

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



**SAPIENTIA  
ERDÉLYI MAGYAR  
TUDOMÁNYEGYETEM**

Részvény árfolyamok előrejelzése és vizualizálása

**DIPLOMADOLGOZAT**

Témavezető:  
Dr. Horobet Emil,  
Egyetemi docens

Végzős hallgató:  
Jakab Etele

**2023**

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



**UNIVERSITATEA  
SAPIENTIA**

Predictia si vizualizarea preturilor la actiuni

**LUCRARE DE DIPLOMĂ**

Coordonator științific:  
Dr. Horobet Emil,  
Conferențiar universitar

Absolvent:  
Jakab Etele

**2023**

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



**SAPIENTIA**  
HUNGARIAN UNIVERSITY  
OF TRANSYLVANIA

Stock price prediction and visualization

**BACHELOR THESIS**

Scientific advisor:  
Dr. Horobet Emil,  
Associate Profesor

Student:  
Jakab Etele

**2023**

## LUCRARE DE DIPLOMĂ

Coordonator științific: **Candidat: Jakab Etele**  
**dr. Horobet Emil** Anul absolvirii: 2023

**a) Tema lucrării de licență:** Predicția și vizualizarea prețurilor la acțiuni

**b) Problemele principale tratate:**

Lucrarea este structurată în două capitole principale. Primul capitol se concentrează pe o aplicație desktop bazată pe Python și Tkinter, prezintându-se structura, principiul de funcționare și funcționalitățile deja implementate. În al doilea capitol, sunt prezentate două sisteme de prognozare dezvoltate în mod independent, precum și o a treia abordare rezultată din combinarea lor. Primul sistem este bazat pe o rețea neuronală de învățare profundă, în timp ce cel de-al doilea funcționează în abordarea analizei de știri utilizând algoritmi de clasificare. Obiectivul principal este ca programul să poată lucra cu datele aproape o sută de companii publice, să le actualizeze automat, să le proceseze cu mai multe metode analitice și să prezinte rezultatele utilizatorului cu ajutorul instrumentelor de vizualizare. Din punct de vedere al planificării programului, trebuie să fie urmat principiul modern de a putea fi extins în orice moment, dar și de a permite modificări ulterioare cu ușurință.

**c) Desene obligatorii:-**

- Schema bloc a aplicației
- Diagrame de proiectare pentru aplicația software realizată.

**d) Softuri obligatorii:-**

O aplicație desktop bazată pe Python și Tkinter pentru prognozarea mișcărilor bursiere.

**e) Bibliografia recomandată:**

- [1] ABARBANEL, Jeffrey S. BUSHEE, Brian J. Fundamental analysis, future earnings, and stock prices. Journal of accounting research, 1997, 35.1: 1-24.
- [2] BOURKE, Paul. Cross correlation. Cross Correlation”, Auto Correlation—2D Pattern Identification, 1996.
- [3] COHEN, Israel, et al. Pearson correlation coefficient. Noise reduction in speech processing, 2009, 1-4.

**f) Termene obligatorii de consultații:**

Minimum o dată din două în două săptămâni.

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

Primit tema la data de: 01.10.2022

Termen de predare: 30.06.2023

Semnătura Director Departament

Semnătura responsabilului  
programului de studiu

Semnătura coordonatorului

Semnătura candidatului

## Declarație

Subsemnatul/a ZAKAB ETELE, absolvant(ă) al/a specializării INFORMATICĂ, promoția 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, TÂRGU MUREŞ  
Data: 06.06.2023

Absolvant

Semnătura.....  
Zakab Etele

# Kivonat

A közigazdaság és a pénzügyi elemzések egyik legvizsgáltabb ágazata a részvénykereskedelem, ahol a tevékenység folyamán a szereplők részvényeket vásárolnak és adnak el a tőzsdén vagy más pénzügyi piacokon. Mondhatni ez a kapitalista országok legdinamikusabb befektetői területe, amely számos előnyvel rendelkezik, de több kritikus veszélyt is magában hordoz.

Az ágazat legnagyobb előnye az, hogy a befektetők gyorsan, akár nagyméretű nyereségre lehetnek szert a részvények emelkedése esetén vagy adott időszakonként fizetett osztalékok után, azonban, az árak volatilitása miatt számolni kell a veszteség lehetőségevel az árfolyamok hirtelen zuhanása következtében. A befektetőknek szükségük van megfelelő tudásra és tapasztalatra ahhoz, hogy sikeresen kereskedjenek és, hogy minimalizálják a kockázatokat.

Az árfolyamok változása számos tényezőtől függ, mint például a gazdasági környezet, a politikai helyzet, a piaci trendek és a vállalati eredmények. Ezeknek a tényezőknek a vizsgálata és elemzése nagyon bonyolult folyamat, tekintve, hogy az árfolyamváltozásokat nagyon összetett impulzusok váltják ki, amelyek mögött magas absztrakciós szintek állnak.

A dolgozat két nagy fejezetből áll. Az első fejezet témája egy Python, Tkinter alapú asztali alkalmazás amely eszközöként szolgál a különféle funkciók használatára, annak felépítése, működési elve valamint a már implementált funkcióinak bemutatása. A dolgozat második fejezetében bemutatásra kerül két különböző saját fejlesztésű és előrejelző rendszer, valamint az azok kombinálásából létrejött harmadik megközelítés. Ezen rendszerekből az első mélytanulású neurális háló alapú, a második osztályozó algoritmusok által a hírelemzés alapján működik.

Az asztali alkalmazás fejlesztése során az volt a célkitűzés, hogy a program legyen képes közel száz nyilvános cég adataival is dolgozni, azokat automatikusan frissíteni, többféle elemzői módszerrel feldolgozni és ezeket az eredményeket vizualizációs eszközök segítségével megmutatni a felhasználónak. Programtervezési szempontból azt a korszerű elvet követtem, hogy bármikor lehessen bővíteni, de akár utólagos változtatásokat is könnyen lehessen elvégezni az applikáción.

A kutatási rész célja olyan módszer kidolgozása, amely a piaci változások tekintetében nagy pontosságú előrejelzésekre képes a korábbi piaci adatok és hírfolyamok alapján. Ezeket a módszereket letesztelni, az eredményeket dokumentálni, összehasonlítani egymással és levonni a lényeges következtetéseket.

A kutatás társadalmi célja az, hogy a létrehozott módszerek segítségével biztonságabban lehessen a tőzsdén befektetni, megvédve az emberek megtakarításait, kiválthatóbban egy olyan időszakban, amikor a gazdasági kilátások meglehetősen viharosak. A 2022-es átlagos éves infláció az Európai Unióban 10.4%, Magyarországon 24.5%, Romániában

14.1% míg az Euró övezetben is 9.2%-os volt, ezzel szemben az átlagos éves betéti kamatozás Magyarországon 6.1%, Romániában 7.2%-ot értek el.

A kutatás során négy különböző cég adatait vizsgáltam, amelyek a piac négy különböző szegmensét képviselik: The Walt Disney Company, Ford Motor Company, UBS Group AG valamint Exxon Mobil Corporation.

**Kulcsszavak:** asztali alkalmazás, részvényelemzés, vizualizálás, neurális háló, előrejelzés.

# Rezumat

Una dintre cele mai studiate sectoare ale economiei și analizei financiare este tranzacționarea cu acțiuni, unde participanții cumpără și vând acțiuni la bursă sau alte piețe financiare. Se poate spune că aceasta este zona de investiții cea mai dinamică în țările capitaliste, care are numeroase avantaje, dar și poartă în sine mai multe riscuri critice.

Avantajul principal al acestei industrii constă în faptul că investitorii pot obține rapid și potențial profituri mari în cazul creșterii prețului acțiunilor sau în urma dividendelor plătite în anumite perioade. Cu toate acestea, din cauza volatilității prețurilor, trebuie să se țină cont de posibilitatea pierderilor cauzate de scăderea bruscă a cursului de schimb. Investitorii trebuie să aibă cunoștințe și experiență adecvate pentru a tranzacționa cu succes și pentru a minimiza riscurile.

Variatia cursurilor de schimb depinde de numeroși factori, cum ar fi mediul economic, situația politică, tendințele pieței și rezultatele companiilor. Analiza și evaluarea acestor factori reprezintă un proces extrem de complex, având în vedere că schimbările de curs sunt declanșate de impulsuri foarte complexe, în spatele cărora se află niveluri ridicate de abstractizare.

Lucrarea este structurată în două capitole principale. Primul capitol se concentreză pe o aplicație desktop bazată pe Python și Tkinter, prezentându-se structura, principiul de funcționare și funcționalitățile deja implementate. În al doilea capitol, sunt prezentate două sisteme de prognozare dezvoltate în mod independent, precum și o a treia abordare rezultată din combinarea lor. Primul sistem este bazat pe o rețea neuronală de învățare profundă, în timp ce cel de-al doilea funcționează în abordarea analizei de știri utilizând algoritmi de clasificare.

În timpul dezvoltării aplicației desktop, obiectivul principal a fost ca programul să poată lucra cu datele aproape o sută de companii publice, să le actualizeze automat, să le proceseze cu mai multe metode analitice și să prezinte rezultatele utilizatorului cu ajutorul instrumentelor de vizualizare. Din punct de vedere al planificării programului, am urmat principiul modern de a putea fi extins în orice moment, dar și de a permite modificări ulterioare cu ușurință.

Scopul părții de cercetare este dezvoltarea unor metode capabile să facă prognoze de înaltă precizie în ceea ce privește schimbările de pe piață, pe baza datelor de piață și a fluxurilor de știri anterioare. Aceste metode trebuie testate, rezultatele documentate, comparate între ele și trase concluziile esențiale.

Scopul social al cercetării este de a permite investiții mai sigure pe piață de valori, protejând economiile oamenilor, în special într-o perioadă în care perspectivele economice sunt destul de tumultuoase. Inflația medie anuală în Uniunea Europeană în 2022 a fost de 10,4%, în Ungaria de 24,5%, în România de 14,1%, în timp ce în zona euro a fost de

9,2%, în contrast cu dobânzile anuale medii la depozite care au atins 6,1% în Ungaria și 7,2% în România.

În cadrul cercetării, am examinat datele a patru companii diferite care reprezintă patru segmente diferite de piață: The Walt Disney Company, Ford Motor Company, UBS Group AG și Exxon Mobil Corporation.

**Cuvinte de cheie:** aplicație desktop, analiză acțiuni, vizualizare, rețea neuronală, prognoză.

# Abstract

One of the most studied sectors of economics and financial analysis is stock trading, where participants buy and sell shares on the stock exchange or other financial markets. It can be said that this is the most dynamic area for investors in capitalist countries, which has numerous advantages but also carries several critical risks.

The biggest advantage of this industry is that investors can quickly gain substantial profits from the rise of stock prices or periodic dividend payments. However, due to price volatility, one must also consider the possibility of losses resulting from sudden drops in stock prices. Investors need proper knowledge and experience to trade successfully and minimize risks.

The fluctuation of stock prices depends on various factors such as the economic environment, political situation, market trends, and corporate performance. Examining and analyzing these factors is a highly complex process, given that stock price changes are triggered by intricate impulses rooted in high levels of abstraction.

The thesis consists of two main chapters. The first chapter focuses on a desktop application built with Python and Tkinter, presenting its structure, operational principles, and already implemented functionalities. The second chapter introduces two independently developed and separate prediction systems, as well as a third approach resulting from their combination. The first system is based on deep learning neural networks, while the second operates using classification algorithms in the approach of news analysis.

The aim during the development of the desktop application was to enable it to work with data from nearly a hundred publicly traded companies, automatically update them, process them using multiple analytical methods, and present the results to the user using visualization tools. From a program design perspective, I followed the modern principle of allowing for easy expansion and post-development modifications of the application.

The research section aimed to develop a method capable of providing highly accurate predictions regarding market changes based on past market data and news feeds. These methods were tested, results were documented, compared with each other, and essential conclusions were drawn.

The social goal of the research is to facilitate safer stock market investments and protect people's savings, particularly during a period when economic prospects are rather turbulent. In 2022, the average annual inflation in the European Union was 10.4%, in Hungary 24.5%, in Romania 14.1%, while in the Eurozone it reached 9.2%. In contrast, the average annual deposit interest rates in Hungary were 6.1%, and in Romania, they reached 7.2%.

During the research, data from four different companies representing four distinct market segments were examined: The Walt Disney Company, Ford Motor Company, UBS Group AG, and Exxon Mobil Corporation.

**Keywords:** desktop application, stock analysis, visualization, neural network, forecasting.

# Köszönetnyilvánítás

Ezúton szeretném kifejezni köszönetemet mindenazonknak, akik támogattak a dolgozat elkészítésében. Köszönetemet szeretném kifejezni témavezetőmnek, Dr. Horobet Emílnék, aki beszél erről a témáról és bátorított e kutatás elkészítésére és mindvégig segített tanácsaival. Köszönetet szeretnék mondani Dr. Dávid Lászlónak is, aki mesterséges intelligencia órán bevezetett a gépi tanulás rejtelmeibe, valamint azért a számos fontos tanácsért, melyek nélkül ez a tanulmány nem készült volna el.

Továbbá szeretnék köszönetet mondani Jakab Barna-Róbert és Jakab Irma-Tünde szüleimnek, akik motiváltak és támogattak az egész tudományos pályafutásom során. Ezt a dolgozatot nem lehetett volna elkészíteni tanáraim és családom segítsége nélkül. Mindannyiuknak nagyon hálás vagyok!

# Tartalomjegyzék

## Köszönetnyilvánítás

<b>1. FEJLESZTETT ALKALMAZÁS</b>	<b>11</b>
1.1. Bevezető . . . . .	11
1.2. A probléma bemutatása . . . . .	11
1.2.1. A dolgozat fontosabb lépései . . . . .	12
1.3. Saját megközelítés . . . . .	12
1.3.1. A fejlesztéshez használt programozási nyelv . . . . .	12
1.3.2. Matplotlib könyvtár . . . . .	13
1.3.3. Pandas könyvtár . . . . .	13
1.3.4. Nupmy könyvtár . . . . .	14
1.3.5. Tkinter könyvtár . . . . .	14
1.3.6. Yfinance könyvtár . . . . .	14
1.4. Az alkalmazás működési elve . . . . .	15
1.5. Szerkezeti elemek . . . . .	17
1.5.1. Adatkezelő rész . . . . .	17
1.5.2. Felhasználói felület oldalai . . . . .	18
1.5.3. Nyitó oldal . . . . .	20
1.5.4. Központi oldal . . . . .	21
1.5.5. More Details oldal . . . . .	22
1.5.6. Harang modell . . . . .	24
1.5.7. Mozgóátlag keresztek . . . . .	25
1.5.8. Polinomiális regresszió . . . . .	26
1.5.9. RBF közelítés . . . . .	27
1.5.10. Lokális valószínűségi tér . . . . .	27
1.6. Összegzés . . . . .	28
<b>2. KUTATÁS</b>	<b>30</b>
2.1. Bevezetés . . . . .	30
2.2. A probléma bemutatása . . . . .	30
2.2.1. A kutatás fontosabb lépései . . . . .	31
2.2.2. A kutatás során vizsgált kérdések . . . . .	31
2.2.3. Hipotézisek . . . . .	31
2.3. Saját megközelítés . . . . .	32
2.3.1. Az Információs források és a felhasznált adatok bemutatása . . . . .	32
2.4. A vizsgált cégek bemutatása . . . . .	32

2.4.1. The Walt Disney Company (továbbiakban DIS) . . . . .	32
2.4.2. Ford Motor Company. (továbbiakban F) . . . . .	33
2.4.3. UBS Group AG. (továbbiakban UBS) . . . . .	33
2.4.4. Exxon Mobil Corporation (továbbiakban XOM) . . . . .	34
2.5. A vizsgált cégek árfolyama közötti kapcsolat . . . . .	34
2.5.1. Keresztkorreláció . . . . .	34
2.5.2. A tanulmányozott árfolyamok korrelációja . . . . .	35
2.6. Az adatok belső szerkezetének vizsgálata . . . . .	36
2.6.1. Az ideális ablak meghatározása . . . . .	36
2.6.2. Reláció egymás utáni adatpontok között . . . . .	37
2.6.3. Osztályozás javítása . . . . .	38
2.7. Az adatsor tisztítása és transzformálása . . . . .	40
2.7.1. SMA50 . . . . .	40
2.7.2. Gramian-Angular transzformáció . . . . .	41
2.8. Eredmények a neurális hálóval . . . . .	42
2.8.1. Besorolási pontosság . . . . .	42
2.8.2. Szimulációs tesztelés . . . . .	44
2.9. A szövegelemzés módszere . . . . .	47
2.9.1. A felhasznált adatok . . . . .	47
2.9.2. A megfelelő algoritmus kiválasztása . . . . .	48
2.9.3. A LogisticRegression további vizsgálata . . . . .	49
2.9.4. Szimulációs hatékonyság . . . . .	52
2.10. A két módszer kombinálása és összehasonlítása . . . . .	54
2.10.1. A kombinált szimuláció . . . . .	54
2.11. A kutatás eredményeinek és tapasztalatainak az összegzése . . . . .	56
2.11.1. Melyik módszer a legjobb? . . . . .	56
2.11.2. Hipotézisek vizsgálata . . . . .	56
2.11.3. További következtetések . . . . .	56
2.11.4. Összegzés . . . . .	57
<b>Ábrák jegyzéke</b>	<b>58</b>
<b>Táblázatok jegyzéke</b>	<b>59</b>
<b>Irodalomjegyzék</b>	<b>61</b>

# 1. fejezet

## FEJLESZTETT ALKALMAZÁS

### 1.1. Bevezető

A részvény kereskedéssel kapcsolatos információk alapját maguk az árfolyamgörbék értékei adják. Az általam fejlesztett alkalmazás lehetővé teszi a részvényadatok vizualizálását, amely egy fontos eszköz a befektetők és elemzők számára. A részvények árának és teljesítményének nyomon követése alapvető fontosságú az informált döntéshozatalhoz a tőzsdén. Az alkalmazásban számos funkció és eszköz található, amelyek segítségével könnyedén feldolgozhatjuk és vizualizálhatjuk a részvényadatokat.

A program lehetővé teszi a részvények árának valós idejű megjelenítését, ami elérhetővé teszi a felhasználó számára, hogy naprakész információkkal rendelkezzen a piaci helyzetről. Emellett különböző grafikonok és diagramok segítségével megjeleníthetjük a részvények árának historikus alakulását előre megadott intervallumok szerint, de lehetőség van egyéni preferenciák szerint is lekérdezni az árfolyamokat.

Az alkalmazás intuitív felhasználói felülettel rendelkezik, amely egyszerű navigációt és könnyű hozzáférést biztosít az összes funkcióhoz. Ennek köszönhetően, még azok a felhasználók is könnyen használhatják az alkalmazást, akik kevés tapasztalattal rendelkeznek a részvényadatok elemzésében.

Az alkalmazásnak számos felhasználási lehetősége van, legyen szó hobbi befektetőkről, professzionális elemzőkről vagy pénzügyi intézményekről. A részvényadatok megjelenítése és elemzése segít a felhasználóknak megérteni a piaci trendeket és az esetleges befektetési lehetőségeket. Ez az alkalmazás egy hatékony és egyszerű eszköz a részvényadatok vizualizáláshoz, amely segít a felhasználóknak megfontolt döntéseket hozni a tőzsdén.

### 1.2. A probléma bemutatása

A dolgozat legfőbb célkitűzése a fent megemlített tulajdonságokkal és képességekkel rendelkező alkalmazás elkészítése volt. Tekintve, hogy az adatok elemzése és vizualizálása voltak a legfőbb prioritások, ezért olyan programozási nyelvet kellet választani a fejlesztéshez, amely ezen technológia szükséleteknek megfelel.

