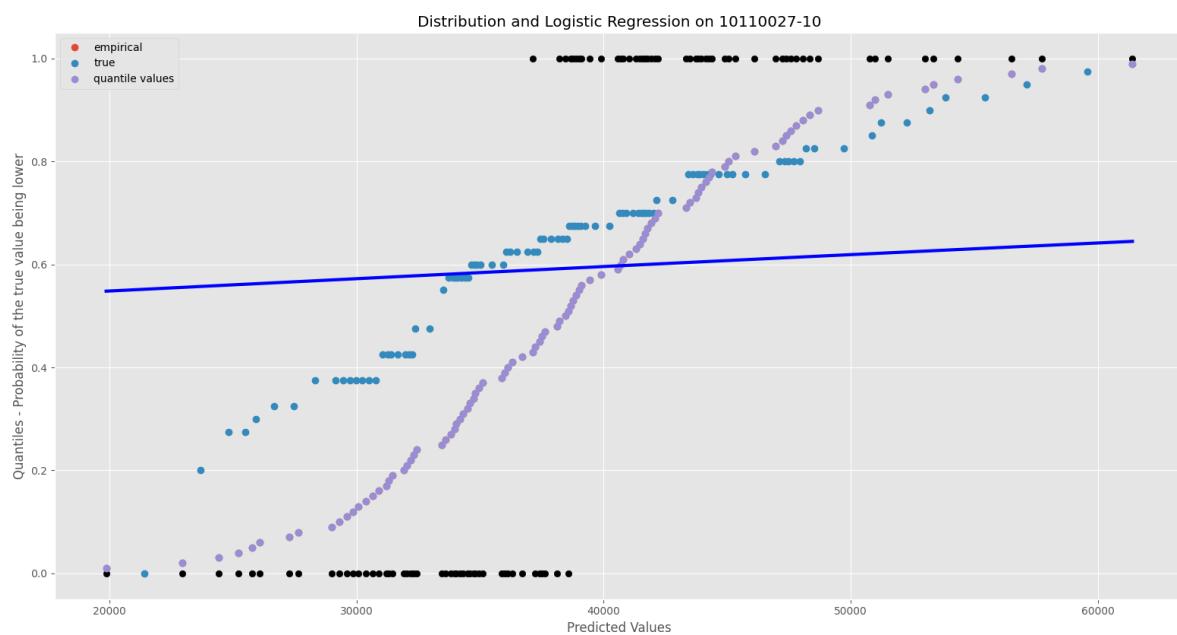
**F.0.14. ábra.** Valószínűségi hisztogram 10160006 - 10 esetén



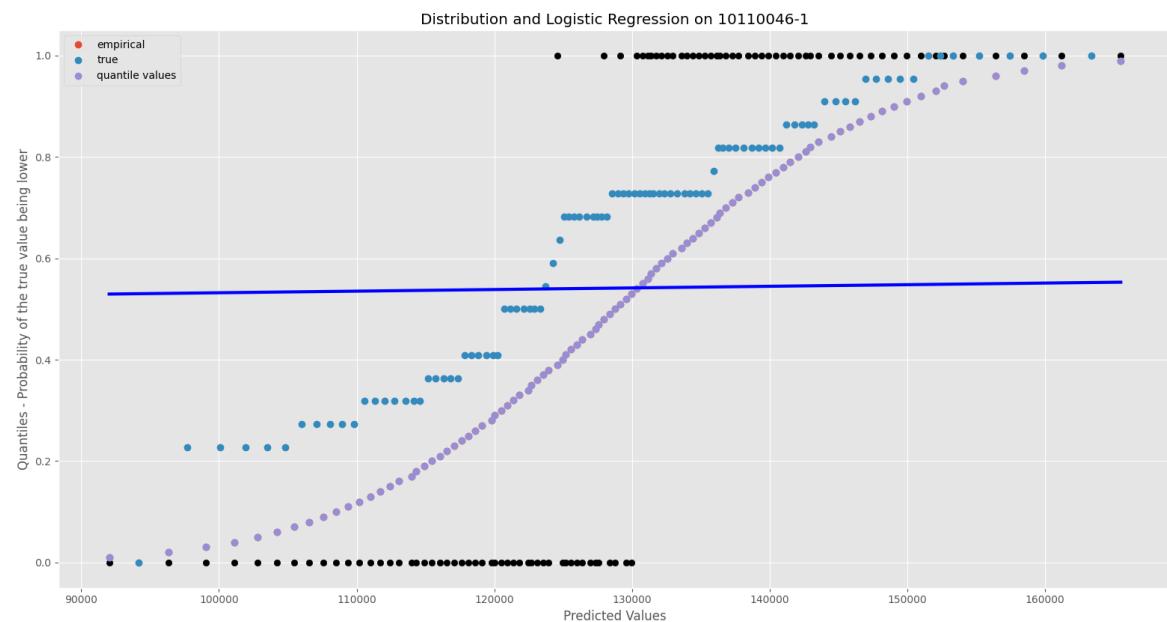
**F.0.15. ábra.** Valószínűségi hisztogram 10840024 - 10 esetén



