

BUDAPESTI GAZDASÁGI EGYETEM

PÉNZÜGYI ÉS SZÁMVITELI KAR

SZAKDOLGOZAT

Juhász Tamás

Nappali tagozat

Gazdaságinformatikus

Pénzügyi informatikus

2021

BUDAPESTI GAZDASÁGI EGYETEM

PÉNZÜGYI ÉS SZÁMVITELI KAR

Értékpapírok értékeinek előrejelzése
algoritmusokkal

Belső konzulens: Dr. Fauszt Tibor

Külső konzulens: Rékasi László

Juhász Tamás

Nappali tagozat

Gazdaságinformatikus

Pénzügyi informatikus

NYILATKOZAT

Alulírott Juhász Tamás büntetőjogi felelősségem tudatában nyilatkozom, hogy a szakdolgozatomban foglalt tények és adatok a valóságnak megfelelnek, és az abban leírtak a saját, önálló munkám eredményei.

A szakdolgozatban felhasznált adatokat a szerzői jogvédelem figyelembevételével alkalmaztam.

Ezen szakdolgozat semmilyen része nem került felhasználásra korábban oktatási intézmény más képzésén diplomaszerzés során.

Tudomásul veszem, hogy a szakdolgozatomat az intézmény plágiumellenőrzésnek veti alá.

Budapest, 2021 év november hónap 21 nap

Juhász

hallgató aláírása

Tartalomjegyzék

1.	Bevezetés.....	5
1.1.	Értékpapírok és típusai.	5
1.2.	Befektetési lehetőségek	7
2.	Történeti és szakirodalmi áttekintés	8
2.1.	Vállalatok pénzügyi analízise	9
2.1.1.	Pénzügyi helyzet elemzése	9
2.1.2.	Vagyoni helyzet elemzése	10
2.1.3.	Jövedelmezőségi helyzet elemzése	11
2.1.4.	Hatékonysági helyzet elemzés	12
2.1.5.	Részvények főbb mutatószámai és következtetések	12
2.2.	Mesterséges intelligencia alapok	13
2.2.1.	Mesterséges intelligencia történelmi háttere	15
2.2.2.	Elemzéshez használt MI algoritmusok.....	16
2.2.2.1.	Lineáris Regresszió	17
2.2.2.2.	Mesterséges neurális hálózatok.....	17
3.	Vállalatok pénzügyi elemzése	18
3.1.	Vállalati adatok automatizált kinyerése.....	20
3.2.	Adatok Feldolgozása	25
3.3.	Adatok elemzése.....	26
4.	Részvény árfolyam előrejelzés mesterséges intelligencia programokkal	30
4.1.	ARIMA modell.....	30
4.1.1.	ARIMA eredmények	39
4.2.	RNN modell.....	40
4.2.1.	RNN eredmények	46

4.3. Részvény elemzék	47
4.3.1. S&P500 index	47
4.4. Tesla.....	49
4.5. Apple.....	50
5. Összegzés	51
Irodalomjegyzék.....	53
Ábrák jegyzéke.....	57

1. BEVEZETÉS

Manapság, aki kicsit is érdeklődik a befektetések, megtakarítások iránt, egy nagyon nehéz helyzetben találja magát. Azt már mindenki tudja, hogy a megtakarított pénzünket be kell fektetni, egyrészt mert az elinflálódik, másrészt pedig mindenki azt szeretné, hogy a pénze érte dolgozzon. Engem is bevezettek ezek a mondatok és el is kezdtem kutatni, keresni, hogy hogyan is kell befektetni a pénzem. Nagyon hamar rájöttem, hogy rengeteg az információ fórumokon, weblapokon és a nagy részük mind csak a felszínen kapargat. Én már lassan fél éve kutatom azt, hogy mibe is fektessek, de még mindig nem találtam meg a megfelelő portfóliót. Időközben arra is rájöttem, hogy mindenkinek más a megfelelő portfólió - valaki szereti a kockázatot, mások biztonsági játékosok, valaki 3 hónap alatt akar meggazdagodni, más meg nyugdíjas éveit szeretné a Bahamákon tölteni.

Rengeteg befektetési lehetőség van, szinte mindent felfoghatunk befektetésként, akár a tanulását is vagy egy új bicikli vásárlását is. Különböző, szerintem a tanulás és a magunkba fektetett energia, idő és pénz azok, amik a legjobban megtérülnek, de akkor így nagyon rövid lenne ez a szakdolgozat. Ebben a szakdolgozatban csak az értékpapír befektetésekről lesz szó és szakdolgozatom végére megpróbálok egy olyan programot írni, amely segít nekünk kiválasztani az értékpapírokat. A program megírásához az előző évi tantárgyaim sarkaltak, főként a Számítógépes adatelemzés, a Pénzügyi és számviteli informatika, és a statisztikai tantárgyak motiváltak. Ezek a tantárgyak során megtanultam, hogy a programok és kimutatások hogyan segítik a stratégia és operatív döntések meghozatalát egy cég életében. Az itt megszerzett tudást szeretném alkalmazni a saját életemben is, hiszen nekem is nap mint nap rengeteg döntést kell meghoznom, és ezek a programok az én döntéseim meghozatalában is segíthetnek, csak egy meg megfelelő rendszert kell kialakítani ehhez. Azt is megtanultam ezeken a tantárgyakon, hogy a probléma megoldása egy jó kérdés feltevésével kezdődik. Az én kérdésem itt az volt, hogy mibe fektessem be a megtakarított pénzem?