Mivel a végtermék egy olyan asztali alkalmazás, amely interaktív felhasználói felülettel rendelkezik, a választásom a Python programozási nyelvre esett, azon belül is a Tkinter könyvtár felhasználásával készült el. Ezen kívül más közismert Python könyvtárak elemeit is felhasználtam a fejlesztés során: a nagyméretű adathalmazok kezelésére

Pandas, a nagy számításigényű matematikai műveletek elvégzésére Numpy, ezeknek az eredményeknek a grafikus ábrázolására Matplotlib könyvtárat használtam.

Mivel az adatok vizsgálatához és megjelenítéséhez nagy mennyiségű pontos adatra van szükség, ezért egy olyan platform használta volt szükséges, amely megbízható, napi szinten frissített, ugyanakkor visszamenőleg sokáig (lehetőleg a kezdetig) tartalmazza az árfolyamokat. A kiválasztásnál figyelembe vettet azt a gyakorlati tényezőt is, hogy ezeket az adatokat korlátlan alkalommal, lehetőleg a Python programozási nyelv segítségével lehessen letölteni.

Ezen tényezők átgondolása után a választás a Yahoo Inc. cég, Yahoo Finance leányvállalatára esett és annak szervereiről töltöttem le az adatokat a rendelkezésre álló Yfinance könyvtár segítségével. Ezen az oldalon, világszinten körülbelül 350.000 nyilvánosan kereskedhető cég részvényárfolyama érhető el bárki számára. A letöltött állományok napi felosztásban négy érték érhető el: Open (nyitó érték), High (napi maximum), Low (napi minimum), Close (záró érték).

### 1.2.1. A dolgozat fontosabb lépései

- Az alkalmazás elemeinek megtervezése;
- Megfelelő programozási nyelv illetve bővítmény kiválasztása a barátságos és interaktív felhasználói felület kialakítása céljából;
- A részvényadatok forrásainak összegyűjtése tesztelése;
- A program szerkezetének kialakítása;
- Az adatok letöltéséért és kezeléséért felelős alrendszer kifejlesztése;
- A felhasználói felületért felelős szerkezet kifejlesztése;
- A különböző szerkezeti egységek kifejlesztése;
- A különböző funkcionálitások implementálása és integrálása;
- Javítás és tesztelés.

## 1.3. Saját megközelítés

Ebben a fejezetben bemutatom a fejlesztés során felhasznált programozási eszközöket és, hogy miként használtam fel őket a fejlesztés során.

### 1.3.1. A fejlesztéshez használt programozási nyelv

Az applikáció tervezési fázisában is az volt az alapötlet, hogy egy olyan rendszert szeretnék létrehozni, amely elemzői-előrejelző és vizualizációs funkciókkal rendelkezik, nem beszélve arról, hogy ez a két dolog szorosan együttműködve kell létezzen, azaz, az elemzői rész eredményeit grafikusan kell a felhasználó számára tálalni.

Az alkalmazás megírásához választott Python egy nagyon népszerű és sokoldalú programozási nyelv, amelyet könnyű megtanulni és használni. A nyelv fő jellegzetességei közé

tartozik a könnyű olvashatóság, a gazdag szintaxis, valamint a moduláris és objektumorientált programozási paradigma támogatása. A nyelv előnyei között kiemelkedik a nagy közösségi támogatottsága és a rengeteg elérhető könyvtár és modul, amelyek széleskörű funkciók és eszközök használatát teszik lehetővé a fejlesztés során, ezáltal gyorsan és hatékonyan fejleszthetünk különböző alkalmazásokat és projekteket.

A Python rugalmas és platformfüggetlen, ami azt jelenti, hogy egyetlen kód bázis létrehozásával több operációs rendszeren is futtatható alkalmazásokat készíthetünk. Ez jelentős idő- és erőforrás-megtakarítást eredményez, hiszen nem kell átírni és vagy optimalizálni a különböző operációs rendszerek számára. A nyelv beépített támogatást nyújt a nagy adatmennyiségek feldolgozására, a mesterséges intelligencia és gépi tanulás alkalmazások fejlesztésére, valamint webfejlesztésre is. Az egyszerűség és a gyors prototípus készítési lehetőségek révén ideális választás a gyors és hatékony fejlesztési ciklusokhoz.

### 1.3.2. Matplotlib könyvtár

A Matplotlib egy sokrétű adatvizualizációs könyvtár, amely lehetővé teszi a változatos diagramok és grafikonok létrehozását. A könyvtár legfőbb jellegzetességei közé tartozik a sokoldalúság, amely lehetővé teszi számos diagramtípus, köztük vonaldiagramok, oszlopdiagramok és szórási diagramok használatát.

A Matplotlib könnyen használható és rugalmas, és támogatja a testreszabhatóságot, lehetővé téve a grafikonok részletes beállítását a címkéktől és tengelyek formázásától a színekig és stílusokig. A könyvtár széles körben használható adatvizualizációs feladatokban, és szorosan kapcsolódik más Python könyvtárakhoz, mint például a NumPyval és a Pandas-szal.

A Matplotlibnek ugyancsak nagy és aktív közössége van, amely folyamatosan fejleszti és bővíti a könyvtárat, valamint rengeteg dokumentációval és példával rendelkezik, amelyek segítenek a fejlesztőknek a hatékony és kreatív adatvizualizációk készítésében.

### 1.3.3. Pandas könyvtár

Már az alkalmazás tervezés során tudva levő volt, hogy nagy mennyiségű adat feldolgozása várható. Ez később, a működtetése során már végképp bebizonyosodott és ebből a célból egy másik Python könyvtárat, a Pandas-t használtam fel.

A Pandas egy hatékony adatmanipulációs és adatelemzési könyvtár Pythonban, amelyet gyakran használnak nagy és komplex adatkészletek kezelésére. A könyvtár legfőbb jellegzetességei közé tartozik a DataFrame struktúra, amely táblázatos adatszerkezetet nyújt, és lehetővé teszi a könnyű adatmanipulációt, szűrést és aggregációt.

A Pandas kényelmes módja az adatok betöltésének és mentésének különböző formátumokban, például CSV, Excel és adatbázisok. A könyvtár rendelkezik számos beépített funkcióval a hiányzó adatok kezelésére, adattranszformációra és időszorok kezelésére. Jól integrálódik más adatelemző és vizualizációs könyvtárakkal, mint például a NumPy és a Matplotlib, ami lehetővé teszi a széleskörű adatelemzési műveletek végrehajtását. A könyvtárnak ugyancsak erős közössége van, amely folyamatosan fejleszti és dokumentálja a Pandas-t, valamint számos online forrás és példa áll rendelkezésre a fejlesztőknek a hatékony adatkezelés és analitika eléréséhez.

### **1.3.4. Nupmy könyvtár**

A fejlesztett alkalmazás egy másik célkitűzése a különböző elemzési módszerek integrálása volt és tekintve, hogy ezek alapvetően matematikai műveletek és különféle algoritmusok kombinálásából jönnek létre, szükség volt egy olyan könyvtár széleskörű használatára, mely megfelel ennek az igénynek.

A kiterjedt matematikai funkciókat és tömbműveleteket biztosító NumPy könyvtár teljesen megfelel ezeknek a szempontoknak. A legfőbb jellegzetességei közé tartozik a hatékony tömbkezelés, amely lehetővé teszi a nagyméretű tömbök létrehozását és manipulálását. A NumPy tömbök gyorsabbak és hatékonyabbak, mint a beépített Python listák, mivel egymás mellett tárolja az elemeket a memóriában, ami hatékonyabb hozzáférést eredményez.

A könyvtár tartalmazza a matematikai műveletek széles skáláját, például lineáris algebrai műveletek, Fourier-transzformációk, véletlenszám-generálás és sok más. Lehetővé teszi a vektorizált műveletek végrehajtását, amelyek nagymértékben felgyorsíthatják a kód végrehajtását és optimalizált numerikus műveleteket eredményeznek. A könyvtár jól integrálódik más adatelemző és vizualizációs könyvtárakkal, például a Pandas és a Matplotlib. A NumPy nagyon népszerű a tudományos és adatelemző közösségekben, és talán ő rendelkezik a legaktívabb közösséggel, amely folyamatosan fejleszti és támogatja a könyvtárat.

### **1.3.5. Tkinter könyvtár**

A korábban felsorolt könyvtárak mondhatni az alkalmazás szentháromságát képezik, ugyanakkor szükség volt ezek összefoglalására egy felhasználói keretbe. Ennek a szerepkörnek a betöltésére alkalmas a Tkinter, amely egy beépített grafikus felhasználói interfész (GUI) könyvtár Pythonban.

A Tkinter már alapból része a Python telepítésének, legfőbb jellegzetességei közé tartozik a könnyű használat. A könyvtár támogatja a széles körben használt widgeteket és beépített elemeket, mint például gombok, címkék, beviteli mezők, listák és sok más. Lehetővé teszi a felhasználói interakció kényelmes kezelését és az eseményekre való reagálást.

A könyvtár egyszerű és intuitív módon lehetővé teszi a felhasználói felület tervezését és testre szabását. Kompatibilis más népszerű GUI könyvtárakkal és keretrendszerrel, például a Ttk-vel és a Pygame-mel, emelet platformfüggetlen, így a létrehozott alkalmazások futtathatók Windows, macOS és Linux rendszereken egyaránt.

A Tkinter esetén is rengeteg szabad felhasználású információ és példa áll rendelkezésre, mivel ugyancsak aktív közösséggel rendelkezik, amely folyamatosan fejleszti és támogatja.

### **1.3.6. Yfinance könyvtár**

Végezetül, de nem utolsó sorban, szükség volt egy olyan könyvtárra is, amelyik már maguknak a feldolgozandó adatoknak a biztosításával, letöltésével foglalkozik a Python keretein belül. Több lehetőség megvizsgálása után a választás a Yahoo Financre esett, amely esetén elérhető volt a Python kompatibilis Yfinance könyvtár.

A legfőbb jellegzetességei közé tartozik, hogy lehetővé teszi a részvények, indexek és más pénzügyi instrumentumok történelmi adatainak letöltését. A Yfinance széles körben támogatott és könnyen használható API-t biztosít a pénzügyi adatok lekérdezéséhez és feldolgozásához és könnyen kezelhető adatszerkezeteket, például Pandas DataFrame-et eredményez.

A könyvtár rugalmas és könnyen testreszabható, paramétereket és beállításokat kínál a letöltött adatok szűréséhez és időszakokhoz igazításához. Az Yfinance nagyszerű eszköz a pénzügyi adatok elemzéséhez, a portfólió kezeléshez és az automatizált kereskedéshez Pythonban.

## 1.4. Az alkalmazás működési elve

Az alkalmazás felépítésében nagyban követi az objektum orientált programozási elvet, vezérlése a polimorfizmusra támaszkodik. A szoftver alapvetően két elkülöníthető részből áll: a tőzsdei adatok letöltéséért, frissítéséért és előfleholgozásáért felelős rész, valamint maga az applikáció amely magában foglalja az felhasználói felületet és az összes funkcionálitást.

### 1.1. kód részlet. Example Python code

```
def main():

    main_thread = threading.current_thread()
    data_refresh_thread = threading.Thread(target=refresh_data,
        args=(main_thread,))
    data_refresh_thread.start()

    print("INITIALIZING DATA...")

    root = Tk()

    page_manager = PageManager.PageManager(root)
    page_manager.create_page("page01", Page01.xPage01)
    page_manager.create_page("page02", Page02.xPage02)
    page_manager.create_page("page03", Page03.xPage03)
    page_manager.create_page("page04", Page04.xPage04)
    page_manager.create_page("page05", Page05.xPage05)
    page_manager.create_page("page06", Page06.xPage06)
    page_manager.create_page("page07", Page07.xPage07)
    page_manager.create_page("page08", Page08.xPage08)

    print("APPLICATION START...")
    page_manager.show_page("page01")

    root.mainloop()

main()
```

A két nagyobb szerkezeti egység működésében el van különítve egymástól, de jelenleg minden rész indítása ugyanabból a forrásból indul, ami a teljes rendszert beindító kezdeti hívás. Ahhoz hogy a két rendszer párhuzamosan és függetlenül tudjon működni, szakosításra volt szükség. Erre a feladatra a Threading könyvtárat használtam, amely lehetővé teszi a Python alapú programok esetén az egyszerre több processzor szálon futó folyamatok működtetését.

A kezdeti meghívás után elindítok egy új szálat és azon keresztül végeztetem el az adatbázis feladatokat, míg az eredeti szálon létrehozom a Tkinter könyvtár Tk() függvénye segítségével a felület fő ablakát és ezzel együtt az eseményfigyelő ablak Manager-t. Innentől kezdve ez az egység lesz felelős azért, hogy a különböző alegységek-widgetek által kiváltott eseményeket kezelje és az azokhoz hozzákötött függvényeket meghívja.

Mivel az alkalmazás több egységből-oldalból áll össze, melyeknek különböző funkcionálitása van, ezért a használat leegyszerűsítve annyiból áll, hogy ezek között az ablakok között kell oda-vissza lépkedni és az azokon végzett műveletek fogják magát a felhasználói élményt adni. Ebből már jól látszik, hogy szükséges lesz egy olyan mechanizmusra, ami ezt az oldalak közötti navigálást vezérli. Erre a célra létrehoztam egy PageManager osztályt, ami amely innentől kezdve azért volt felelős, hogy egyik oldalról át lehessen tértani egy másikra.

## 1.2. kód részlet. Example Python code

```
import tkinter as tk

class PageManager:
    def __init__(self, root):
        self.root = root
        self.pages = {}

    def show_page(self, page_name):
        page = self.pages.get(page_name)

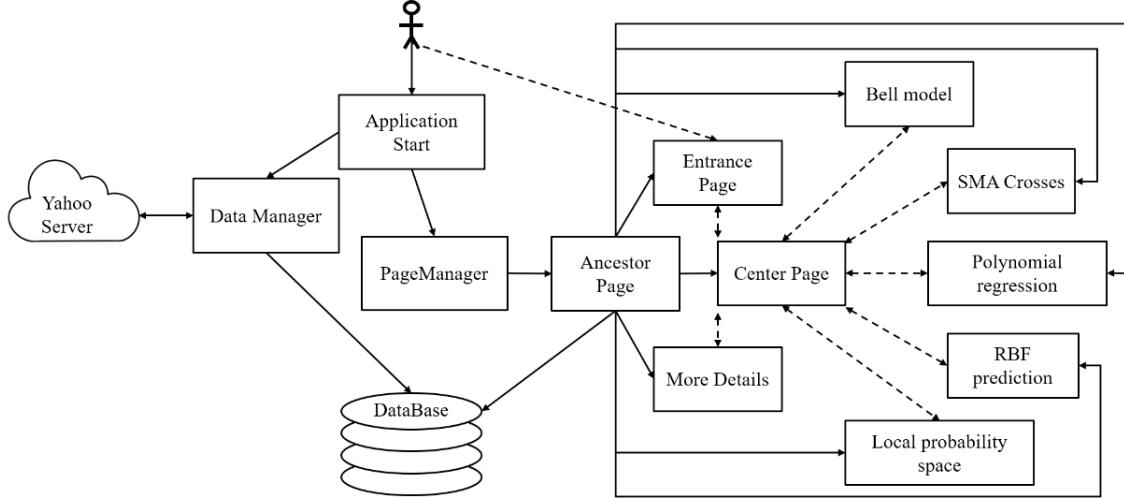
        if page:
            page.show()
        else:
            raise ValueError(f"Page '{page_name}' does not exist.")

    def create_page(self, page_name, page_class):
        page = page_class(self.root, self)
        page.place(x=0, y=0, relwidth=1, relheight=1)
        self.pages[page_name] = page
```

Ennél a pontnál már kiderül, hogy az egész alkalmazást az objektum orientáció paradigmájában érdemes írni, egy szóval szükség lesz egy ōsosztályra/óspage-re, és innentől kezdve a külön funkcionalitásokat megtestesítő oldal már ennek a származtatott osztálya lesz. Ez nem csak a kód írás spórolása miatt hasznos, hanem azért is, mert így a vezérlést végző PageManager a saját adatszerkezetében tárolhatja és hivatkozhat ezekre az elemek-

re, kihasználva a polimorfizmus [Mil78] és a helyettesíthetőség (Liskov helyettesíthetőség) [L<sup>+</sup>77] tulajdonságait, anélkül, hogy elrontaná a program működését.

A program a következő szerkezet szerint működik:



**1.1. ábra.** A program szerkezete

## 1.5. Szerkezeti elemek

### 1.5.1. Adatkezelő rész

Az adatkezelő rész, mint azt, ahogy korábban már leírtam, függetlenül egy külön szalon működik a teljes rendszeren belül. A működése alapvetően nagyon egyszerű, mivel az egész egy manuálisan szerkesztett boot fájlból indul ki, mely tartalmazza azoknak a cégeknek a legalapvetőbb adatait, amelyek árfolyamát le akarjuk tölteni.

A boottfájl azért szerkeszthető manuálisan, mert így utólag bármikor hozzá lehet adni újabb vállalatokat is a rendszerhez, anélkül, hogy újra kellene fordítani az egész kódot. Esetünkben ez egy csv fájl és ennek tartalma két oszlop: a cégek neve és tőzsdei szimbóluma. Miután ez a fájt betöltöttük, akkor megkezdődik a fájl betöltése után elkezdődik az adatok szerverről történő letöltése. Ezért fontos, hogy amikor először, vagy a későbbiekben frissítés céljából töltjük le az árfolyamokat, szükségünk lesz internet kapcsolatra. Az alkalmazást lehet használni internet kapcsolat nélkül is, de ebben az esetben értelemszerűen az eszközre letöltött, már meglévő adatokat lehet feldolgozni.

A letöltés az a része a folyamatnak, ahol a Yfinance könyvtárat használjuk, amely nem egy bonyolult folyamat, hiszen néhány paraméter megadása utána már korlátlan alkalmallal lehet ezt a művelet elvégezni. A legfontosabb paraméter a letöltendő cégek tőzsdei szimbóluma vagy „ticker”-je mivel az API ez alapján azonosítja be a megfelelő adathalmazt. A többi paraméter már inkább csak a letöltendő adatokra vonatkozik, a legfontosabbak a „period” és az „interval”, ami a letöltendő árfolyamok periódusára vonatkozik, esetünkben ez „max”, ami azt jelenti, hogy minden elérhető adat letöltünk, valamint az interval az árfolyamok felosztására vonatkozik, ami az alkalmazás esetén „1d”, azaz napi felosztású. A letöltött adatok egy DataFrame adattípus formájában kerülnek a rendszerbe.

Mielőtt a program lementené az adatokat az eszközre, elvégez néhány számítást. Összesen öt darab új információt számol ki a rendszer, mégpedig a 13, 50 és 200-napos mozgóátlagot valamint az 50 és 200-napos mozgó szórást. Ezeknek a lépéseknek az elvégzésére azért van szükség, mert több későbbi elemző funkció esetén is fel lesznek használva, így elkerülendő az, hogy minden egyes új betöltés esetén újra meg újra ki kelljen ezeket számolni.

Miután ez a lépés is megvan, akkor már csak az eszközre való lementés van hátra minden egyes esetben ugyancsak .csv fájltípusba. Erre, és később az összes adatkezelési feladatkörre, a Pandas könyvtárat használtam.