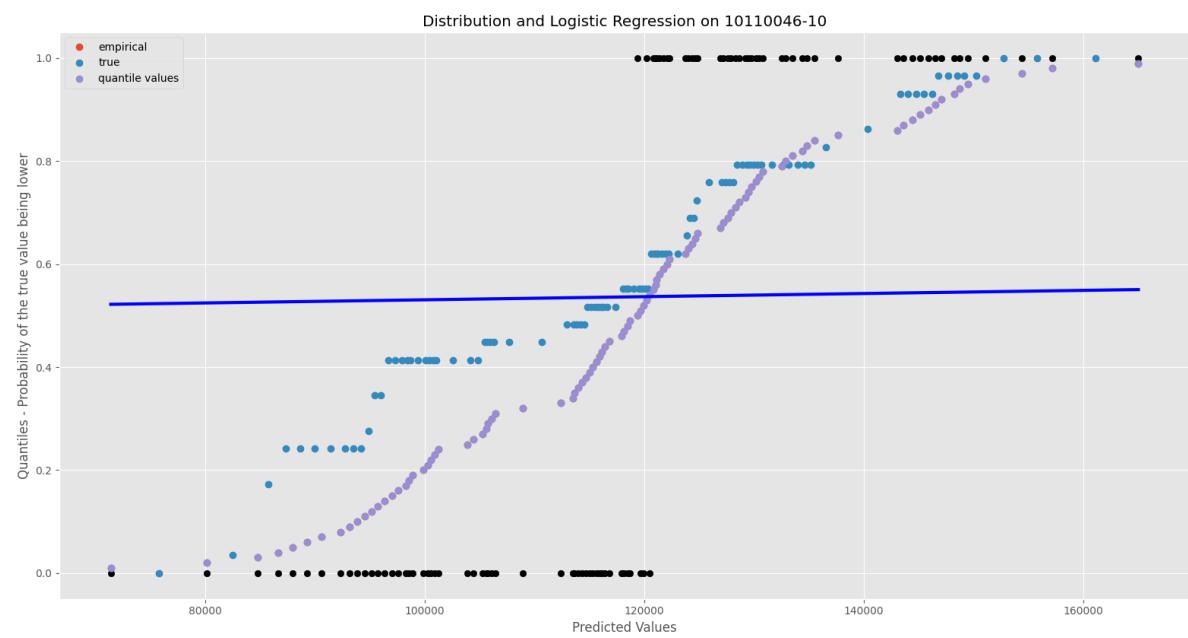
**F.0.16. ábra.** Eloszlás és Logisztikus görbe 10110027-1 esetén



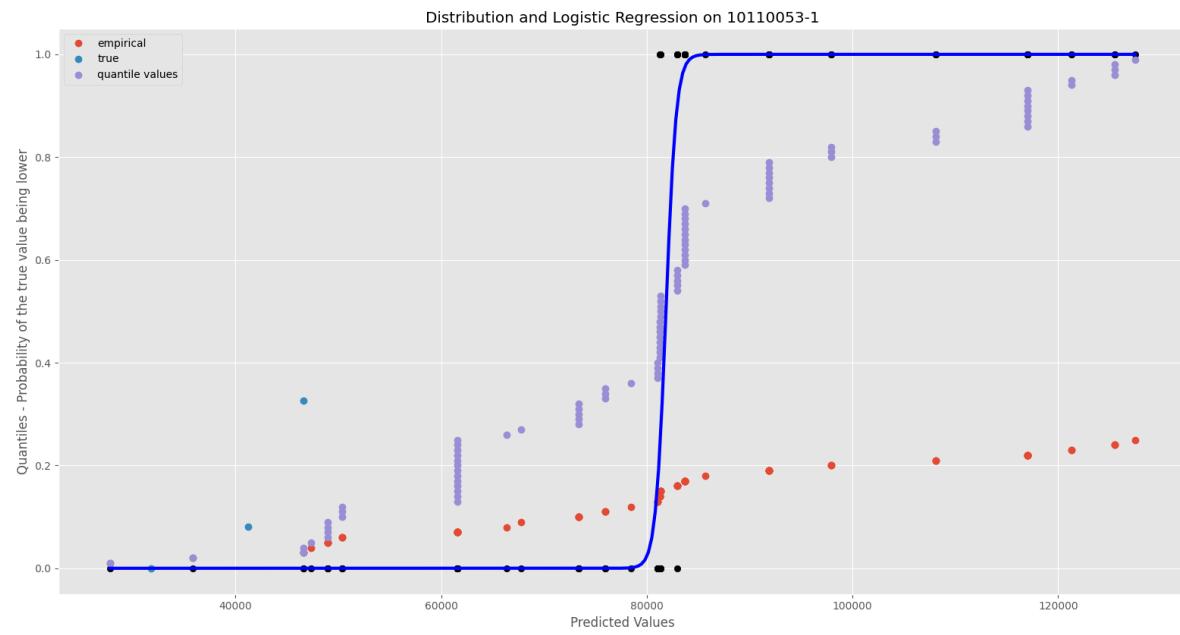
**F.0.17. ábra.** Eloszlás és Logisztikus görbe 10110027-10 esetén