### 1.3. kód részlet. Example Python code

```
def download_one_company_data(ticker):

    data = yf.download(
        tickers = ticker,
        progress= False,
        period = "max",
        interval = "1d",
        ignore_tz = False,
        group_by = 'ticker',
        auto_adjust = False,
        prepost = False,
        threads = False,
        proxy = None
    )

    data.index = pd.to_datetime(data.index, format = '%Y-%m-%d
        %H:%M:%S').strftime('%Y-%m-%d')

    data['SMA13'] = data['Open'].rolling(13).mean().fillna(data['Open'])
    data['SMA50'] = data['Open'].rolling(50).mean().fillna(data['Open'])
    data['R_STD50'] = data['Open'].rolling(50).std().fillna(data['Open'])
    data['SMA200'] = data['Open'].rolling(200).mean().fillna(data['Open'])
    data['R_STD200'] = data['Open'].rolling(200).std().fillna(data['Open'])

    data.to_csv("E:/SZAKDOLGOZAT/A_DOLOGZAT/downloads/" + str(ticker + ".csv"))
```

## 1.5.2. Felhasználói felület oldalai

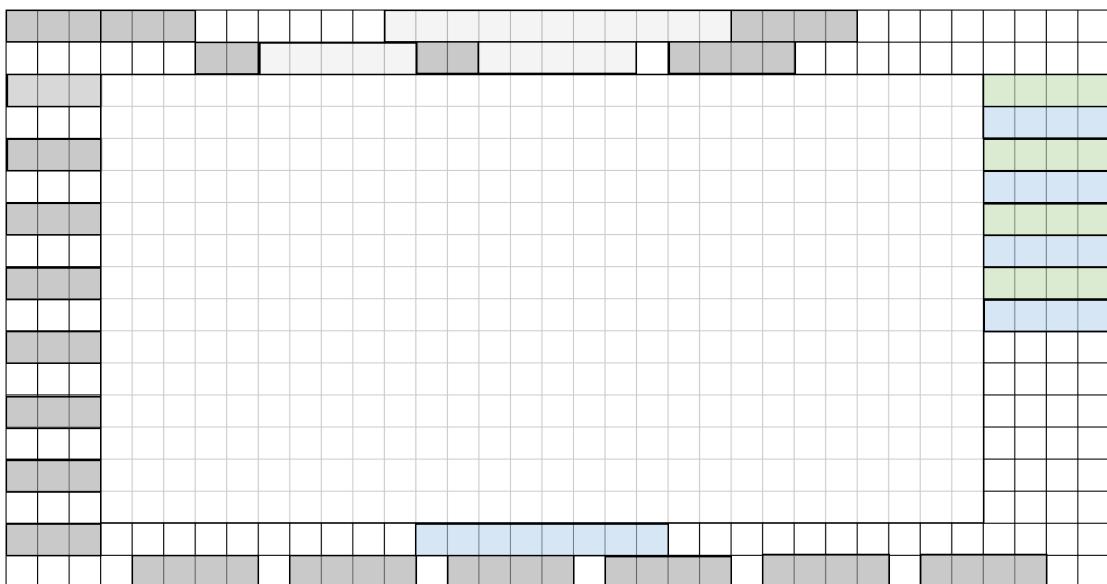
A felhasználói felület oldalai igen egyszerű, de ugyanakkor dinamikus és reaktív elv szerint működnek, azaz, automatikusan változnak a felületek méretei és velük együtt változik minden egyes látható elem. Alapvetően két részre lehet bontani rendszert: adattagok és widgetek.

Az adattagok esetén a legtöbb elemet az űrosztályból veszik át az oldalak, de vannak statikus változók is, amelyekből rendszer szinten egy darab van és közösen fér hozzá minden egyes oldal. Ennek az a lényege, hogy mivel több információt ugyanúgy használ

fel minden egység, ezért hatékonysági szempontból előnyösebb azokat statikus változókon keresztül elérni.

A widgetek esetén olyan alapértelmezett elemeket használok, mint gombok, címkék, beviteli mezők valamint az applikáció lényegét képező beágyazott grafikus ablak, ahol megjelennek az árfolyamok által leképezett görbék és más diagramok. A tervezés legfőbb szempontja egy arányos méretekből és szabályos elhelyezéssel felépülő felület volt, amelyek mérete dinamikusan és arányosan változik a teljes ablak nagyítása és vagy zsugorítása esetén.

Ennél fogva négyzetrács felbontás alapján állítottam be a felületet, mert így testre szabhatóan fel lehet építeni minden oldalt függetlenül a többitől. Ez azért volt fontos, mert habár a legtöbb oldal pontosan ugyanazokat az elemeket tartalmazza, ráadásul ugyanoda elhelyezve, bizonyos oldalak esetén teljesen más a tartalom és az elrendezés. Konkrét értékekről beszélve, az X-tengely szerint 35, míg az Y-tengely szerint 18 részre van bontva a felület, amely arány az általános képarany kerekítése.



**1.2. ábra.** A felhasználói felület szerkezete

A felület változása esetén, az egységek négyzetek adott tengely szerinti oldalhossza is változik, azaz, a négyzetrács felbontási száma megmarad, de már nem ugyanazokkal az oldal értékekkel. Ennek menedzselésére minden oldal esetén le van kezelve az ablakméret változás eseménye, ahol be van állítva annak frekvenciája, ki van számolva minden arányszám és minden egyes widget is újra van méretezve.

#### 1.4. kód részlet. Example Python code

```
def show_widgets(self):
    self.root.bind("<Configure>", self.resize_window)
    self.companies.set(self.company_data[0])

    self.resize_window(None)
    self.change_initial_vizualization()
```

## 1.5. kódrészlet. Example Python code

```
def resize_window(self, event):
    self.end = time.time()

    if self.end - self.start < 0.2:
        return

    self.start = time.time()

    self.x_len = self.root.winfo_width()
    self.y_len = self.root.winfo_height()

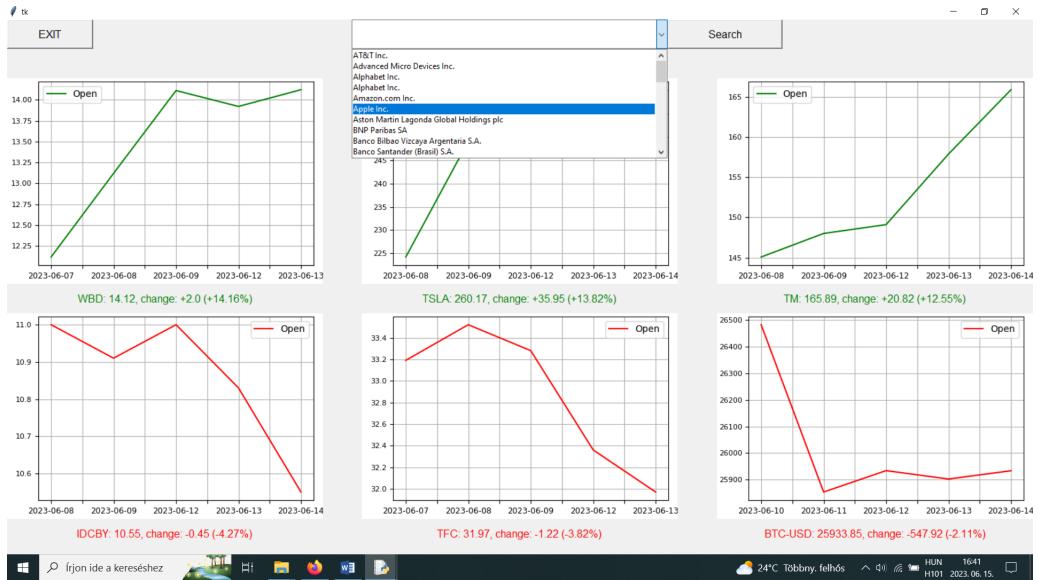
    self.box_x = int(self.x_len/35)
    self.box_y = int(self.y_len/18)

    self.x_font = self.box_x * 20/100
    self.y_font = self.box_y * 18/100
    self.label_font = self.box_x * 28/100
```

### 1.5.3. Nyitó oldal

Az alkalmazás előkészítő folyamatainak lezárása után a felhasználó számára először a nyitó oldal jelenik meg. Ezen az oldalon az alkalmazás mutatja az felhasználónak az elmúlt egy hét három legnagyobb százalékos növekedésű és három legnagyobb százalékos csökkenésű cégek egyhetes részvényárfolyam változásainak grafikonját és a konkrét értékeket. Ezt hat darab beágyazott grafikus ablak és hat darab címke segítségével valósítom meg.

Erről az oldalról kétféleképpen lehet továbbmenni: egyfelől rá lehet kattintani magukra az grafikonokra és akkor az adott cég az adataival megyünk tovább, a második azaz, a központi ablakra, illetve rendelkezésünkre áll egy beviteli mező is, ahova beírhatjuk és vagy görgetéssel kereshetjük a rendszerben szereplő céget és a „Search” gombra kattintva szintén a második oldalra navigálunk, ahol ugyancsak a kiválasztott cég adatai fognak megjelenni.



**1.3. ábra.** Második oldal

A bevíteli mező-Search kombináció az összes további oldalon elérhető, de esetükben nem vezet más oldalra, hanem csak simán az újonnan kiválasztott cég adatainak betöltéséhez és a beállított intervallum szerinti árfolyam görbe megjelenítéséhez vezet.

#### 1.5.4. Központi oldal

A második oldalt lehet tekinteni a központi oldalnak, mivel mindenképpen ide fogunk navigálni az első oldal után, és csak innen tudunk tovább menni az összes többi aloldalra, azokról pedig megint ide kell visszajöjjünk egy másik oldalra való váltás végett. Ennek az oldalnak négy új fő szerkezeti egysége van.

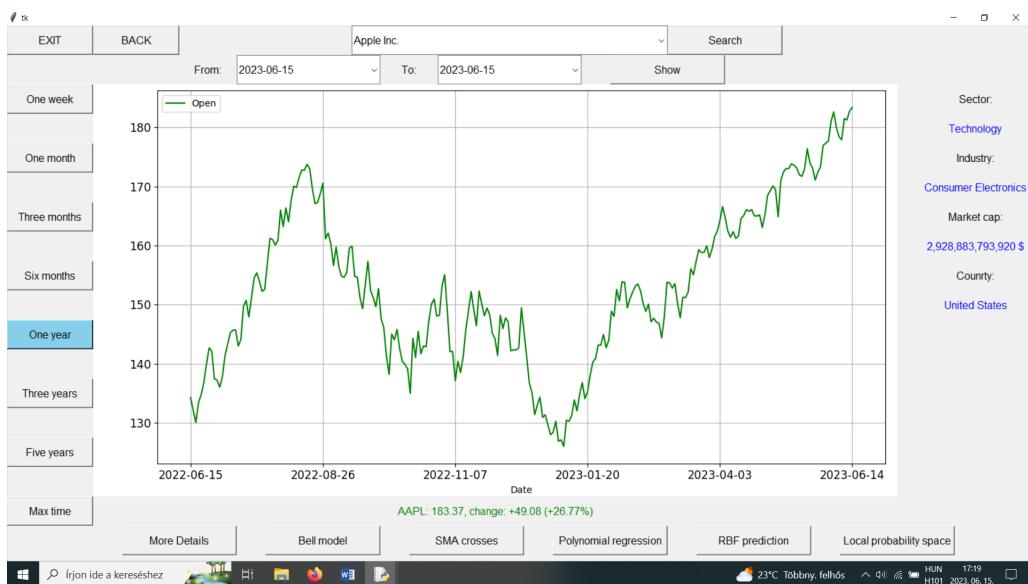
Elsőként az ezúttal egyetlen beágyazott grafikus ablak. Értelemszerűen ez sokkal nagyobb mint az előző oldal ablakai, az X-tengelyen az árfolyam adat pontjainak a dátumai, Y-tengely szerint meg maguk az árfolyamok. Amikor erre az oldalra kerülünk vagy új cég adatai töltődnek be, vagy ha már korábbi interakciók során volt kiválasztva cég, akkor automatikusan annak az utolsó egy hét adatai vannak kirajzolva. Az ablak alatt még szerepel egy címke is, amely leírja a grafikonon látható legfontosabb információkat, mint: a cég tőzsdei szimbóluma, a változás pénzbeli és százalékos értéke. Ezek az információk zölddel vannak kiírva amennyiben növekedés volt tapasztalható a kiválasztott perióduson és pirossal ha csökkenés. A további oldalak esetén ez a rész is megtalálható és nagyjából ugyanúgy működik.

A második egység a vizualizálandó árfolyam intervallum váltásához szükséges gombok és dátum bevíteli mezők. Az ablak bal oldalán előre konfigurált nyolc darab gomb található, amelyek a következő időszakokra vannak beállítva: az utolsó egy hét, az utolsó egy hónap, az utolsó három hónap, az utolsó hat hónap, az utolsó egy év, az utolsó három év, az utolsó öt év valamint a teljes árfolyam sorozat. Még ide tartozik a keresőmező alatti sorban található két dátum bevíteli mező és az aktiváló gomb, melyek segítségével tetszőleges időszakot lehet kiválasztani és az alatti árfolyamot megjeleníteni. Annak érdekében, hogy mindig lehessen tudni, hogy pontosan melyik beállítás szerint van meg-

jelentetve az árfolyam, ezért amikor egy aktiválás megtörténik, akkor annak a gombja világoskék színűre vált. Ez a funkcionalitás a többi oldal esetén is fennáll.

A harmadik egység, amely a felület jobb oldalán található, a cégekkel kapcsolatos plusz információk megjelenítéséért felelős. Összesen négy darab információ jelenítődik meg: a cégek szektorá, a cégek iparága, a cégek pillanatnyi piaci összértéke valamit a cégek származási országa. Az alkalmazás jelenleg nem végez plusz számításokat ezen információk alapján, de belátható, hogy komoly lehetőség van az ezen címkék alapján összeállított kapcsolatrendszerek beható vizsgálatában, mint például a cégek részvény árfolyamának korrelációja.

A negyedik egység a további vizsgálatokat végző aloldakhoz való navigáláshoz szükséges gombokat tartalmazza. Ez a rész a felület alján található és hat darab gombot tartalmaz: More Details, Bell Model, SMA Crosses, Polynomial regression, RBF prediction és Local probability space. Amennyiben a felhasználó rákattint ezekre, akkor azonnal átkerül a kiválasztott funkcionálitást végző oldalra.



**1.4. ábra.** Második oldal

Nem szerkezeti egység, de fontos megemlíteni a bal felső sarokban megjelenő „Exit” és „Back” gombokat, amelyek az applikáció lezárását és az előző oldalra való visszalépést aktiválják. A központi oldal esetén a visszalépés az nyitó oldalra visz, de a többi aloldal esetén minden erre az oldalra kerül vissza a felhasználó.

### 1.5.5. More Details oldal

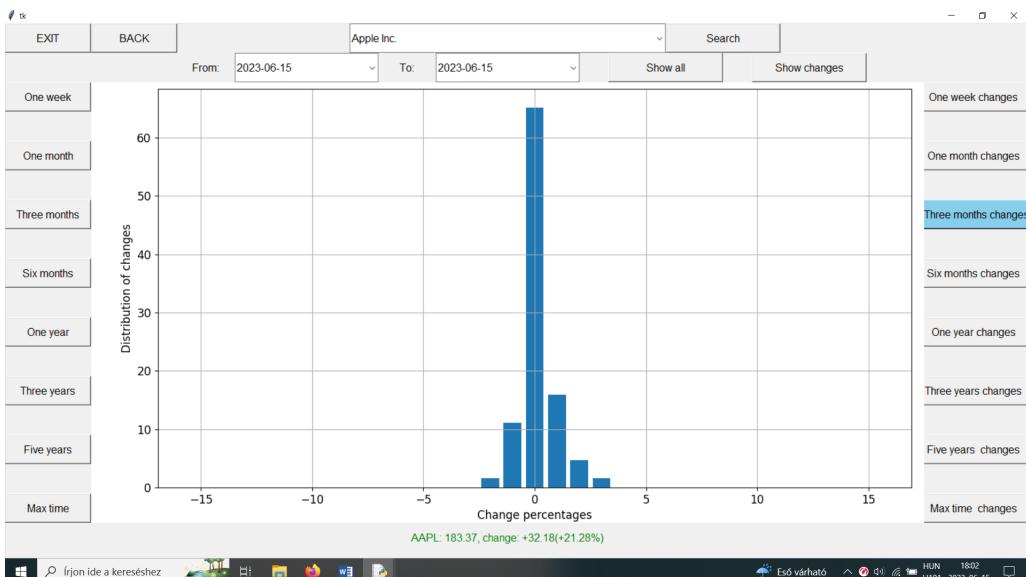
Az első többlet információt biztosító funkcionálitás a More Details oldal, amely még lényeges elemzést nem végez, de további már alapból elérhető adattal bővíti a megjelenítést. A szerkezeti egységek terén az első és a második ugyanazt a szerepet tölti be, a negyedik értelemszerűen nem létezik, míg a harmadik helyett egy tényleges új funkcionálitási csomag került be.

Az első két egység terén annyi változás azért történt, hogy itt nem csak a napi nyitó árfolyam görbéje van megjelenítve, hanem minden elérhető érték.



**1.5. ábra.** Bővített a megjelenítés

A harmadik egység helyett itt a másodikhoz hasonló előre beállított nyolc darab gomb kap szerepet, de esetükben már nem az árfolyamgyörbék jelennie meg, hanem az adott intervallumon végzett százalékos változások darabszámának hisztogramja. Ezt úgy kell elképzelni, hogy a lekérdezett periódusban szereplő árfolyamból készül egy differenciált sorozat, ami az egymás utáni napok egymáshoz viszonyított százalékos változását jelenti. Utána, ebben a sorozatban az egész értékre való kerekítés után megszámolódik, hogy azok hány alkalommal fordultak elő és az eredeti periódus mérési darabszámának függvényében százalék alapú hisztogram készül a napi változások darabszámai alapján.



**1.6. ábra.** Hisztogram

Ezzel a módszerrel egy másik szemszögből is meg lehet azt nézni, hogy milyen tendenciában van az éppen betöltött cég árfolyama. Mivel itt is fennállhat az egyéni intervallum

lekérdezése, ezért a dátum beviteli mező sorában a második egységhez tartozó gomb mellett, melynek neve „Show All”, beillesztésre került egy „Show Changes” gomb is, amely a beállított periódus hisztogramjának vetítését fogja aktiválni.

### 1.5.6. Harang modell

A Harang modell, vagy Bell model esetén is megtartjuk az előző funkció felületi elrendezését, azaz, középen a grafikus ablak, jobb és bal oldalon előre kalibrált gombok, középen fent pedig a tetszőleges dátum beállítás beviteli mezői és gombjai. Ugyan akkor, értelemszerűen más eljárásokat alkalmazását váltják ki a gombok: adott mozgó átlag és mozgó szórás alapján készített harang modell. Két megközelítést vizsgálok: ötven és kétszáz napos mozgó értékek.

A felület bal oldalára helyezett gombok az ötven napos mozgó értékek feldolgozásáért felelősek, míg a jobb oldalon lévők a kétszáz napos mozgó értékek ábrázolását és vizsgálatát váltják ki. Ez ugyanúgy igaz a dátum beviteli mezők sorába található gombok esetén is, mivel itt is a baloldali az ötven, míg a jobb oldali a kétszáz napos mozgó értékekért felelős.

Ennek az oldalnak az előrejelző funkciója a Chebyshev egyenlőtlenség alkalmazásán alapszik. Tekintve, hogy minden mérési pontra rendelkezésünkre áll a pillanatnyi érték, az átlag és a szórás, ezért a Chebyshev egyenlőtlenség alapján ki tudjuk számolni azt, hogy a pillanatnyi árfolyam mekkora valószínűséggel következhetett be [SYM84], és ennek az egyenlőtlenségnek a tovább gondolásával azt is ki lehet számolni, hogy mekkora valószínűséggel merre mozdulhat el:

$$Pr(|X - \mu| \geq k \cdot \sigma) < \frac{1}{k^2} \Rightarrow Pr(\text{elmozdulás}) \geq 1 - \frac{1}{k^2}$$

Ennek a számításnak az eredménye a grafikus ablak alatt jobb oldalon található. Alapvetően három dolgot lehet kijelenteni: az elmozdulás irányát nem lehet meghatározni, adott valószínűséggel fog csökkeni, valamint, hogy adott valószínűséggel fog növekedni. Ennek a vizsgálati módszernek a bővítésére rengeteg lehetőség van, tekintve, hogy csak a pillanatnyi helyzet alapján számol pillanatnyi változási valószínűséget.



1.7. ábra. Harang modell

### 1.5.7. Mozgóátlag keresztek

A mozgóátlag keresztek módszer esetén, újból felhasználtam az ötven és kétszáz napos mozgóátlagot, de még hozzá vettem egy harmadikat, nevezetesen a tizenhárom napos mozgóátlagot. Ez a három mozgóátlag a tőzsdei elemzésben közismert metrikák és a piac rövid (tizenhármas), közép (ötvenes) és hosszútávú (kétszázas) állapotát hivatottak felmérni.

Ezeknek az átlagoknak az egymáshoz viszonyított állapota alapján meglehetősen hasznos észrevételeket lehet tenni, kiváltképp, amikor a rövidtávú és hosszútávú átlagok kapcsolatát vizsgáljuk [Gen96].

Ennek az oldalnak a felületi elrendezése a következőképpen alakult: középen továbbra is a grafikus ablak ezúttal négy görbével (a nyers árfolyam és a három mozgóátlag), bal oldalon és a grafikon fölött a periódus váltásért felelős gombok és ismét a felület jobb oldalán szerepelnek az újonnan kiszámolt információk, amelyek minden a mozgóátlagokra vonatkoznak és elemenként két értéket jelentenek.

Az első érték szimplán csak a mozgó átlag százalékos változása ugyanúgy kiszámolva, ahogy az eredeti árfolyam esetén, azaz a vetített periódus első és utolsó értékének a különbsége lett kiszámolva. A második új információ már valamivel érdekesebb, mivel minden napos mozgóátlag esetén, az adott vetített periódusban szereplő értékekre a program kiszámol egy lineáris regressziós egyenest, és ezeknek az egyeneseknek kiszámolja a dőlési szögét. Ez az érték azért fontos, mert így számbeli értéket kapunk az indexek általános trendjéről az adott perióduson, és nem esünk abba csapdába, hogy az esetlegesen a görbe végénél lévő növekvő vagy csökkenő irányt, az egész rendszer irányának vesszük. Emelet, az egyéni trendértékeket összehasonlíthatjuk egymás között is, és ezek relatív kapcsolata is további információt jelent a felhasználó számára.



1.8. ábra. Mozgóátlag keresztek

### 1.5.8. Polinomiális regresszió

A következő elemzési módszer a Polinomiális regresszió elvégzése az eredeti árfoleyamon, és a keletkezett modell használata a predikcióhoz. Az oldal alapfelépítése nem változott, a grafikus ablak, a gombok és a dátum beviteli mezők a szokott helyükön. Az új és módszerspecifikus mezők ismét a jobb oldalon kerültek elhelyezésre. Ezzel együtt, egy fontos használat béli különbség áll fenn, ugyanis a felhasználónak be kell állítani a regressziós modell fokszámát és a predikciós eltolás paramétereit is.

Mindkét paraméter esetén van egy-egy megadott alapbeállítás, de mivel egy modell nagyban függ attól, hogy mi az eredeti vizsgálandó periódus, amire rá kell építeni, ezért a felhasználónak kell ezeket is igazítani. A rendszerbe be van építve „Warning” és „Error” funkció is, ami azt jelenti, hogy figyelmeztetni fog, ha a megadott paraméterek kétségesek a további számítások eredménye végett, mint például túl kicsi vagy nagy fokszám és eltolás, vagy nem is engedi a lefutást, ha egyértelműen hibásak a paraméterek, mint például nullás vagy az adatpontok számánál nagyobb fokszám, esetleg nullás eltolás.

Miután a beállítások megtörténtek, ezután meg kell nyomni a számítást indító gombot és a gomb alatti sorokban azonnal megjelennek az eredmények. Összesen három érték jelenítődik meg, amiből kettő a modelre és egy már magára az előrejelzése vonatkozik.

A modellel kapcsolatban kiszámításra kerül a négyzetes hiba és az R négyzet hiba. Ezekre azért van szükség, mert az előrejelzésre ideális modell esetén nem egézséges, ha az túlságosan felületes, tehát túl nagy hibával dolgozik, és az se, ha túlságosan élesen követi le az eredeti adathalmazt, vagyis túl kis hibával dolgozik. A harmadik megjelenített érték a már az előrejelzett százalékos eltérés. Ennek esetén is be van építve egy figyelmeztető rendszer, ami túlságosan nagy kilengések esetén aktiválódni fog és felhívja erre a figyelmet.



1.9. ábra. Polinomiális regresszió

### 1.5.9. RBF közelítés

A RBF közelítés elemzési módszere kvázi testvér módszer a regressziós módszerrel. Az oldal felépítése szinte ugyanaz: grafikus ablak, periódus kezelő gombok és beviteli mezők, a létrehozandó modell beállításához szükséges paraméterek beviteli mezői és a kiszámított számbeli eredmények a modell és a predikció esetén. Hasonlóan a korábbi oldalhoz, itt is be van építve egy figyelmeztető és hibajelentő funkció is.

Természetesen a hasonlóságokon túl, lényeges különbségek is fennállnak, hiszen ezúttal a közelítő modellt RBF háló segítségével számolja ki a rendszer. Ennek alapján már rögtön eggyel több beviteli mező található a jobboldalon, hiszen ki kell választani az RBF rendszer típusát is. Jelenleg három lehetséges verzió van: Gauss függvény alapú, koszinusz függvény alapú, valamint szigmoid függvény alapú bázisokkal operáló háló.

A számításokat ismét egy gomb lenyomásával kell elkezdeni, és ugyanazokat az értékeket számítjuk ki, melyek a modell élességét vizsgálják az eredeti árfolyamhoz képest és a predikció százalékos kilengését. Tekintve, hogy talán ez a leginkább parametrizálható elemző módszer a már elérhetők közül, a felhasználóknak rengeteg kísérletezésre van lehetőségük az ideális beállítások megtalálása végett, de további fejlesztések szintjén is komoly lehetőség áll fenn.



1.10. ábra. RBF közelítés

### 1.5.10. Lokális valószínűségi tér

A lokális valószínűségi tér módszere egy saját fejlesztésű módszer, részletében felhasználja a napi változások eloszlásának vizsgálatát, a polinomiális regressziót és a lehetséges jövőbeli értékek algebrai kiszámítását.

Az oldal elemeiben teljesen megegyezik a regressziós vizsgálat oldalával, viszont lényeges jelentés béli különbségek vannak. Ugyanúgy van grafikus ablak, periódus beállító gombok és beviteli mezők, modell fokszám és predikciós eltolási beviteli mező, négyze-

tes hiba, R négyzet valamint az előrejelzett eltérést megjelenítő címke. Ugyanakkor, a jobboldali elemek ezúttal nem teljesen arra vonatkoznak, mint a regresszió esetén.

A módszer lényege a következő: hasonlóan a More Details oldal egyik funkciójához, a beállított periódusról elkészül a napi százalékos változások előfordulási hisztogramja. Mivel ez a hisztogram kiegyenlítetlen, hiszen százalékos változások darabszáma nem fog egyenletesen terjedni a maximum darabszámtól a minimum darabszámig (középről a ball és jobb szélsőség felé), ezért a hisztogramra készül egy polinomiális regressziós becslés ami simább átmenetet fog adni az egyes egységek között, anélkül, hogy a tendencia elromlana. Ennek a modellnek a fokszámára vonatkozik az első beviteli mező valamit a közelítés minőségét írja le a négyzetes hiba és R négyzet hiba is.

Következő beviteli mező itt is a predikciós eltolásra vonatkozik. Magát a predikciót a következőképpen számolom ki: a százalékos változás vektorát elemenként összeszorzom a változások újonnan számított eloszlási vektorával, az eredmény vektort összeszorzom az utolsó ismert értékkel és összeadom az eredményvektorban szereplő értékeket. Ezt az összeget pedig elosztom az eloszlási vektor elemeinek az összegével és így kapok egy súlyozott értéket. A további elemek esetén megismétlek ez a folyamatot, de az utolsó ismert érték helyét az eggyel korábban kiszámolt eredmény veszi át.

Ezzel a módszerrel azt érem el, hogy egy adott periódus változásait normalizálom napi szintű százalékos változásokra, majd pedig a végső soron súlyozott eredmény magába foglalja az öt megelőző periódus összetett tulajdonságait. Itt fontos megjegyezni, hogy nagyobb periódusok esetén, a változások statisztikai eloszlása a normál eloszlás felé igazodik, amely szimmetrikus mivolta miatt, ez a módszer semleges eredmény fog adni, de erről a második fejezetben még lesz szó.



**1.11. ábra.** Lokális valószínűségi tér

## 1.6. Összegzés

A bemutatott alkalmazás egy most is teljeskörű és bő eszközökészletet biztosít a tőzsdei árfolyamok ábrázolására és elemzésére. Hálá a Python könyvtárainak és a nagymér-

tékű testreszabhatóságnak, az alkalmazás fejlesztése során minden megtervezett vizuális is funkcionális elemet integrálni tudtam. Ezzel együtt rengeteg további fejlesztési lehetőség van, hiszen még rengeteg már ismert elemző és előrejelző módszerrel lehet bővíteni a funkcionálitásokat, nem beszélve a lehetséges sajátfejlesztésű módszerekről, tekintve, hogy a második fejezet ilyen lehetőségek kutatásáról szól.

# **2. fejezet**

# **KUTATÁS**

## **2.1. Bevezetés**

A részvénykereskedelem mai formája igen szofisztikált módszerek segítségével valósul meg. A 21-századra felhalmozódott egy óriási tudáshalmaz az adatelemzés terén, amely főként az internet segítségével mindenki számára elérhető és a nap mint nap gyorsuló tendenciával bővül.

Ugyanakkor, a legtöbb publikált módszer valamelyen közgazdasági heurisztikán alapuló stratégiát követ, nem pedig az adatok hátterében álló összefüggések alapos vizsgálatán alapszik. Az egyik legismertebb módszer ebben a kategóriában a Fundamentum Elemzés (angolul Fundamental Analysis). A Fundamentum Elemzés során egy adott cég és a részvényei ár-érték arányának a vizsgálata a tárgy, azaz, hogy a jelenlegi ár megfelel-e a vállalat lehetséges, jövőbeli értékének. Ilyen elemzések során is több tényezőt kell figyelembe venni, mint például a mindenkorai gazdasági helyzetet, a cég pénzügyi helyzetét és további kulcsmutatókat a vállalat részvényeinek valódi értékéhez [AB97].

Ezzel a felfogással valamelyest szemben áll a piaci mozgásokat leíró nyers adatok elemzése, mint például a Technikai Elemzés (angolul Technical Analysis). Ez magában foglalja a piaci tevékenységek során keletkezett adatokat, például a mennyiség és árak vizsgálatát. Az ilyen típusú részvényelemzést követő elemzők technikai mutatókat és eszközöket, például diagramokat és oszcillátorokat használnak, hogy azonosítsák azokat a mintákat, melyek a historikus adatokban fellelhetőek és amelyek jelezhetik a jövőbeni ár trendeket vagy irányokat [PI07].

Habár a két módszer megközelítésben különbözik egymástól, hiszen az első esetén sokkal inkább a közgazdasági mutatók révén elemzik az adott árfolyamot, míg utóbbi sokkal inkább egy információs folyamként kezeli azt, valójában nem kizártak egymást, hanem kiegészítik egymást, legtöbbször egyszerre alkalmazzák őket [PJ16].

## **2.2. A probléma bemutatása**

Bármelyik megközelítést is használjuk, így is-úgy is meg kell találjuk azokat a marrekereket, amelyek alapján a legjobb előrejelzéseket tudjuk számítani. Mivel mára már igen nagyra nőt a vizsgálható adathalmaz, ezért a kézi módszerek praktikus okokból már nem alkalmazhatóak, de a matematikai, azon belül is a statisztikai módszerek használata továbbra is alkalmas.

### **2.2.1. A kutatás fontosabb lépései**

- Az összegyűjtött adatok esetében meggyőződni arról, hogy nem állnak összefüggésben, ezzel is igazolva a vizsgált módszerek hitelességét;
- A nyers adatok természetének és típusának vizsgálata;
- A meglévő adatokat megtisztítása a zajuktól;
- A megtisztított adatokat osztályozása;
- Az adatok olyan módon történő transzformálása, hogy a vizsgált módszerek alkalmazhatóak legyenek rajtuk;
- Végül pedig a módszerek hatékonyságának elemzése és az eredmények összehasonlítása.

### **2.2.2. A kutatás során vizsgált kérdések**

1. Van-e bármilyen kapcsolat a vizsgált árfolyamok között, ha tudjuk, hogy a piacot befolyásoló események időben egyszerre következnek be?
2. Melyik adattípusba tartoznak a részvényárfolyamok?
3. Van-e a nyers adatokba kódolva bármilyen matematikai-statisztikai módon vizsgálható és vagy igazolható összefüggés?
4. Milyen módszerekkel lehet megtisztítani a nyers adatokat és miért jobb az egyik a másiknál?
5. Hányféléképpen lehetséges és vagy egyáltalán érdemes osztályokba rendelni az adatokat?
6. Milyen elérhető módszerek vannak az adatok átalakítására, hogy neurális hálókkal vagy osztályozó algoritmusokkal lehessen őket vizsgálni?
7. Következtetésként: A vizsgált módszerek segítségünkre vannak-e abban, hogy a piaci mozgásokat előre tudjuk jelezni?

### **2.2.3. Hipotézisek**

Ahogy az a dolgozat felvetésében ki van jelentve, a következő hipotéziseket állnak fenn:

**H00:** A neurális hálók által generált előrejelzések képesek a szabad piaci mozgás felett teljesíteni.

**H01:** A hírfolyamok elemzése során generált előrejelzések képesek a szabad piaci mozgás felett teljesíteni.

**H02:** A fent vizsgált két módszer kombinációja felülteljesíti minden két részmódszert.

## **2.3. Saját megközelítés**

Az ebben a kutatásban használt megközelítés gyakorlatilag a Technikai Elemzési fell fogásával egyezik meg, leginkább a gyakorlati megközelítésben kíván újat mutatni és célja olyan szerkezeti összefüggések felismerése, amelyekre alapozva nagy kapacitású módszerek segítségével a lehető leg pontosabb előrejelzéseket lehet adni, főként neurális hálók és osztályozó algoritmusok használatával. A kutatás célja, hogy az egy hónapi adatból a lehető leg pontosabban lehessen megjósolni a következő nap változását. Azért került kiválasztásra az egy hónapos ciklus, mert közgazdasági szempontból értelmezhető időegység és még belefér az aktuális számítási kapacitásba.

### **2.3.1. Az Információs források és a felhasznált adatok bemutatása**

Az asztali alkalmazás során már megfelelően leteszteltem a Yfinace könyvtárat és mivel megbízhatónak és kellően részletesnek bizonyult, valamint nem volt limitálva a letöltések szerint, ezért ebben a fejezetben is ezt a forrást használtam. Mivel négy különböző részvényindex érhető el napi szinten, ezért el kellett döntenem, hogy pontosan melyiket fogom felhasználni a vizsgálatok során. Mivel a napi maximum és minimum indexek gyakorlatilag véletlenszerűen alakulnak ki a nap folyamán, ezért ezt a kettőt egyből el is vettetem. Maradék kettőből végül a napi nyitó (Open) mellet döntöttem, mivel értelemszerűen az következik be leghamarabb, így valós időben ennek az elemzése során tudunk legkorábban további információkhöz jutni, nem utolsó sorban, lehetőségünk van még a tőzsden is reagálni az elemzés eredményére, tekintve, hogy az egész további része a napnak rendelkezésünkre áll. Ezzel szemben, a záró értéket csak a tőzsdei kereskedés befejeztével kapjuk meg, tehát leghamarabb a következő kereskedési nap kezdetével tudunk csak reagálni erre.

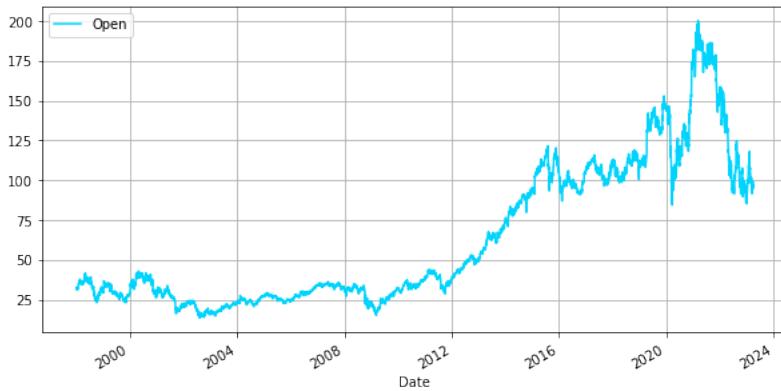
Következő lépésként ki kellet vállasszak néhány eltérő ipari környezetből származó céget, ezzel is diverzifikálva feldolgozott adatokat, biztosítva a vizsgált módszerek rugalmaságát és az eredmények validációját.

Ezek alapján a következő négy céget választottam: The Walt Disney Company - média és szórakoztatóipar, Ford Motor Company.- autóipar, UBS Group AG. - bank és pénzügy valamint Exxon Mobil Corporation - olaj és gáz.

## **2.4. A vizsgált cégek bemutatása**

### **2.4.1. The Walt Disney Company (továbbiakban DIS)**

A The Walt Disney Company-t, közismert nevén Disney egy amerikai multinacionális, tömegmédia és szórakoztatóipari konszern, amelynek központja a Burbankban (Kalifornia, Egyesült Államok) találhat. A Disney-t 1923-ban Walt és Roy O. Disney testvérek alapították Disney Brothers Studio néven. 2023. március 31-én a cég összértéke 180.84 milliárd amerikai dollár.

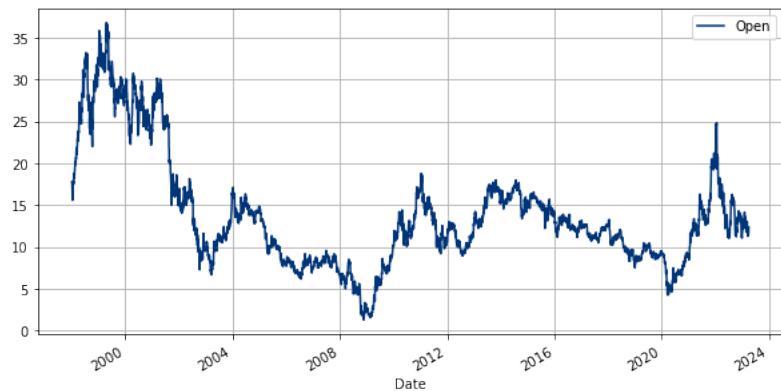


**2.1. ábra.** Walt Disney Company

#### 2.4.2. Ford Motor Company. (továbbiakban F)

A Ford Motor Company (közismert nevén Ford) egy amerikai multinacionális autógyártó, amelynek központja Dearborn-ban (Michigan, Egyesült Államok) található. Henry Ford alapította, és 1903-ban jegyezték be. A vállalat Ford márkanév alatt autókat és haszongépjárműveket, Lincoln luxusmárka alatt pedig luxus autókat értékesít.