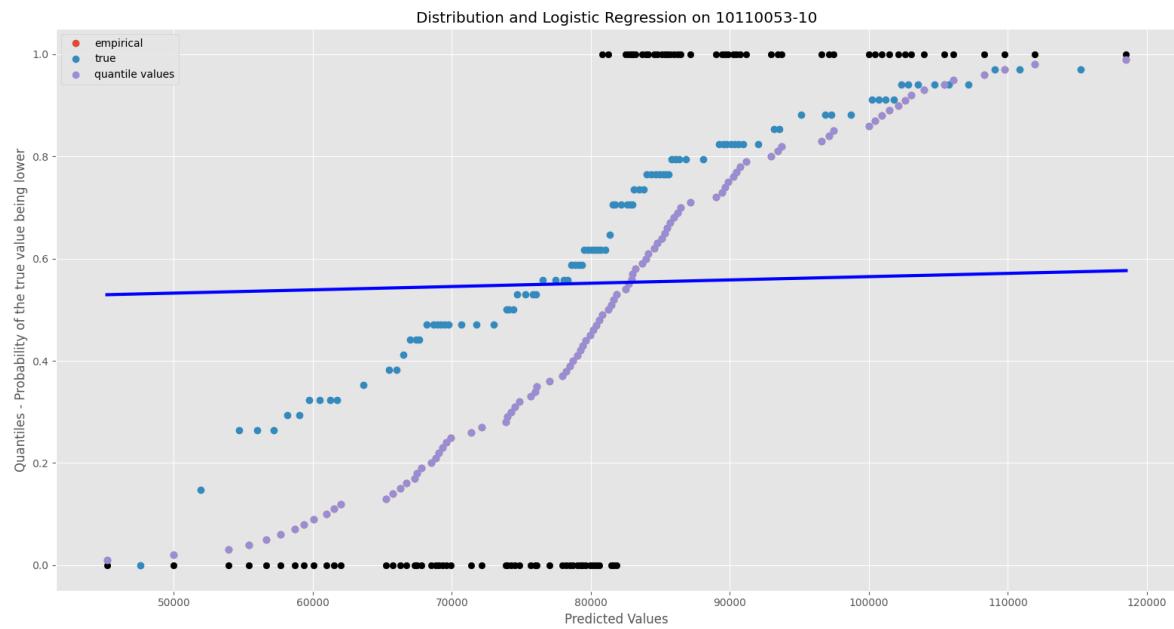
**F.0.18. ábra.** Eloszlás és Logisztikus görbe 10110046-1 esetén



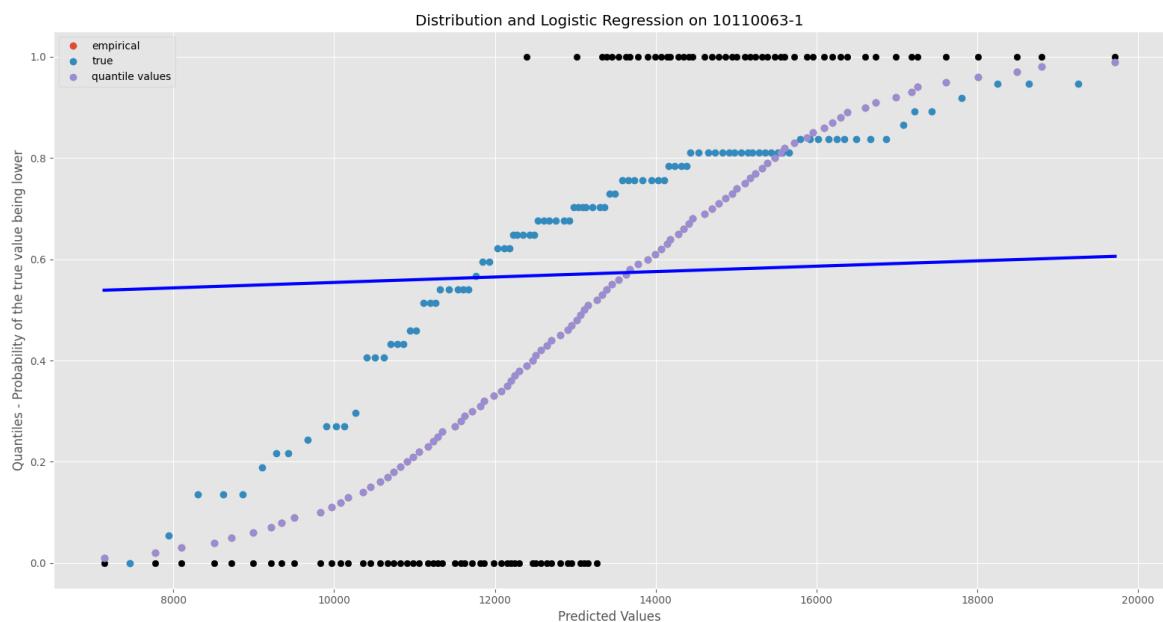
**F.0.19. ábra.** Eloszlás és Logisztikus görbe 10110046-10 esetén



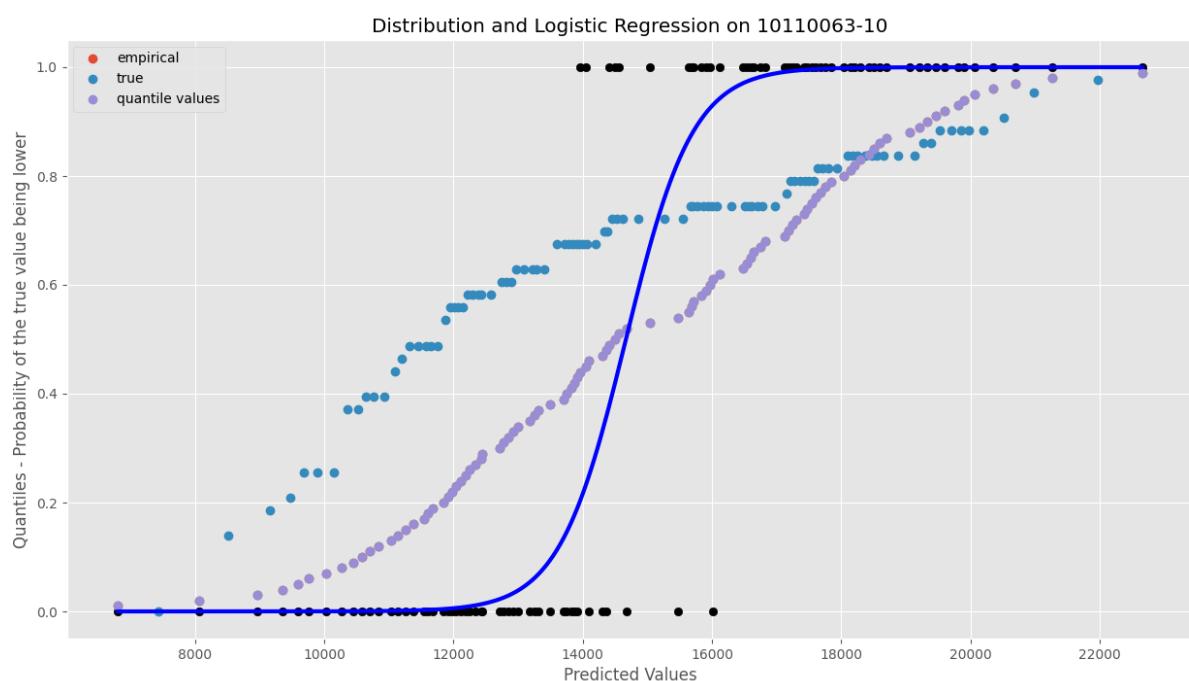
**F.0.20. ábra.** Eloszlás és Logisztikus görbe 10110053-1 esetén