2023. március 31-én a cég összértéke 50.23 milliárd amerikai dollár.



**2.2. ábra.** Ford Motor Company

#### 2.4.3. UBS Group AG. (továbbiakban UBS)

Az UBS Group AG svájci multinacionális befektetési bank és pénzügyi szolgáltató vállalat, amelyet 1998-ban alapítottak az Union Bank of Switzerland és a Swiss Bank Corporation egyesülésével. Központi székhelyei Zürich és Bázel (Svájc) városokban találhatóak, és a legnagyobb svájci banki intézeteként és a világ legnagyobb magán bankjaként minden jelentősebb pénzügyi központban jelen van.

2023. március 31-én a cég összértéke 69.95 milliárd amerikai dollár.

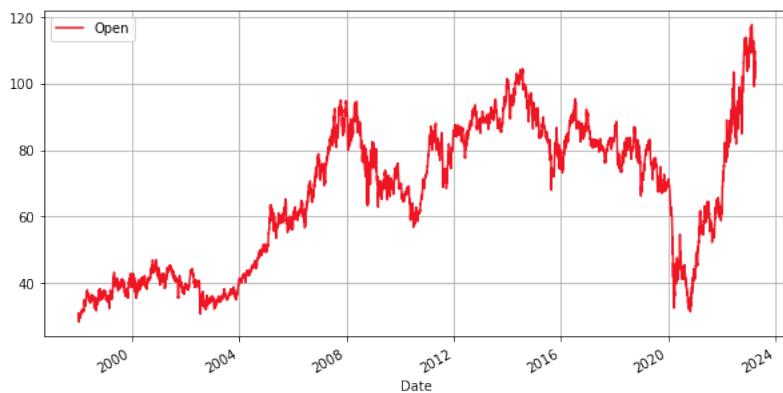


**2.3. ábra.** UBS Group AG

#### 2.4.4. Exxon Mobil Corporation (továbbiakban XOM)

Az ExxonMobil Corporation egy amerikai multinacionális olaj- és gázipari vállalat, amelynek központja Springben (Texas, Egyesült Államok) található. John D. Rockefeller Standard Oiljának legnagyobb közvetlen leszármazottja és 1999-ben jött létre az Exxon és a Mobil egyesülésével.

2023. március 31-én a cég összértéke 446.42 milliárd amerikai dollár.



**2.4. ábra.** Exxon Mobil Corporation

### 2.5. A vizsgált cégek árfolyama közötti kapcsolat

#### 2.5.1. Keresztkorreláció

Ahhoz, hogy e kutatás során vizsgált módszerek hatékonyságáról meg tudjunk bizonyosodni, fontos, hogy egymástól független adathalmazokon is leteszteljük őket. Lényegében ez volt a követett elv a cégek kiválasztása során, tekintve, hogy minden egymástól eltérő és független iparágak képviselői. Azonban, a tényleges bizonyosság kedvéért, szükséges ezt a feltételezést ellenőrizni.

Ennek a kapcsolatnak a vizsgálatára Keresztkorrelációs elemzést (Cross-correlation) használtam [Bou96]. A keresztkorreláció egy jellemző mérőszám, amelyet a jelek hason-

lóságának vizsgálatára használnak az egyik jel eltolásával a másikhoz képest. Ezt gyakran nevezik csúszó pontszorzatnak vagy csúszó belső szorzatnak is. A módszer gyakran alkalmazható hosszabb jelben való keresésre rövidebb, ismert jellemzők után kutatva. A kereszt-korrelációt alkalmazzák mintafelismerésre, a részecskékből álló anyagok elemzésére, az elektron tomográfiába, az átlagolás, a kriptoanalízis és a neurofiziológia terén is. A keresztkorreláció jellegében hasonló a két függvény konvolúciójához. Az autó korreláció, amely egy jel kereszt-korrelációjának speciális esete önmagával, mindenkor lesz egy csúcs a nullás időeltolásnál. A korreláció definíciója mindenkor standardizálási tényezőt tartalmaz azért, hogy a korrelációk értékei -1 és +1 között legyenek, ami annyit tesz, hogy -1-esetén negatív korreláció áll fenn (az egyik jel változásakor a másik azzal tökéletesen ellentétesen változik), míg +1 esetén tökéletes korreláció van (az egyik jel változásakor a másik pontosan ugyan úgy változik).

Ha  $f$  és  $g$  az  $X$  és  $Y$  két független véletlen változó sűrűségfüggvénye, akkor a különbség  $Y-X$  valószínűség-sűrűségfüggvénye a kereszt-korrelációban (a jelzés feldolgozási értelemben)  $f^*g$  formában megadott. Azonban ez a terminológia nem használatos a valószínűségszámításban és statisztikában. Ezzel szemben a  $fg$  konvolúció (amely ekvivalens a  $f(t)$  és  $g(-t)$  kereszt-korrelációjával) megadja az  $X+Y$  összeg valószínűségi sűrűségfüggvényét.

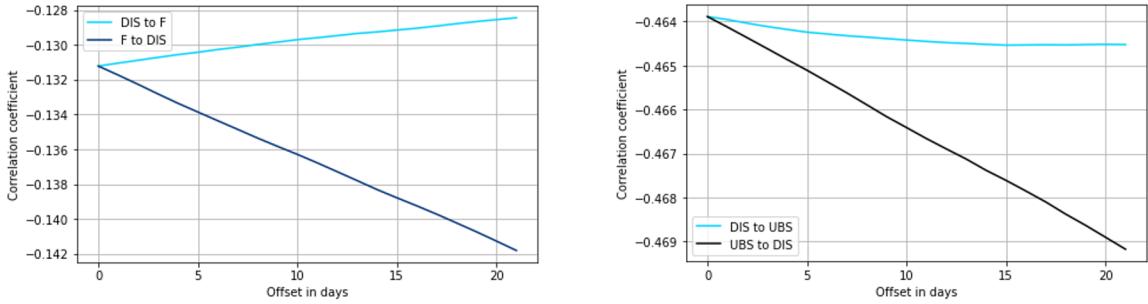
Erre a vizsgálatra Pearson-korrelációs együttható kiszámítását használtam [C<sup>+09</sup>]. A Pearson-féle korrelációs együttható (másként Pearson szorzatmomentum), két sorozat kovarienciájának hányadosa a szórások szorzatával. Definíció szerint magába foglalja a "szorzat-momentumot", vagyis az átlaggal korrigált valószínűségi változók szorzatának átlagát (az eredetre vonatkozó első momentumot) – innen ered a szorzat-momentum módosító a névben.

## 2.5.2. A tanulmányozott árfolyamok korrelációja

Mivel a kutatás során 21 napos periódusokból terveztem előrejelzéseket készíteni (ennek okát később kifejtem), ezért megvizsgáltam a négy cég egymással való korrelációját mindenkor irányba 0-tól 21-napos eltolásig. Ezek alapján a következő eredményeket kaptam:

Az **2.5-ös ábra bal** oldalán a DIS és F között fennálló korreláció van ábrázolva. Nullás eltolás esetén ez az együttható -0.131, az első eltolása esetén ez elkezd nőni egészen -0.128-ig, míg utóbbi eltolása esetén elkezd csökkeni -0.142-ig. Ez azt jelenti, hogy a kezdeti kapcsolat is nagyon minimális, az első eltolása esetén ez tovább romlik, de fordított esetben is csak nagyon kis javulás észlelhető.

Az **2.5-ös ábra jobb** oldalána DIS és UBS között fennálló korreláció van ábrázolva. Nullás eltolás esetén ez az együttható -0.464, az első eltolása esetén ez gyakorlatilag stagnál, míg utóbbi eltolása esetén elkezd csökkeni -0.469-ig. Ez azt jelenti, hogy a kezdeti kapcsolat itt már valamivel nagyobb, de még mindenkor nem számottevő, az első eltolása esetén ez nem változik, de fordított esetben is csak minimális közeledés van a negatív korreláció fele.



**2.5. ábra.** Az árfolyamok korrelációja

A DIS és XOM és között fennálló korreláció a kiinduló értéke 0.355, ami nem számottevő és mindenkor előtolása esetén, még ha keveset is, de ez az érték nagyságrendileg 0.348-ig romlik.

A F és UBS között fennálló korreláció kiinduló értéke -0.096, az első előtolása esetén ez -0.092-ig romlik, de fordítva is csak -0.118-ig erősödik az ellentétes kapcsolat.

Az F és XOM között fennálló korreláció kiinduló értéke -0.397, az első előtolás esetén ez -0.392-ig romlik, fordítva -0.414-ig erősödik az ellentétes kapcsolat.

Az utolsó, UBS és XOM között fennálló korreláció kiinduló értéke -0.159, az első előtolás esetén ez -0.145-ig romlik, de az fordítva is csak -0.185-ig erősödik az ellentétes korreláció.

Mindezekre alapozva ki lehet jelenteni, hogy a vizsgált árfolyamok között nem áll fenn semmilyen számottevő korreláció. Ebből következik, hogy a fenti idősorokon végzett tesztelés során nyert eredmények hitelesek.

## 2.6. Az adatok belső szerkezetének vizsgálata

### 2.6.1. Az ideális ablak meghatározása

A részvény adatok, de általánosságában véve minden idősor adat egy diszkrét sorozat. A mérések frekvenciától függően el lehet menteni az értékeket és az így kapott sorozatot utána lehet feldolgozni. A részvények esetében lehetőség van arra, hogy perces, vagy akár másodperces frekvenciában kérjük le adatokat, de ez elemzési szempontból értelmetlen. Az ilyen sűrűségű sorozat esetén az árfolyamváltozásokat a pillanatnyi kereskedés befolyásolja, ami egy valós idejű tevékenység és ebből adódóan teljesen előre jelezhetetlen. Erre alapozva a kutatás során az egynapos sűrűségű adatsorral dolgoztam.

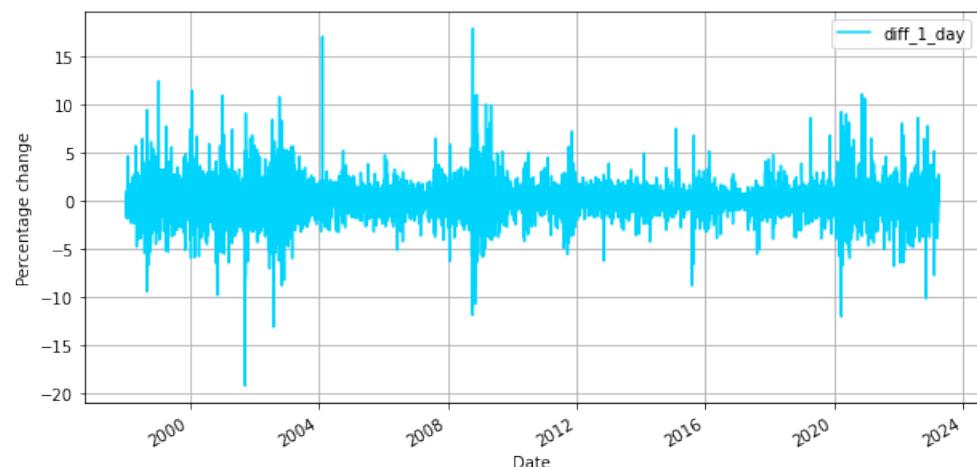
A megfelelő sűrűség kiválasztása után, meg kellett találjam azt az ideális méretű ablakot az adatsorban, amelynek elemzése már magában hordozhat olyan információkat, amelyeket egy mesterséges intelligencián alapuló rendszer már tud detektálni. Közgazdaságilag de intuitívan is igazolható, hogy erre az egyhónapos periódus megfelelő, így egy-egy vizsgált ablakot az adott hónap mérési adatai alkotnak.

Mivel a hónapok különböző napszámán túl még az a tényező is szerepet játszik, hogy a részvénykereskedeleml csak hétköznap zajlik, ráadásul ünnepnapok esetén a tőzsde zárva van, tehát nem lehet új adatot regisztrálni, azért szigorúan dátum szerint nem lehetséges egyező méretű ablakkal dolgozni. Ennek a jelenségnek a korrigálása végett kiszámoltam,

hogy átlagosan hány adatpontom van egy hónap során, a sokéves átlag eredményéül 20.(9669)-et kaptam, amit végül 21-re kerekítettem.

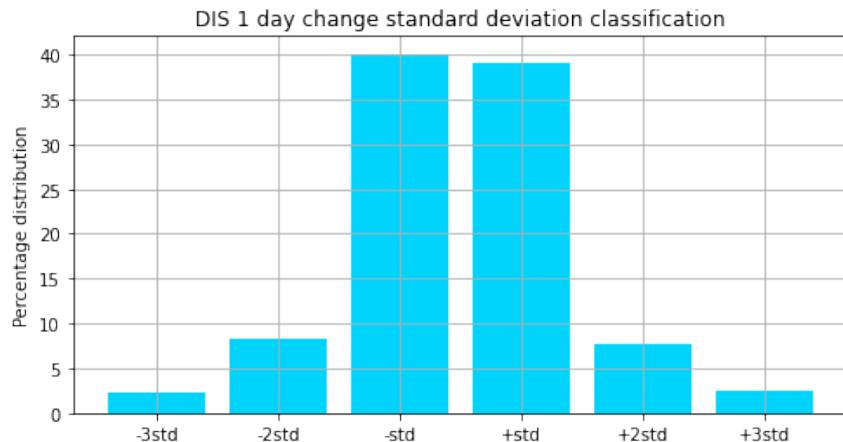
### 2.6.2. Reláció egymás utáni adatpontok között

Az előző fejezetben ki lett jelentve, hogy a részvényárfolyamok a diszkrét adatfolyamok csoportjába tartoznak így egy adott mérési pont információihoz az is hozzátarozik, hogy miként viszonyul a környezetéhez. Ennek a fejezetnek az a célja, hogy megvizsgálja, hogy fennáll-e valamilyen szervező elv az árfolyamváltozásban. Ennek vizsgálatához érdemes az adatokat átírni differenciált sorozatra. A differenciált sorozat annyit tesz, hogy nem magukat a nyers árfolyamokat vizsgáljuk, hanem az egymás utáni napok árfolyamai közötti százalékos változást. A DIS árfolyamának Open oszlopának differenciált sorozata a következőképpen alakul:



**2.6. ábra.** A DIS árfolyam

Észrevehető, hogy a differenciált sorozat meglehetősen zajos és az  $y=0$ -ás tengely körül mozog. Erre alapozva érdemes olyan statisztikai mutatókat kiszámolni, mint az átlag és a szórás, majd pedig megvizsgálni, hogy az árfolyamváltozásnak milyen eloszlása van ezek alapján. Az  $\text{avg}(\text{diff(DIS)}) = 0.036$ ,  $\text{std}(\text{diff(DIF)}) = 1.929$ , a szórások szerinti eloszlás a következő:  $-3\text{std} = 2.32\%$ ,  $-2\text{std} = 8.31\%$ ,  $-\text{std} = 40.09\%$ ,  $+\text{std} = 39.05\%$ ,  $+2\text{std} = 7.66\%$ ,  $+3\text{std} = 2.54\%$



**2.7. ábra.** A DIS hisztogram

A vizsgálat alapján ki lehet jelenteni, hogy a differenciált egyenlet változása erősen igazodik a normál eloszláshoz. Ellenőrzésképpen, kiszámoltam a másik három árfolyam százalékos eloszlását a szórásaik szerint és hasonló eredményeket kaptam:

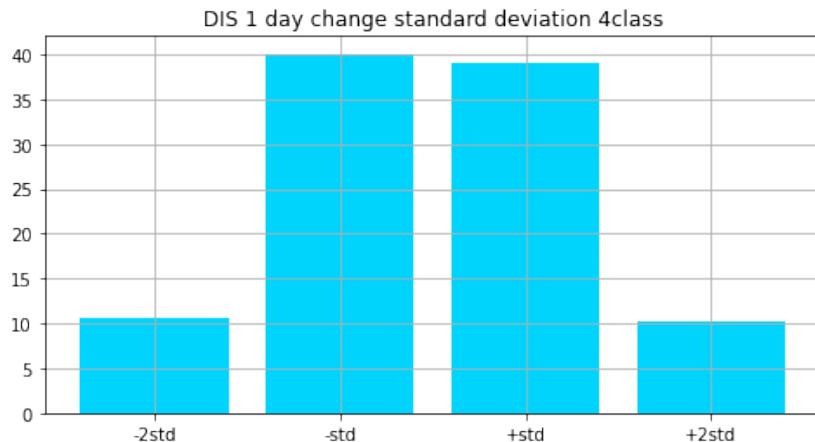
Név	átlag	szórás	-3std	-2std	-std	+std	+2std	+3std
<b>DIS</b>	0.036%	1.929%	2.32%	8.31%	40.09%	39.05%	7.66%	2.54%
<b>F</b>	0.034%	2.856%	1.66%	7.14%	43.25%	38.89%	7.00%	2.04%
<b>UBS</b>	0.026%	2.337%	2.53%	7.31%	39.55%	41.40%	6.81%	2.38%
<b>XOM</b>	0.033%	1.64%	2.61%	9.34%	37.33%	39.49%	8.84%	2.36%

**2.1. táblázat.** 6class eloszlás

Ezen eredmények alapján már fel tudok címkézni minden egyes napot egy diszkrétt halmazból választható tulajdonságok szerint, azaz, minden naphoz hozzárendelem az következő napból visszaszámolt változás szerinti besorolást.

### 2.6.3. Osztályozás javítása

Mivel a középső két osztálynak megfelelő elemek darabszáma nagyságrendekkel meghaladja a szélső csoportok számosságát, emiatt fennáll az a veszély, hogy a tanítás során túlságosan középsúlyos lesz a rendszer. Ennek kivédése érdekében érdemes minden két oldalon összevonni a két szélső csoportot és új osztályozást készíteni. A DIS esetében ez a következőképpen alakul:



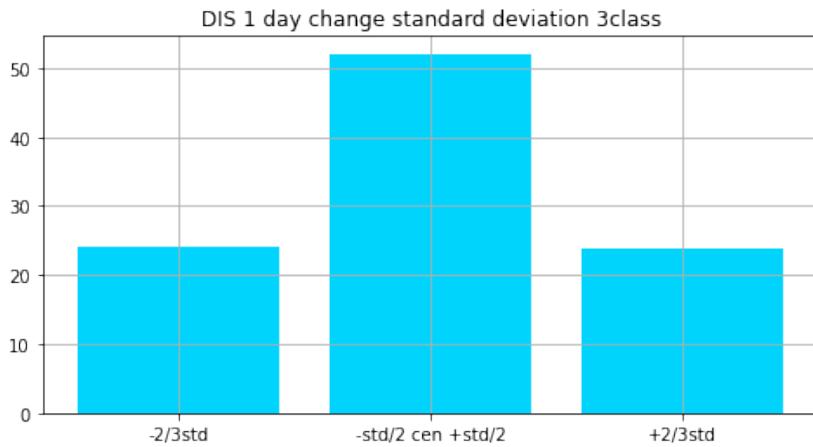
**2.8. ábra.** Az osztályozás javítása

Hasonlóan ehhez, újra számoltam a másik három árfolyam megfelelő osztályozását is:

Név	átlag	szórás	-2std	-std	+std	+2std
<b>DIS</b>	0.036	1.929	10.64%	40.09%	39.05%	10.22%
<b>F</b>	0.034	2.856	8.81%	43.25%	38.89%	9.02%
<b>UBS</b>	0.026	2.337	9.85%	39.55%	41.40%	9.19%
<b>XOM</b>	0.033	1.64	11.96%	37.33%	39.49%	11.19%

## 2.2. táblázat. 4class eloszlás

Ugyanakkor, mindenkorábbi osztályozás páros, azaz, a legsűrűbb terület az eloszlás szerint, ami az átlag körül van, ketté van vágva. Ezzel az a probléma, hogy a kis pozitív és negatív változásokat, amelyek az átlag közelében vannak, gyakorlatilag lehetetlen megkülönböztetni egymástól az előrejelzések szintjén. Ennek okán készítettem egy olyan eloszlási csoportosítást mely csak három részből áll: az átlagtól plusz-mínusz fél szórás, valamint az annál kisebb (balra) és az annál nagyobb (jobbra). A DIS esetében ez a következőképpen alakul:



**2.9. ábra.** Eloszlási csoportosítás

Hasonlóan korábbi csoportosításokhoz, ezt is újra számoltam a másik három árfolyam esetén is:

Név	átlag	szórás	$-2/3\text{std}$	centrum	$+2/3\text{std}$
<b>DIS</b>	0.036	1.929	24.03%	52.06%	23.87%
<b>F</b>	0.034	2.856	22.39%	55.80%	21.78%
<b>UBS</b>	0.026	2.337	22.61%	54.60%	22.75%
<b>XOM</b>	0.033	1.64	25.37%	48.96%	25.64%

**2.3. táblázat.** 3class eloszlás

## 2.7. Az adatsor tisztítása és transzformálása

### 2.7.1. SMA50

Az időszor adatok esetében két dolgot nagyon fontos figyelembe venni: a zajt és a környező adatok relevanciáját. Ez különösen igaz a részvényárfolyamoknál, ahol a pillanatnyi árfolyam állapotába bele van kódolva az azt megelőző periódus is. Ennek a két tényezőnek az együttes kezelésére én a vizsgált árfolyamok SMA50-es (50 day Simple Moving Average) mutatóját használtam. Ez a mutató, ugyanúgy diszkrét adatfolyam, az elemei az adott nappal bezárt 50-napos periódus átlagértéke. Ebből kifolyólag, minden egyes adatpontba bele lesz kódolva egy általános és közgazdasági szempontból is fontos kapcsolat az öt megelőző periódussal [M+21].