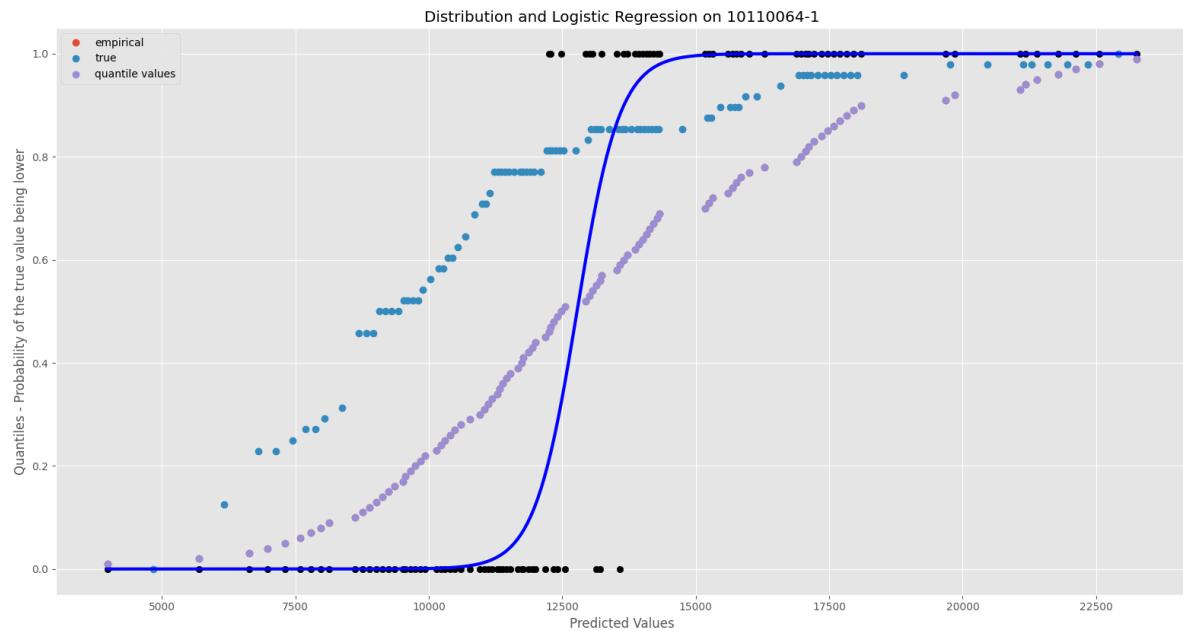
**F.0.21. ábra.** Eloszlás és Logisztikus görbe 10110053-10 esetén



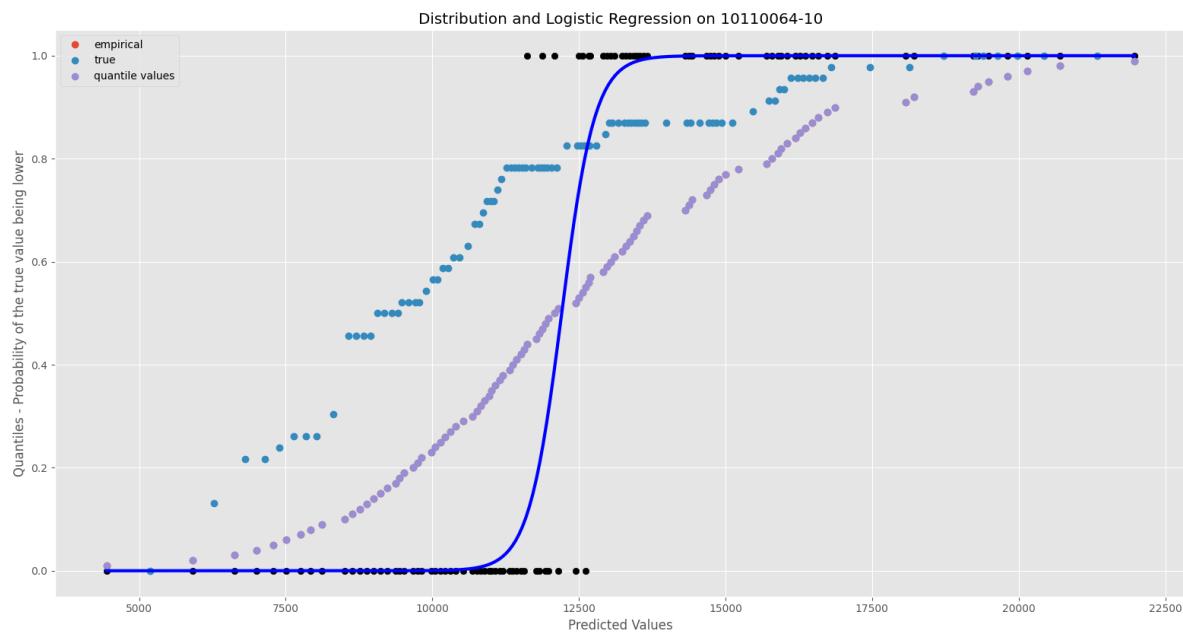
**F.0.22. ábra.** Eloszlás és Logisztikus görbe 10110063-1 esetén



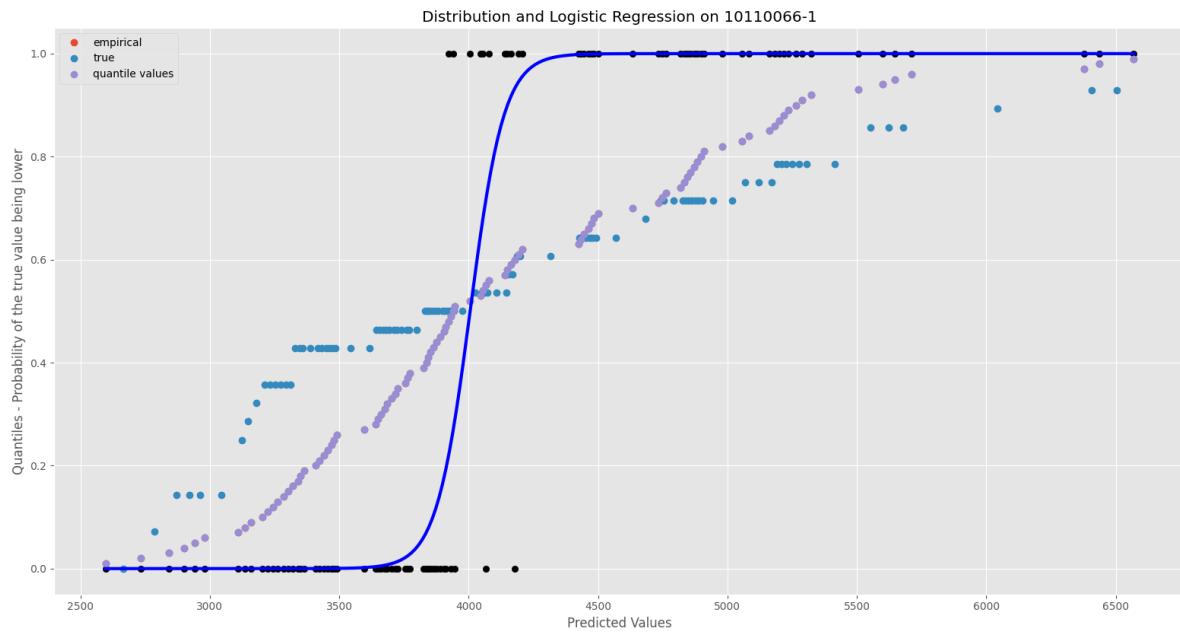
**F.0.23. ábra.** Eloszlás és Logisztikus görbe 10110063-10 esetén



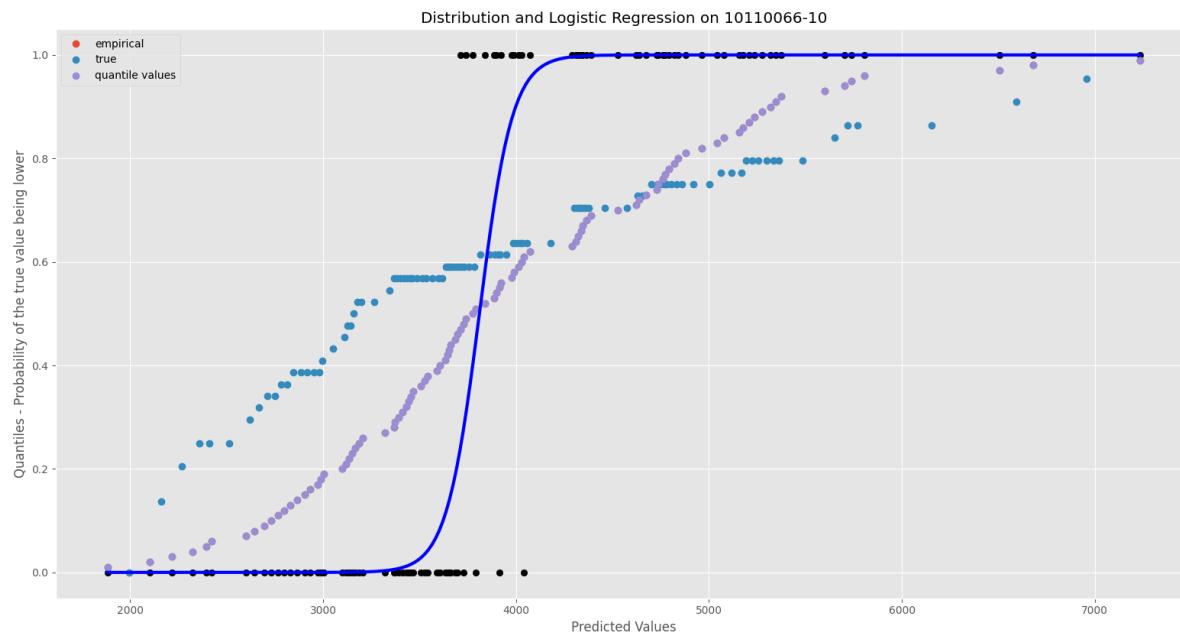
**F.0.24. ábra.** Eloszlás és Logisztikus görbe 10110064-1 esetén