Annak bizonyítására, hogy ez a típusú adatsor a legmegfelelőbb a további kísérletezésre a zaj szempontjából, kiszámoltam a vizsgált részvények átlagos fehérzaj koefficiensét a 21 pontos ablakokra az Open adatsorra és az ebből számolt differenciált és SMA50-es sorozatra és a következő eredményeket kaptam:

Ez a vizsgálat kimutatta, hogy a leginkább zajmentes adatsor a SMA50, hiszen mind a négy esetben ez rendelkezik a legkisebb átlagos fehérzaj koefficienssel a 21 pontos ablakok esetén.

Név	Open	Differenciált	SMA50
<b>DIS</b>	0.028	-2.789	0.011
<b>F</b>	0.040	0.347	0.016
<b>UBS</b>	0.033	-550.389	0.012
<b>XOM</b>	0.023	-3.948	0.008

**2.4. táblázat.** Fehérzaj koeficiensek

### 2.7.2. Gramian-Angular transzformáció

A kutatás során az egyik vizsgált eszköz annak a kérdésnek a megválasztására, hogy korábbi rész szekvenciákból meg lehet-e játsolni egy előzetes kezendő nap változását a neurális háló volt. Ez az eszköz azért alkalmas erre a feladatra, mert a neurális hálókat tekinthetjük univerzális függvény approximátoroknak [HSW89], ami azt jelenti, hogy bármi, az adathalmazban felmerülő és rendszerezhető kapcsolatrendszer érzékelni tud, képes kódolni egy olyan mechanizmust, amely alapján egy új elem esetén is igen nagy megbízhatósággal megadja az annak megfelelő osztályt. Esetünkben ez azért fontos, mert a részvényárfolyamoknak, de általánoságban véve minden jelsorozat, amely nem szabályos, a következő eleme pontosan nem jelezhető előre. Legjobb esetben tudunk készíteni egy valószínűségi eloszlást, amely megad egy elméleti korlátot, de még ennek tudatában is lehetetlen elemről-elemre előre jelezni az, hogy hova esik a következő.

Ugyan akkor, a kutatás egy olyan logika szerint készült, hogy egy elemet az előtte levő 21-ből minél hatékonyabban lehessen előre jelezni. Ezen a ponton már logikusan következik, hogy ezeket az ablakokat, mint egymástól független elemek kell kezelní, transzformálni, a megfelelő címkét hozzájuk rendelni és elkezdeni velük a tanítást.

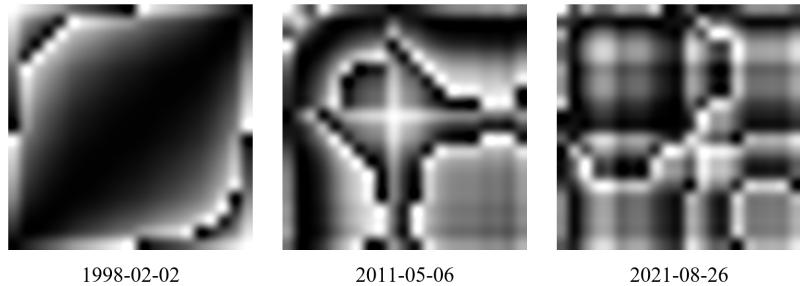
A neurális hálók egyik erősebb változata a mély tanulású neurális hálók. Ezek ereje abban rejlik, hogy a háló mélyebb szerkezettel rendelkezik és így képes sokkal absztraktabb kapcsolatokat is megtalálni a bemenet és kimenet között, azt pedig majd általánosan belekódolni az egész rendszerbe. Azonban, a mélytanulású hálók főként képekből tanulnak, mivel képesek azokon detektálni különböző jellegzetességeket, és végül ezeknek a jellegzetességeknek az összessége adja majd a besoroláshoz szükséges információt.

Mivel egy jelsorozat esetében rendelkezésre álló információ egy diszkrét számsorozat, ezért ebben a formában nem alkalmas egy mélytanulású háló tanítására. Ebből következik, hogy ezt a jelet valamilyen módon át kell alakítani képpé. Ezt a problémát oldja meg a spektrogram és skalogram használata.

Az első esetben, a tanulmányozott ablak spektrumát illetve az ablak eltolásának a spektrumát alakítjuk át képinformációvá, ahol az Ox tengely az eltolást, az Oy tengely a spektrumok frekvenciáját illetve az ezek által meghatározott kép pixel a spektrumok nagyságát kódolja. Hasonló technikát jelent a skalogram készítés, ahol a spektrogram analízis által használt Fourier helyet Wavelet transzformációt használunk, tipikusan ilyen alkalmazás a Morlet skalogram. Az így generált képek alkalmasak lehetnek akár emberi szem számára észrevehetetlen jelenségek kimutatására, ezek felismerését végzi majd a mély tanulású konvolúciós neuronháló. Mindennek az a lényege, hogy az elmúlt időszakban megtörténő előrejelzésekhez a megfelelő kép mintázatot tudunk rendelni. Miután a neuronháló megtanulja ezeket a mintázatokat, és az aktuális ablakban hasonló mintát

fedez fel, előre tudjuk jelezni a tőzsdei trendeket, de általánosan is jó módszer a véletlen sorozatok prediktálásában.

Az általam használt módszer a Gram szög transzformáció volt [T<sup>+20</sup>], amely részleges korrelációt képezi le az idősor egyes értékpárjai között. Két módszer áll rendelkezésre: a Gram szög összegző mező és a Gram szöökülönbség mező. Az általam használt verzió a szög összegző mező volt. Ennek segítségével fekete-fehér képeket generáltam, melyek közül néhány a következőképpen nézik ki a DIS esetén:



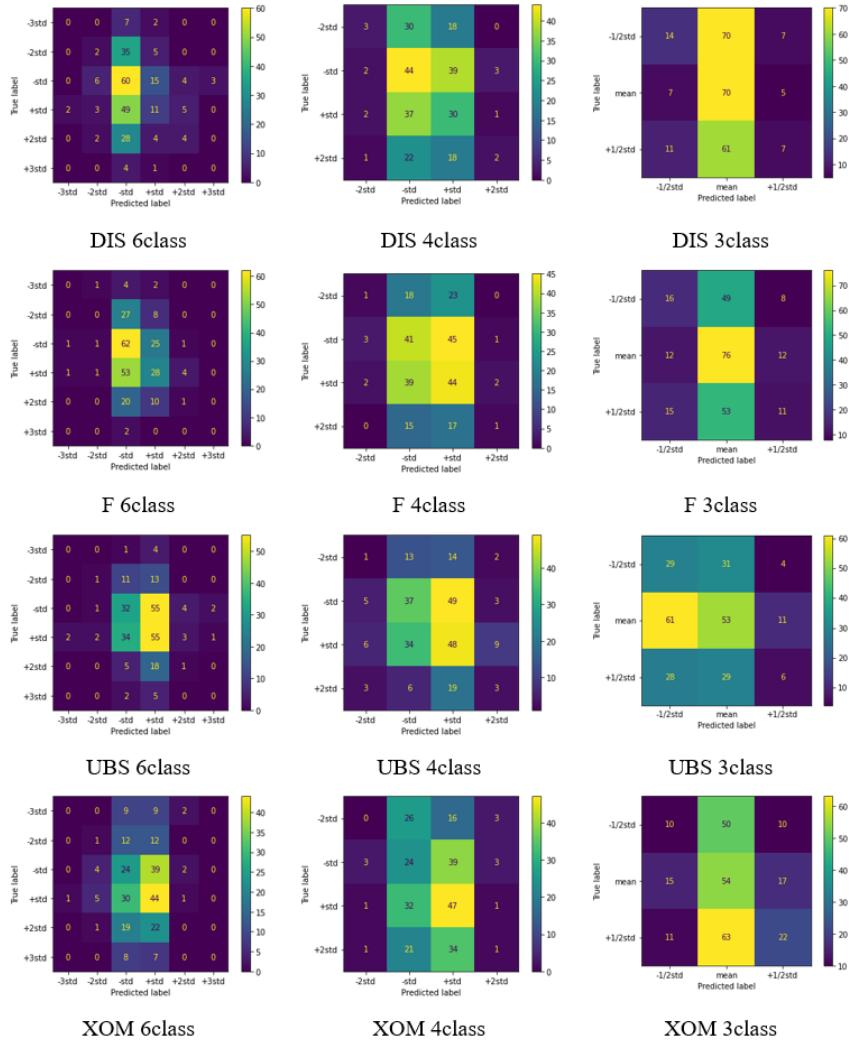
**2.10. ábra.** Gramian-Angular

## 2.8. Eredmények a neurális hálóval

Az eredményeket kétféleképpen lehet vizsgálni: a besorolás pontossága szerint és a pénzügyi hatékonyság szerint. Az első szempont szerint megint két metrikát érdemes megemlíteni: a konfúziós mátrixot és az átlagos távolságot. A második szempont szerint úgy lehet vizsgálni, hogy adott pénzügyi stratégia szerint milyen hozamokat produkál a rendszer a tesztelt perióduson végzett szimuláció során.

### 2.8.1. Besorolási pontosság

A besorolási pontosság első metrikáját, a konfúziós mátrixot, úgy kapom meg, hogy a tesztelt felosztású besorolás szerkezete szerint készítsek egy négyzetes mátrixot, ahol a sorok fogják azt jelenteni, hogy melyik kategóriát kellett volna előre jelezze a rendszer, az oszlopok fogják azt jelenteni, hogy melyiket számolta ki valójában. Értelemszerűen a fő átló fogja a tökéletes találatokat jelenteni. A vizsgált cégek mindenkor osztályozása szerint a következő eredmények születtek:



**2.11. ábra.** Neurális háló confusion mátrixai

A besorolási pontosság második metrikáját, az átlagos távolságot úgy kell értelmezni, hogy átlagosan mennyivel osztályozza mellé az adott elemeket. Ezt a következő képlet szerint számolom ki:

$$avgdist = \frac{1}{n} \cdot \sum_{i=1}^N \frac{1}{(1 + |pred_i - valós_i|)}$$

. Ennek az a lényege, hogy minden tökéletes előrejelzés egy pontot ér és egy előrejelzés minél inkább mellé megy, annál inkább csökken ez a pontszám. Ezeket pontszámokat összeadom, majd pedig átlagolom. Ez a metrika megpróbálja kifejezni az egész szimuláció nagyságrendi hatékonyságát egyetlen számban.

Emelet a metrika mellett, kiszámoltam még a rendszerek pontos találati számát a tanuló bemeneten, a teszt bemeneten, valamint csak a teszthalmaz esetén középső értékek pontos osztályozásának arányát és a következő értékeket kaptam százalékos bázis alapon:

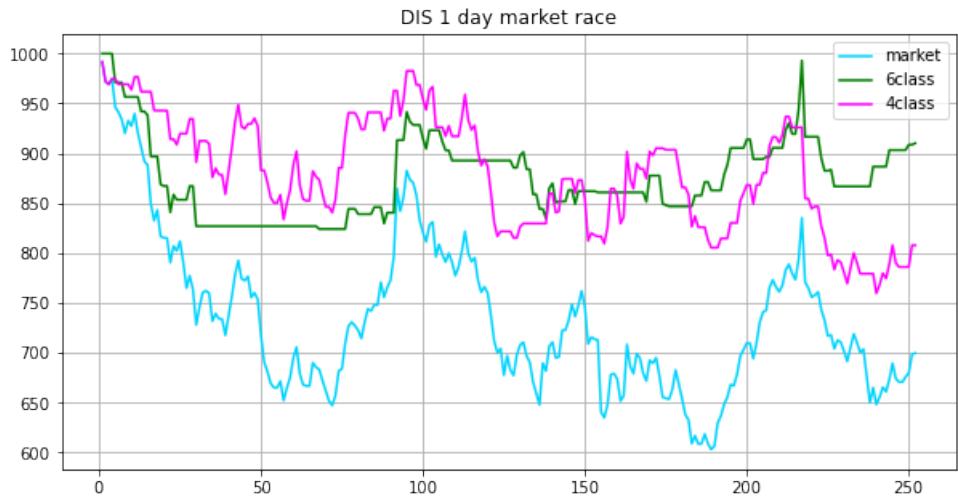
Eset	Tanító pontos	Teszt pontos	Átlagos távolság	Teszt közép
<b>DIS 6class</b>	61.05%	33.33%	62.41%	50.63%
<b>DIS 4class</b>	72.79%	25.00%	59.06%	36.07%
<b>DIS 3class</b>	76.25%	36.11%	66.86%	85.36%
<b>F 6class</b>	77.12%	36.11%	65.24%	50.84%
<b>F 4class</b>	75.625%	34.52%	64.55%	48.02%
<b>F 3class</b>	77.53%	40.87%	68.91%	76.00%
<b>UBS 6class</b>	75.54%	35.31%	64.61%	45.54%
<b>UBS 4class</b>	81.81%	35.31%	65.24%	44.50%
<b>UBS 3class</b>	67.99%	34.92%	65.34%	42.40%
<b>XOM 6class</b>	80.24%	27.38%	57.38%	45.33%
<b>XOM 4class</b>	69.16%	28.57%	61.17%	47.33%
<b>XOM 3class</b>	80.72%	34.12%	65.67%	62.79%

**2.5. táblázat.** Neurális háló alap statisztikái

### 2.8.2. Szimulációs tesztelés

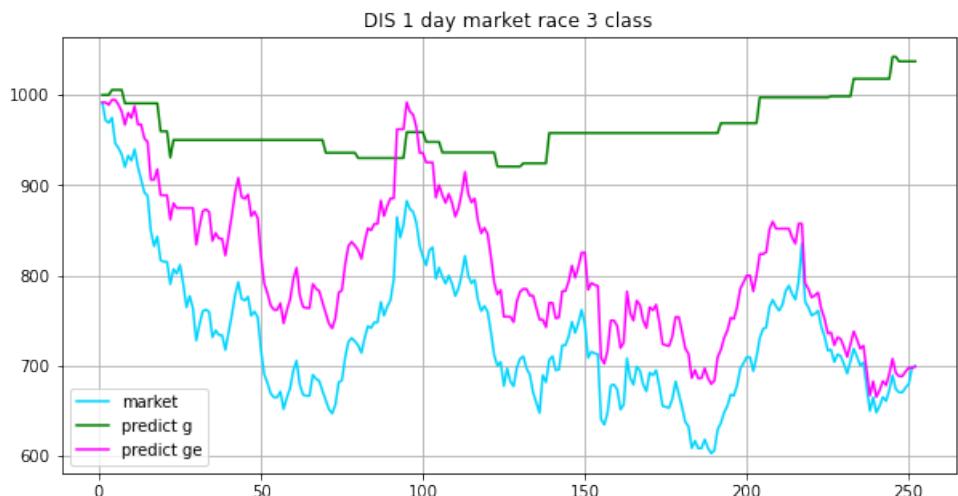
A szimulációs hatékonyság vizsgálata során egy versenyt imitálunk az adott cég valós árfolyamváltozása és az előrejelzések alapján végrehajtott lépések között. Itt két dolgot nagyon fontos megemlíteni: a módszer csak előrejelzéseket tud adni de a piacot befolyásolni nem tudja és nyerni csak akkor tudunk, ha a rendszer növekedést jelez előre és az valóban be is következik. Ezek alapján, a haszon így is-úgy is nagyban fog függeni attól, hogy a valóságban végül milyen mozgások következtek be. Ha egy árfolyam csökkenő trendben van, akkor valójában a veszteségeinket akarjuk minimalizálni, ha egy árfolyam növekvő trendben van, akkor pedig a nyereség maximalizálásra törekszünk.

Ezek alapján a következő stratégia szerint működik a rendszer: amennyiben pozitív előrejelzést kapunk, akkor tartjuk a pozíciókat/visszavásárlóunk, amennyiben negatív előrejelzést kapunk akkor kímaradunk/eladunk. A páros osztályozásoknál értelemszerűen a jobb oldali osztályok mind pozitívat jelentenek, a páratlan osztályozás szerint pedig le lehet szimulálni azt, ha a közepét is pozitívnak kezeljük, de azt is, ha csak a szigorúan jobb oldalt tekintjük annak. Ezzel a módszerrel gyakorlatilag a konfúziós mátrix True-Positive oldalát hasznosítjuk és alkalmazzuk elemenként. A szimuláció kiinduló értéke 1000, DIS esetében a következő versenyt jött létre:



**2.12. ábra.** DIS 6class és 4class szimuláció

Az ábrán a 6class osztályozás és 4class osztályozás versenye látható a piaccal szemben. Látható, hogy a piac komoly veszteséget könyvelt el a vizsgált időszakban, ennek révén nagyon nehéz volt nyereséget generálni ellenben könnyen nagy veszteséget lehetett volna szenvedni. Emiatt se a 6class módszer, se a 4class módszer sem tudott nyereséget termelni a kiinduláshoz képest, de minden esetben felültejesítették a piacot. Hasonlóan ehhez, elvégeztem a DIS páratlan osztályozás szimulációját minden két stratégia szerint és a következő verseny jött létre:



**2.13. ábra.** DIS 3class g és 3class ge szimuláció

Ebben az esetben a két módszer teljesítménye között komoly különbség alakult ki. A csak a jobb oldalra tevő (predict g) sokkal jobban teljesített a közepére is tevő módszerrel szemben (predict ge), hiszen még az első még a kiindulást is fölülteljesítette, addig utóbbi sokat vesztett.