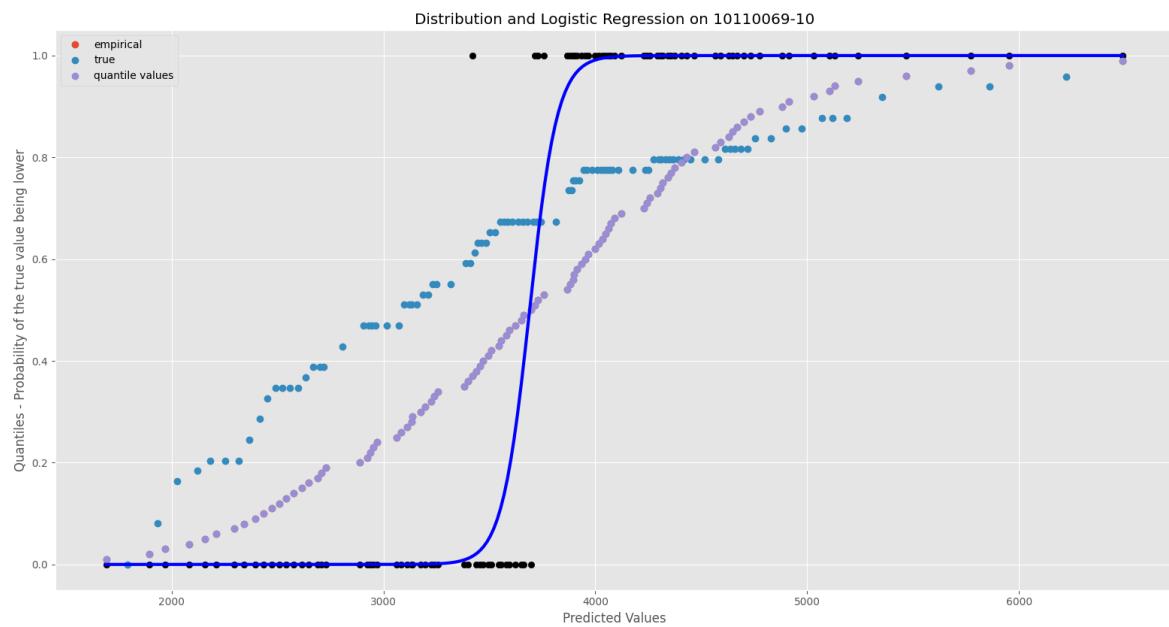
**F.0.25. ábra.** Eloszlás és Logisztikus görbe 10110064-10 esetén



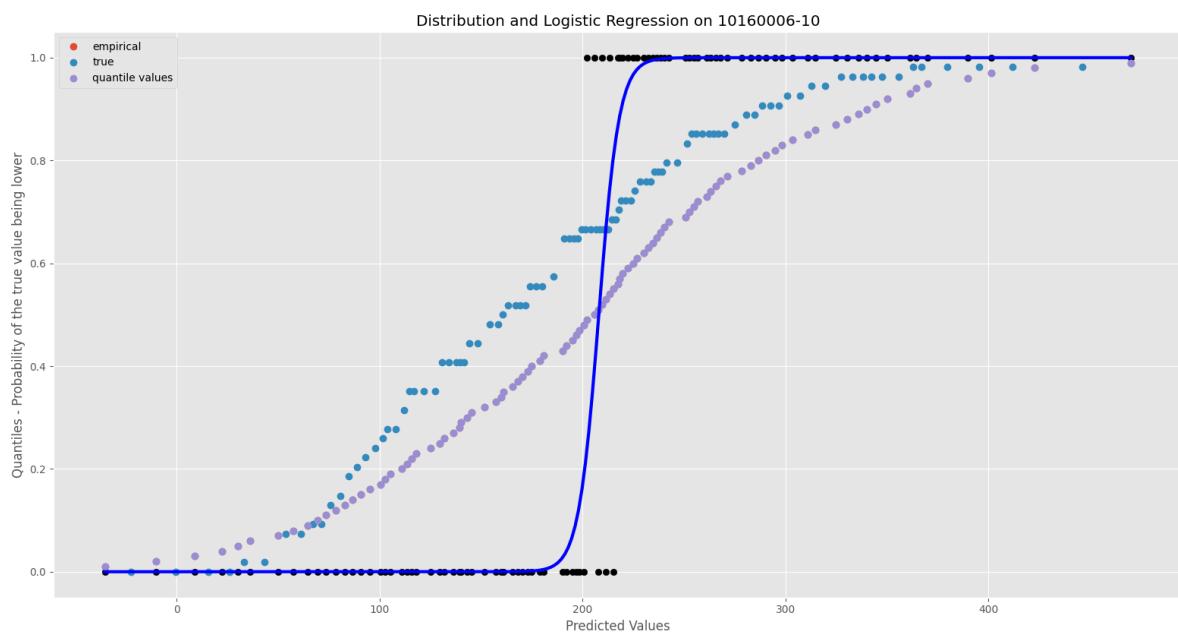
**F.0.26. ábra.** Eloszlás és Logisztikus görbe 10110066-1 esetén



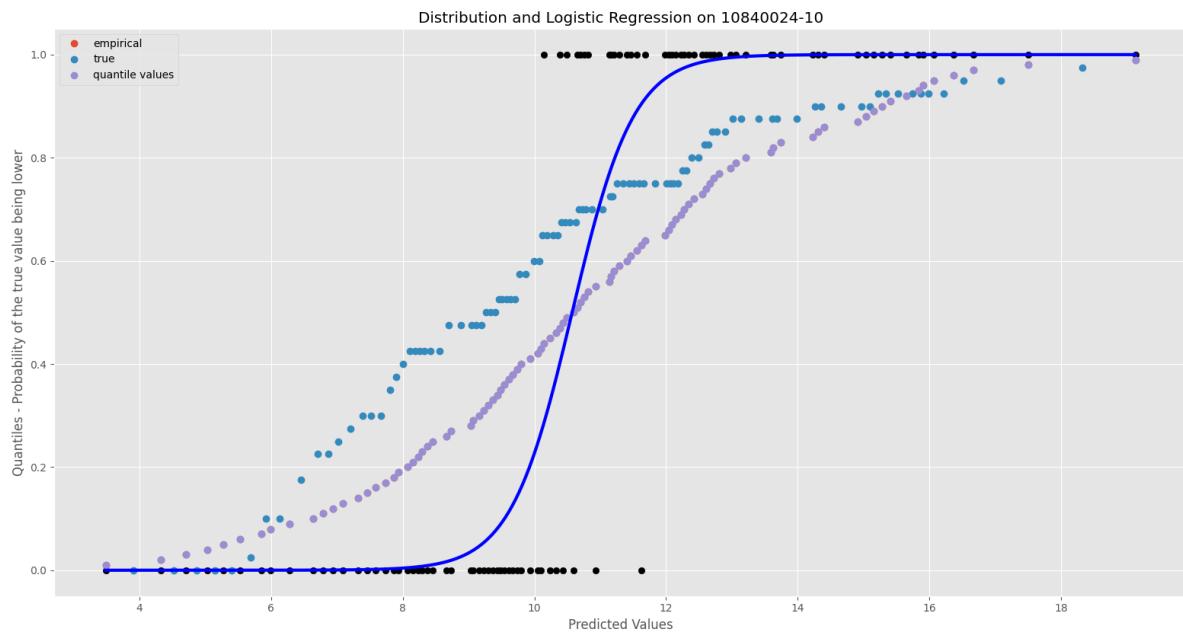
**F.0.27. ábra.** Eloszlás és Logisztikus görbe 10110066-10 esetén



**F.0.28. ábra.** Eloszlás és Logisztikus görbe 10110069-10 esetén



**F.0.29. ábra.** Eloszlás és Logisztikus görbe 10160006-10 esetén



**F.0.30. ábra.** Eloszlás és Logisztikus görbe 10840024-10 esetén