A DIS-hez hasonlóan leteszteltem a rendszert a másik három árfolyamon és összegeztem az összes eredmény és piaci mozgással összevetve. A piac teljesítménye a következő volt:

DIS	-29.43%	F	-27.84%	UBS	+5.43%	XOM	+32.35%
-----	---------	---	---------	-----	--------	-----	---------

## 2.6. táblázat.

Cégek árfolyamváltozása tesztelési periódusban

A szimulációk a következőképpen alakultak:

Eset	Kezdethez	Markethez	Testvér módszerhez
<b>DIS 6class</b>	-8.98%	+30.06%	+10.23%
<b>DIS 4class</b>	-19.21%	+15.44%	-10.23%
<b>DIS 3class g</b>	+3.71%	+48.20%	+33.80%
<b>DIS 3class ge</b>	-30.08%	-0.09%	-33.80%
<b>F 6class</b>	+15.78%	+63.46%	+27.69%
<b>F 4class</b>	-11.09%	+24.36%	-27.69%
<b>F 3class g</b>	-1.00%	+39.76%	+17.72%
<b>F 3class ge</b>	-18.72%	+14.74%	-17.72%
<b>UBS 6class</b>	+20.19%	+13.12%	+0.01%
<b>UBS 4class</b>	+20.18%	+13.11%	+0.01%
<b>UBS 3class g</b>	+12.75%	+6.12%	+32.15%
<b>UBS 3class ge</b>	-19.39%	-24.14%	-32.15%
<b>XOM 6class</b>	+14.69%	-13.14%	-19.13%
<b>XOM 4class</b>	+33.83%	+1.35%	+19.13%
<b>XOM 3class g</b>	+26.85%	-3.93%	-4.59%
<b>XOM 3class ge</b>	+31.45%	-0.45%	+4.59%

## 2.7. táblázat.

Neurális háló szimulációs eredményei

A látottak alapján le lehet vonni azt a következtetést, hogy rendszer jól működik és a piachoz képest átlagosan jobban teljesít. Ezt bizonyítandó összehasonlítottam a vizsgált négy cégre adott szimulációk átlagát az SP500-as részvény indexel, amely egy átlagos indexe az Egyesült-Államok 500 legnagyobb nyilvános vállalatának:

Eset	Kezdethez	S&P500-hoz
<b>S&amp;P500</b>	-10.33	
<b>6class</b>	+10.42	+20.75
<b>4class</b>	+5.73	+16.06
<b>3class g</b>	+10.58	+16.73
<b>3class ge</b>	-9.19	-1.14

## 2.8. táblázat.

Neurális háló összevont teljesítménye

Az eredmények összegzése alapján a következő következtetéseket lehet levonni:

1. A kidolgozott módszer működik és teljesíti az elvártakat;

2. A módszerek közül összességében a 6class és 3class g módszerek működnek a legjobban, de ezek a leg szkeptikusabb stratégiák. Ez annyit tesz, hogy hosszú távú növekedési trendeknél nem annyira hatékonyak (pl. XOM-esetén);
3. A módszerek közül összességében a 3class ge. módszer működik a legrosszabbul, ennek viszont az az alapja, hogy a négy vizsgált részvényből kettő is súlyosan negatív évet zárt, és mivel ez a módszer követi le legszorosabban a piacot, ezért a sok esés ezt a stratégiát jobban bünteti, ellenben folyamatos növekedéskor ez a leghatékonyabb (pl. XOM-esetén);
4. A módszert leghatékonyabbá úgy lehet tenni hogy, ha előre tudjuk jelezni a hosszú távú piaci hangulatot, mert akkor kiválaszthatjuk az abban legjobban működő esetet és stratégiát.

## **2.9. A szövegelemzés módszere**

### **2.9.1. A felhasznált adatok**

A részvények árfolyamának változását a való életben, amint azt már korábban is megemlíttettem, rengeteg különböző külső behatás befolyásolja. Ezek nagyrésze a közéleti, gazdasági valamint világpolitikai történések. A cégek részvényeire ráhatással van azok direkt kommunikációja vagy a velük kapcsolatos közvetlen híresztelések, mint például a pénzügyi jelentések, marketing kampányok vagy új termékek-szolgáltatások bejelentése, de ezeknek a nagyságrendi mértéke elenyésző a világot meghatározó folyamatos történésekhez képest. Ráadásul, legtöbbször maguk a világi események váltanak ki reakciókat a cégek kommunikációjában vagy saját politikájában, emiatt ha az eseményfolyamok oldaláról akarjuk vizsgálni és előre jelezni a változásokat, akkor érdemes a gyökerénél kezelní a témát.

Mivel gyakorlatilag a legjelentősebb eseményeket és a rájuk adott reakciók mintázatát akarjuk vizsgálni, ezért lényegében olyan forrásra van szükség, amely azokat tömören és tárgyilagoson kezeli, és lehetőleg már meg is szűrte azokat a jelentőségük szerint. Ennek a kritériumrendszernek felel meg a Wikipédia évenkénti összesítője. Ezek a bejegyzések rövid, tömör és objektív stílusban foglalják össze az évek legfontosabb eseményeit.

The screenshot shows a computer browser displaying the Wikipedia page for the year 2001. The left sidebar contains a navigation menu with links to various categories like Population, Conflicts, Culture, Economy, Politics, Religion, Science and technology, Events, Nobel Prizes, References, Further reading, and External links. The main content area is titled "Events [edit]" and lists significant events from January 1 to January 29, 2001. Some of the listed events include: Greece becoming the 12th country to join the Eurozone, Ghana undergoing its first peaceful transfer of power, Apple launching iTunes, AOL and Time Warner merging, a train collision in Congo killing at least 30 people, a 7.6-magnitude earthquake in El Salvador, the assassination of Laurent-Désiré Kabila, George W. Bush becoming the 43rd president of the United States, impeachment proceedings against Philippine President Joseph Estrada, the Second EDSA Revolution, the final documented case of the 2000-2001 Uganda ebola outbreak, the Taba Summit between Israel and the Palestinian National Authority, the 2001 insurgency in Macedonia, a Falun Gong self-immolation incident in Beijing, a 7.7 magnitude earthquake in Gujarat, India, and corruption scandals surrounding Indonesian President Abdurrahman Wahid. To the right of the text, there is a small map titled "Map of the 2001 insurgency in Macedonia".

## 2.14. ábra. Wikipédia évösszegző bejegyzés

Miután összegyűjtöttem az elmúlt nagyjából 25 év történéseit (1998.01.06 és 2023.03.26 között), ezeket az adatokat regularizáltam és megtisztítottam, felvetődött a kérdés, hogy miként lehet őket hasznosítani. Mivel ezek csoportosításánál a címkét szintén az fogja adni, hogy az események után milyen változás következett be, valamint továbbra is az egy hónapos ablakkal dolgozok, azért a következőképpen jártam el: minden címkére meghatároztam azt az egy hónapos időintervallumot, ami vele záródik be, majd összegyűjtöttem az összes eseményt, ami ebben a periódusban következett be és végül összekapcsoltam őket.

### 2.9.2. A megfelelő algoritmus kiválasztása

Miután elvégeztem a szövegek feldolgozását és összekapcsolását a megfelelő mérési pontokkal, újfent létrehoztam a különböző kategorizálási szempontok szerinti eseteket. A következő lépés az volt, hogy kiválasszam a megfelelő osztályozó algoritmust, amelyre három lehetséges jelölt volt: MultinomialNB, SGDClassifier és LogisticRegression.

A MultinomialNB, angolul Naive Bayes classifier for multinomial models, egy olyan osztályozó algoritmus, amely alkalmas diszkrét jellemzőkkel történő csoportosításra pl. szószám a szöveg osztályozáshoz. A multinomiális eloszlás általában egész számokat igényel, a gyakorlatban azonban a töredékszámítások, például a kifejezés gyakorisága – inverz dokumentum gyakoriság is működhetnek [S+19].

Az SGDClassifier, angolul Stochastic Gradient Descent classifier, egy becslő algoritmus amely szabályos lineáris modellekkel valósít meg sztochasztikus gradiens süllyedés (SGD) tanulással: a veszteség gradiensét minden mintán egy időben becsülik meg, és a modellt minden közbén frissítik egy csökkenő szilárdsági ütemezéssel, más néven tanulási sebességgel [K+15].

A LogisticRegression, angolul Logistic Regression, egy statisztikai modell (más néven logit modell), melyet gyakran használnak osztályozásra és prediktív elemzésre. A logisztikai regresszió egy esemény bekövetkezésének valószínűségét becsüli meg az igen-nem tengelyen (esetünkben a nőt vagy nem nőt az árfolyam) független változók adott adatkészlete alapján [S+20].

Ezek után letesztem mind a négy vizsgált céget, mind a három osztályozás szerint, és a következő eredményeket kaptam a tanító halmazon végzett kísérletek után:

Eset	MultinomialNB	SGDClassifier	LogisticRegression
<b>DIS 6class</b>	49.22%	50.82%	59.78%
<b>DIS 4class</b>	49.85%	51.95%	61.26%
<b>DIS 3class</b>	54.07%	56.05%	64.06%
<b>F 6class</b>	49.76%	51.43%	62.15%
<b>F 4class</b>	50.21%	51.79%	62.26%
<b>F 3class</b>	55.46%	58.81%	66.66%
<b>UBS 6class</b>	50.00%	51.77%	60.73%
<b>UBS 4class</b>	50.99%	53.05%	62.41%
<b>UBS 3class</b>	55.88%	57.86%	65.42%
<b>XOM 6class</b>	48.37%	50.18%	58.60%
<b>XOM 4class</b>	49.06%	50.83%	59.96%
<b>XOM 3class</b>	51.66%	53.66%	63.33%

**2.9. táblázat.** Hírelemzési módszerek teljes statisztikája

Ezek után összesítettem az eredményeket:

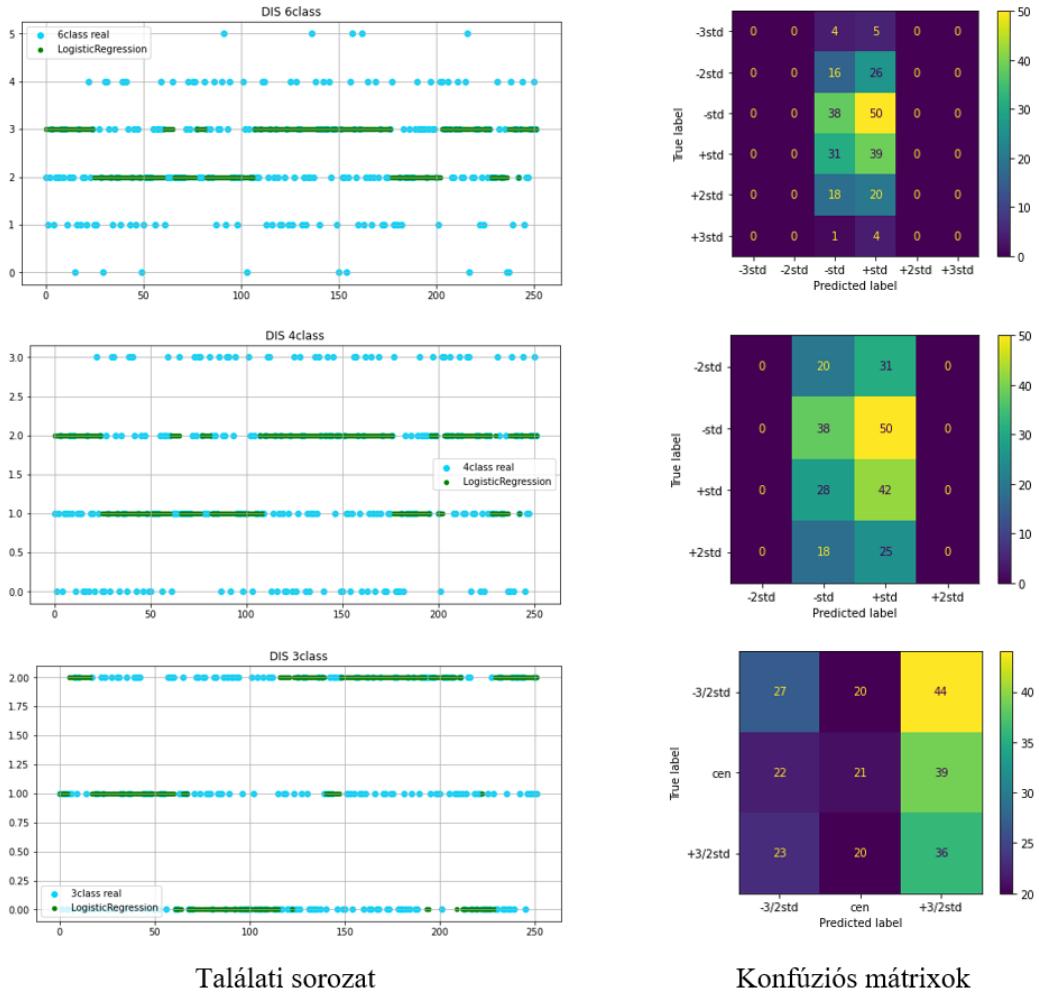
Eset	MultinomialNB	SGDClassifier	LogisticRegression
<b>6class</b>	49.33%	51.05%	60.31%
<b>4class</b>	50.02%	51.90%	61.47%
<b>3class</b>	54.26%	56.59%	64.86%
<b>Globális</b>	51.21%	53.18%	62.21%

**2.10. táblázat.** Hírelemzési módszerek összegzett statisztikája

Ezek alapján kiderül, hogy ezen felfogás szerint csoportosítva a hírfolyamokat, a LogisticRegression algoritmus messze felül teljesíti a másik két jelöltet. Következtetésekén ezt a módszer érdemes a leginkább vizsgálni.

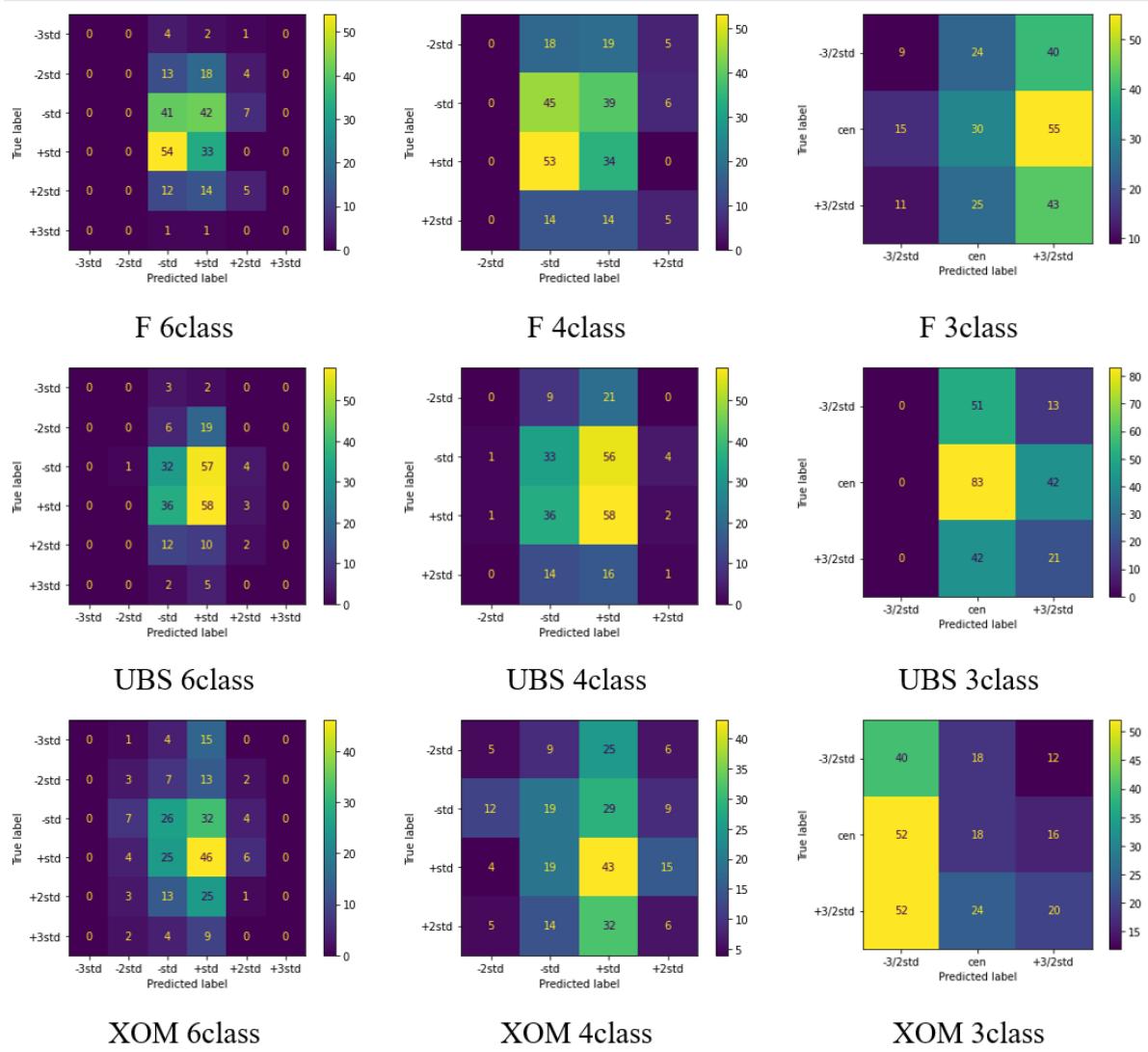
### 2.9.3. A LogisticRegression további vizsgálata

Tekintve, hogy ennek a módszernek eddig csak a tanító halmazon mutatott eredményei voltak elemezve, ezért szükséges letesztni, hogy miként teljesít a tesztelő halmazon is. Ezek alapján megnéztem, hogy milyen előrejelzéseket ad a valósághoz képest és milyen a konfúziós mátrixa:



**2.15. ábra.** Hírelemzés DIS találati sorozat és konfúziós mátrix

Az ábrák alapján már rögtön látszik, hogy a páros osztályok esetén gyakorlatilag csak a két középső esetre igazodik, míg a class3 esetén igen nagy a szórás. A konfúziós mátrixokat kiszámítottam a másik három cég esetén is, és a következő eredményeket kaptam:



**2.16. ábra.** Hírelemzés további konfúziós mátrixok

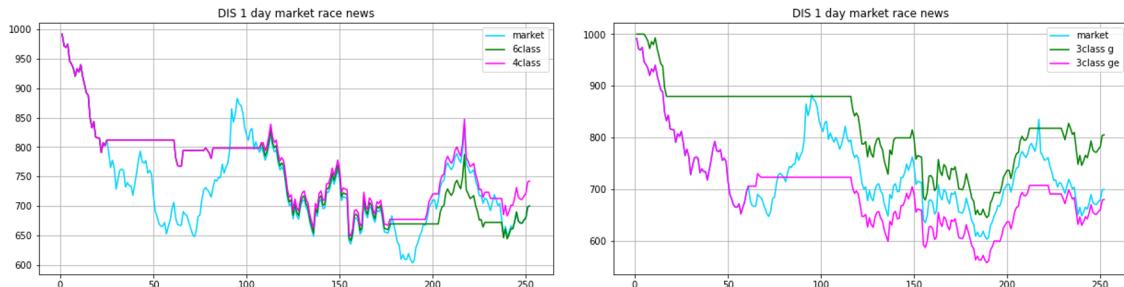
Ezek alapján már észrevehető, hogy ez a módszer sokkal nagyobb hibával dolgozik. A másik három cég esetén is megmaradt a páros osztályok középre torlódása és a class3 nagy szórása. Ezt számszerűsítendő, kiszámoltam az esetek pontos statisztikai mutatóit:

Eset	Teszt pontos	Átlagos távolság	Teszt közép
<b>DIS 6class</b>	30.55%	61.24%	48.73%
<b>DIS 4class</b>	31.74%	62.63%	50.63%
<b>DIS 3class</b>	33.3%	62.23%	25.60%
<b>F 6class</b>	31.34%	62.08%	41.80%
<b>F 4class</b>	33.33%	63.59%	44.63%
<b>F 3class</b>	32.53%	62.69%	30%
<b>UBS 6class</b>	36.50%	65.01%	47.12%
<b>UBS 4class</b>	36.50%	65.60%	47.64%
<b>UBS 3class</b>	41.26%	69.77%	66.40%
<b>XOM 6class</b>	30.15%	59.35%	48.00%
<b>XOM 4class</b>	28.96%	59.95%	41.33%
<b>XOM 3class</b>	30.95%	61.24%	20.93%

**2.11. táblázat.** Hírelemzés statisztikai mutatók

#### 2.9.4. Szimulációs hatékonyság

Hasonlóan a neurális háló piaci hatékonyságának teszteléséhez, ezen a módszeren is végrehajtottam a piaccal szembeni verseny szimulációját. DIS esetén a következő grafikon rajzolódott ki:



**2.17. ábra.** Hírelemzés DIS szimuláció

A páros osztályok esetén 6class gyakorlatilag a piaccal megegyezően teljesít, míg a 4class megveri azt, de mindenketten veszteséggel zártak. A páratlan osztály esetén sincs nyereség és itt is érvényesül az, hogy a ge-reverzió sokkal sebezhetőbb a folyamatos piaci eséssel szemben. Elkészítettem a teljes szimulációt az összes vizsgált cégre:

Eset	Kezdethez	Markethez	Testvér módszerhez
<b>DIS 6class</b>	-30.04%	-0.03%	-4.23%
<b>DIS 4class</b>	-25.81%	+6.01%	+4.23%
<b>DIS 3class g</b>	-19.47%	+15.06%	+12.52%
<b>DIS 3class ge</b>	-32.0%	-2.83%	-12.52%
<b>F 6class</b>	-29.87%	-0.99%	+4.55%
<b>F 4class</b>	-25.31%	+5.43%	-4.55%
<b>F 3class g</b>	-2.62%	+37.47%	+17.89%
<b>F 3class ge</b>	-20.51%	+12.20%	-17.89%
<b>UBS 6class</b>	-7.75%	-13.18%	-0.48%
<b>UBS 4class</b>	-7.27%	-12.72%	+0.48%
<b>UBS 3class g</b>	+1.09%	-4.85%	+31.70%
<b>UBS 3class ge</b>	-19.39%	-24.14%	-31.70%
<b>XOM 6class</b>	+14.91%	-12.97%	-11.41%
<b>XOM 4class</b>	+26.33%	-4.32%	+11.41%
<b>XOM 3class g</b>	+4.34%	-20.97%	-15.44%
<b>XOM 3class ge</b>	+19.79%	-9.28%	+15.44%

**2.12. táblázat.** Hírelemzés szimulációs eredményei

Ezek után megismételtem az S&P500-al való összehasonlítást:

Eset	Kezdethez	S&P500-hoz
<b>S&amp;P500</b>	-10.33%	
<b>6class</b>	-13.18%	-2.86%
<b>4class</b>	-8.01%	+2.31%
<b>3class g</b>	-4.16%	+6.16%
<b>3class ge</b>	-13.02%	-2.70%

**2.13. táblázat.** Hírelemzés összevont teljesítménye

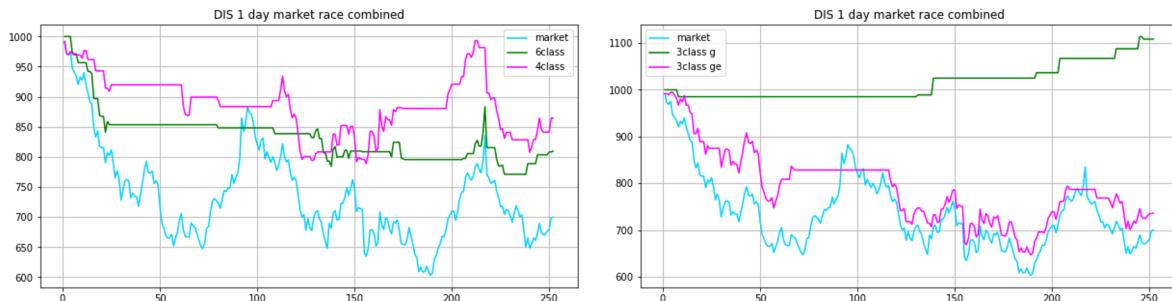
1. Ez a módszer is képes felül teljesíteni a piacot, hiszen van olyan iteráció amely összességében jobban szerepel (4class és 3class g);
2. Ugyanakkor, minden egyes összegző esetben veszteséggel zárta a szimulációt a kiinduláshoz képest. Ez egyszerre a rendszer gyengesége, de azt sem szabad elfelejteni, hogy a négy vizsgált részvényből kettő is erősen veszteséges volt (DIS, F);
3. A mélytanulású háló esetén megállapított 3. következtetés itt is érvényesül;
4. Ezt a módszer valószínűleg úgy lehetne hatékonyabbá tenni, ha sokkal nagyobb visszamenőleges ablakkból építeném fel az egyes elemekhez rendelt információs bázist, hiszen akkor sokkal nagyobb merítésből lehetne keresni a kulcsmintázatoknak megfelelő eseményeket.

## 2.10. A két módszer kombinálása és összehasonlítása

A kombinált módszer során nincs értelme újraszámolni pontossági metrikákat, mivel az gyakorlatilag a minimuma a két korábbi módszer soros kombinációjának, azonban a szimulációs vizsgálatot érdemes elvégezni.

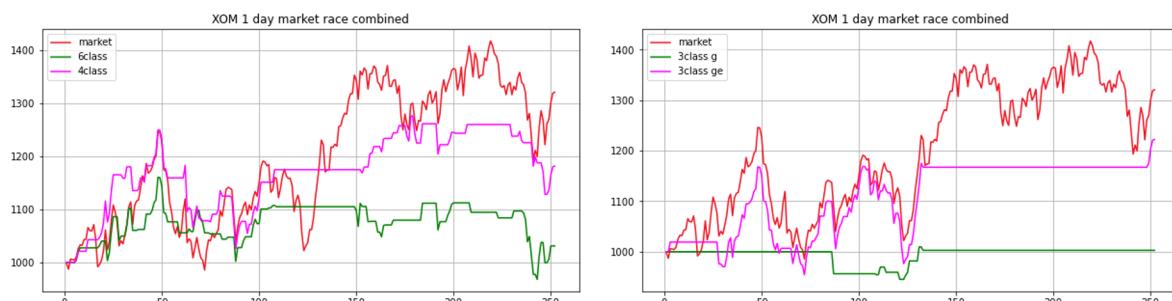
### 2.10.1. A kombinált szimuláció

A kombinált szimuláció módszerét hasonlóan alkalmazzuk, mint eddig, azaz csak akkor tartjuk a részvényeket/vásárolunk be, ha pozitív és ennek ellenettje, ha negatív az előrejelzés, de ezúttal, csak azt tekintjük pozitív előrejelzések, ha minden előrejelzés ezt mondja. Ezek alapján a DIS –en végrehajtott kísérlet a következőképpen alakul:



2.18. ábra. Kombinált módszer DIS szimuláció

Észrevehetjük, hogy az a módszer egy csökkenő tendenciájú piac esetén jobban szerepel, mint az előző, hiszen a kettős megerősítés miatt még inkább védi magát az esésektől. Emiatt a jelenség miatt érdemes közelebbről megvizsgálni, hogy miként viselkedik egy növekvő piaccal szemben például XOM-esetén:



2.19. ábra. Kombinált módszer XOM szimuláció

Ebben az esetben azt látjuk, hogy nem áll fenn a teljes blokkolás esete, hiszen sok esetben követi le jól a növekedést. Következik a teljes szimuláció a kombinált módszerrel:

Eset	Kezdethez	Markethez	Testvér módszerhez
<b>DIS 6class</b>	-19.03%	+15.68%	-5.47%
<b>DIS 4class</b>	-13.5%	+23.50%	+5.47%
<b>DIS 3class g</b>	+10.83%	+58.38%	+37.20%
<b>DIS 3class ge</b>	-26.37%	+5.20%	-37.20%
<b>F 6class</b>	+4.27%	+47.20%	+26.07%
<b>F 4class</b>	-21.80%	+10.38%	-26.07%
<b>F 3class g</b>	-0.37%	+40.64%	+3.74%
<b>F 3class ge</b>	-4.11%	+35.36%	-3.74%
<b>UBS 6class</b>	+9.91%	+3.44%	+2.20%
<b>UBS 4class</b>	+7.70%	+1.36%	-2.20%
<b>UBS 3class g</b>	+1.09%	-4.85%	+20.48%
<b>UBS 3class ge</b>	-19.39%	-24.14%	-20.48%
<b>XOM 6class</b>	+3.08%	-21.93%	-15.05%
<b>XOM 4class</b>	+18.14%	-10.53%	+15.05%
<b>XOM 3class g</b>	+0.26%	-24.06%	-21.96%
<b>XOM 3class ge</b>	+22.22%	-7.43%	+21.96%

**2.14. táblázat.** Kombinált módszer szimulációs eredményei

Szintén az S&P500-al való összehasonlítással vizsgálom az általános teljesítményt:

Eset	Kezdethez	S&P500-hoz
<b>S&amp;P500</b>	-10.33	
<b>6class</b>	-0.44%	+9.87%
<b>4class</b>	-2.36%	+7.95%
<b>3class g</b>	+2.95%	+13.27%
<b>3class ge</b>	-6.91%	+3.40%

**2.15. táblázat.** Kombinált módszer összevont teljesítménye

Az eredmények összegzése alapján a következő következtetéseket lehet levonni:

1. A korábbi módszerek érzékenysége és erőssége a tendenciákkal szemben továbbra is fennáll, de a feleadatot maximálisan teljesíti;
2. Ez a módszer lényegesen kevesebb interakciót enged megvalósulni bármelyik korábbi módszerhez képes, főként a kifejezetten változékony időszakokban (XOM 3class eset szimulációjának második fele);
3. Lévén, hogy ez a módszer már a kombinációja két teljes értékűnek, minden bizonnyal ebben rejlik a legtöbb optimalizációs potenciál akár újabb adatok nélkül, pusztán csak abból kiindulva, hogy a stratégia szabályán is lehet változtatni és/vagy a két egyéni vizsgálat válaszát súlyozni.

## 2.11. A kutatás eredményeinek és tapasztalatainak az összegzése

### 2.11.1. Melyik módszer a legjobb?

Annak a kérdésnek a megválaszolására, hogy általánosságában véve melyik eset melyik módszerrel kombinálva a leghatékonyabb, összehasonlítottam a szimuláció kezdetéhez vett összes átfogó teljesítményt és meghatároztam a globálisan legjobb verziót:

Eset	Neurális háló	Hírelemzés	Kombinált	Esetek átlaga
6class	+10.42%	-13.18%	-0.44%	-0.11%
4class	+5.73%	-8.01%	-2.36%	-3.27%
3class g	+10.58%	-4.16%	+2.95%	+1.84%
3class ge	-9.19%	-13.02%	-6.91%	-9.70%

**2.16. táblázat.** A kutatás összegzett eredményei

Ezek alapján tarol a csak a neurális hálóra alapozó módszer, hiszen a legjobb három eredmény is ennek segítségével lett elérve. Pusztán az esetek szempontjából a 3class g teljesített a legjobban, nem mellesleg erre az esetre alapszik a legjobb összesített eredmény is.

### 2.11.2. Hipotézisek vizsgálata

**H00:** Neurális hálók által generált előrejelzések képesek a szabad piaci mozgás felett teljesíteni.

Ez a hipotézis teljességgel beigazolódott, hiszen ez a módszer külön-külön is, de összesítve az SP500-al szemben is jobban teljesít. Az egyetlen gyengesége az irreálisan nagyon növő indexekkel szemben van (pl. XOM), ám bizonyos iterációk hibahatáron belül vannak még ahhoz képest is, hiszen olyan közeli eredményeket produkálnak, hogy lehetséges egy másik vagy esetleg hosszabb tanítási fázis után optimálisabb háló alakulna ki.

**H01:** A történelmi hírfolyamok elemzése során generált előrejelzések képesek a szabad piaci mozgás felett teljesíteni.

Ez a hipotézis is igazoltnak tekinthető, bár az eredmény nagyon szoros. Bizonyos iterációja nagyban felülteljesíti a piacot, mások viszont kikapnak tőle. A végső kijelentést az alapján lehet megtenni, hogy a négy eset átlagos teljesítménye az SP500-al szemben pozitív (0.7275%), de annyira kicsi, hogy felmerül a hibahatár okozta kétésség.

**H02:** A fent vizsgált két módszer kombinációja felülteljesíti minden részmódszert.

Ez a hipotézis megdölt, hiszen annak ellenére, hogy jól teljesít a kombinált módszer és a piacot mind a négy eset szimulálása esetén megveri, ami egyedülálló teljesítmény, csak a második, a hírelemzésre alapuló módszerrel szemben teljesít jobban.

### 2.11.3. További következtetések

A módszerek további javításában óriási potenciál van. Rengeteg paraméter megváltoztatásával lehet kísérletezni, kezdve onnan, hogy nagyobb ablakkal dolgozunk, az újragondolt stratégiákon keresztül, egészen a bővített és sűrűbb híradathalmazig. Kutatásom során több komoly korlátba is ütköztem. Először is igen komoly számítási kapacitás béli

limitációkkal küzdöttem, hiszen egy 21x21-es méretű generál kép feldolgozása is 441-darab bemeneti értéket jelent a neurális hálónak, de például egy lehetséges következő ablak, a negyedéves periódus-után generált 63x63-as kép már 3969- darab bemeneti értéket kell feldolgozni. Ez, egy általam is használt többrétegű neurális háló esetén exponenciális számítási igény növekedést jelent.

Egy másik limitáció maga az idő volt. Akár a hírelemzés, akár maga a képgenerálási módszer esetén sok más algoritmus létezik, de még az általam megvizsgált lehetőségek közül sem tudtam minden végig vinni és az összes kombinációt letesztelni. Itt szintén jelentkezik a hirtelen szélesedés veszélye.

#### 2.11.4. Összegzés

A kutatás legfontosabb eredménye az, hogy lehetséges olyan általános módszert tervezni, amely hatékonyan tud előre jelezni véletlen sorozatokat, de speciel a részvényelemzés terén, a másodlagos információs halmazok hatékony feldolgozása jó alapot jelenthet egy abszolút teljesítő rendszer létrehozására.

[A dolgozat kódja elérhető itt.](#)

# Ábrák jegyzéke

1.1.	A program szerkezete . . . . .	17
1.2.	A felhasználói felület szerkezete . . . . .	19
1.3.	Második oldal . . . . .	21
1.4.	Második oldal . . . . .	22
1.5.	Bővített a megjelenítés . . . . .	23
1.6.	Hisztogram . . . . .	23
1.7.	Harang modell . . . . .	24
1.8.	Mozgóátlag keresztek . . . . .	25
1.9.	Polinomiális regresszió . . . . .	26
1.10.	RBF közelítés . . . . .	27
1.11.	Lokális valószínűségi tér . . . . .	28
2.1.	Walt Disney Company . . . . .	33
2.2.	Ford Motor Company . . . . .	33
2.3.	UBS Group AG . . . . .	34
2.4.	Exxon Mobil Corporation . . . . .	34
2.5.	Az árfolyamok korrelációja . . . . .	36
2.6.	A DIS árfolyam . . . . .	37
2.7.	A DIS hisztogram . . . . .	38
2.8.	Az osztályozás javítása . . . . .	39
2.9.	Eloszlási csoportosítás . . . . .	40
2.10.	Gramian-Angular . . . . .	42
2.11.	Neurális háló confusion mátrixai . . . . .	43
2.12.	DIS 6class és 4class szimuláció . . . . .	45
2.13.	DIS 3class g és 3class ge szimuláció . . . . .	45
2.14.	Wikipédia évösszegző bejegyzés . . . . .	48
2.15.	Hírelemzés DIS találati sorozat és konfúziós mátrix . . . . .	50
2.16.	Hírelemzés további konfúziós mátrixok . . . . .	51
2.17.	Hírelemzés DIS szimuláció . . . . .	52
2.18.	Kombinált módszer DIS szimuláció . . . . .	54
2.19.	Kombinált módszer XOM szimuláció . . . . .	54

# Táblázatok jegyzéke

2.1.	6class eloszlás	38
2.2.	4class eloszlás	39
2.3.	3class eloszlás	40
2.4.	Fehérzaj koeficiensek	41
2.5.	Neurális háló alap statisztikái	44
2.6.	Cégek árfolyamváltozása tesztelési periódusban	46
2.7.	Neurális háló szimulációs eredményei	46
2.8.	Neurális háló összevont teljesítménye	46
2.9.	Hírelemzési módszerek teljes statisztikája	49
2.10.	Hírelemzési módszerek összegzett statisztikája	49
2.11.	Hírelemzés statisztikai mutatók	52
2.12.	Hírelemzés szimulációs eredményei	53
2.13.	Hírelemzés összevont teljesítménye	53
2.14.	Kombinált módszer szimulációs eredményei	55
2.15.	Kombinált módszer összevont teljesítménye	55
2.16.	A kutatás összegzett eredményei	56

# Irodalomjegyzék

- [AB97] Jeffrey S. Abarbanell and Brian J. Bushee. Fundamental analysis, future earnings, and stock prices. *Journal of accounting research*, 35(1):1–24, 1997.
- [Bou96] Paul Bourke. Cross correlation. In *Cross Correlation", Auto Correlation—2D Pattern Identification*, 1996.
- [C<sup>+</sup>09] Israel Cohen et al. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.
- [Gen96] Ramazan Gencay. Non-linear prediction of security returns with moving average rules. *Journal of Forecasting*, 15(3):165–174, 1996.
- [HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [K<sup>+</sup>15] Fasihul Kabir et al. Bangla text document categorization using stochastic gradient descent (sgd) classifier. In *2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, pages 1–4. IEEE, 2015.
- [L<sup>+</sup>77] Barbara Liskov et al. Abstraction mechanisms in clu. *Communications of the ACM*, 20(8):564–576, 1977.
- [M<sup>+</sup>21] Massoud Metghalchi et al. Trading rules and excess returns: evidence from turkey. *International Journal of Islamic and Middle Eastern Finance and Management*, 2021.
- [Mil78] Robin Milner. A theory of type polymorphism in programming. *Journal of computer and system sciences*, 17(3):348–375, 1978.
- [PI07] Cheol-Ho Park and Scott H Irwin. What do we know about the profitability of technical analysis? *Journal of Economic surveys*, 21(4):786–826, 2007.
- [PJ16] Nada Petrusheva and Igor Jordanoski. Comparative analysis between the fundamental and technical analysis of stocks. *Journal of Process Management and New Technologies*, 4(2):26–31, 2016.
- [S<sup>+</sup>19] Gurinder Singh et al. Comparison between multinomial and bernoulli naïve bayes for text classification. In *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pages 593–596. IEEE, 2019.

- [S<sup>+</sup>20] Kanish Shah et al. A comparative analysis of logistic regression, random forest and knn models for text classification. *Augmented Human Research*, 5:1–16, 2020.
- [SYM84] John G Saw, Mark CK Yang, and Tse Chin Mo. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.
- [T<sup>+</sup>20] K Palani Thanaraj et al. Implementation of deep neural networks to classify eeg signals using gramian angular summation field for epilepsy diagnosis. *arXiv preprint arXiv:2003.04534*, 2020.