1.1. ÉRTÉKPAPÍROK ÉS TÍPUSAI.

Az értékpapír, mind jogilag, mind gazdaságilag eléggé tág fogalom. Jogi értelemben véve „meghatározott kellékekkel rendelkező okirat, a benne tanúsított jog kizárólagos eszköze, a papír nélkül sem érvényesíteni, sem bizonyítani, sem átruházni nem lehet”. (www.oktato.econ.unideb.hu, 2021). Közgazdasági értelemben pedig „vagyon jogot

megtestesítő, forgalomképes okirat, számlán megjelenő összeg vagy elektronikus jel.” (www.oktato.econ.unideb.hu-2, 2021). Értékpapírokat rengeteg szempontból lehet csoportosítani. Számunkra egy fontos csoportosítás az értékpapír megjelenési formája:

- Fizikai: szó szerint megfogható, papír alapú.
- Dematerizált: ez a fajta értékpapír csak nyilvántartásokban létezik, elektronikus úton létre jött és terjesztett értékpapír. Ebben az esetben mindig kell, hogy létezzen egy elektronikus számla, amit legtöbbször egy bróker cég üzemeltet, mellé különböző más szolgáltatásokat is nyújt, amik lehetővé teszik, hogy tőzsdén kereskedhessünk.

Munkám során én csak is dematerizált papírokkal fogok foglalkozni. Egy másik fontos csoportosítás számunkra a foglalt jogszerinti:

- Hitelviszonyt testesít meg (kötvény, váltó...). Innen csak a kötvényekkel fogunk foglalkozni pár szó erejéig.
- Részesedést vagy tagsági jogokat testesít meg (részvény...). A részvény lesz a fő profil ebben a dolgozatban.
- Árura szóló tulajdonjogot testesít meg (közraktárjegy...). Ennek a kereskedése kiemelt kockázattal jár, úgyhogy én ezt kihagynám a portfóliómból, hiszen még nincs meg a megfelelő gyakorlatom. (www.oktato.econ.unideb.hu-3, 2021)

Megemlítenék a számomra még egy fontos fogalmat az ETF-et. Az ETF - az Exchange Traded Fund szavakból származik, jelentése Tőzsdén Kereskedett Alap. Viselkedése nagyon hasonlít a részvényhez, ugyanúgy lehet adni, venni és a tulajdonságai is hasonlóak. Kezdő befektetőknek nagyon ajánlott, mivel az ETF-et úgyis felfoghatjuk, mint egy kosár, amibe már bele vannak dobálva a részvények, kötvények, árucikkek. Legnépszerűbb ETF-ek általában valamilyen tőzsdei indexet követnek, azaz ugyanazokat a részvényeket tartalmazzák ugyanolyan fajsúllyal, mint maga az index, ezért is szeretik a kezdő befektetők, hiszen ezzel a portfóliójuk diverzifikációja is meg van oldva. Az utóbbi években nagyon elterjedtek, mert a jól kiválasztott ETF-ek könnyedén felülmúlják az aktívan kezelt befektetési alapok hozamát is és a költségük pedig eltörpül mellettük. Két nagy csoportra lehet bontani az ETF-et - osztalékfizető és újrabefektető. Ez azért fontos, mert pl. ha veszünk egy olyan ETF-et, ami tartalmaz osztalékfizető részvényt, akkor az osztalék e szerint kerül elszámolásra. A magyar adózási törvényeket nézve érdekesebb az osztalékot újrabefektető ETF-ben keresni, hiszen így

nem kell még az osztalék után is adózni, és ezáltal rengeteg munkától kímélhetjük meg magunkat. (Szendrei, 2021)

1.2. BEFEKTETÉSI LEHETŐSÉGEK

Mielőtt fejest ugranánk a mély vízbe, nézzük meg, milyen értékpapír befektetési lehetőségei vannak egy átlagembernek. Amelyek szinte mindenkinek eszébe jutnak, azok a magyar államkötvények, a Magyar Állampapír Plusz, Prémium Magyar Állampapír és az új Prémium Euró Magyar Állampapír. Hozamban ez a három a legmagasabb és mind a 3 középtávú befektetésnek mondható, hiszen lejáratuk mindegyiknek 5 év, a kamata is mindegyiknek 3% és 6% között mozog. Mi a baj ezekkel? Az, hogy a jelenlegi infláció mértéke



1. Ábra: „Infláció alakulása”

megeszi mindegyiknek a hozamát. Illetve a prémium állampapírok infláció fölött vannak, de 1,25%-kal, ami nem túl jó, mivel, ha az inflációs környezet alacsony, akkor a hozamunk is alacsony lesz. Tehát ha inflációval korrigált hozamot nézünk, akkor szinte semmit se nyertünk a befektetésünkön. Ha a mostani inflációs adatokat nézzük, akkor láthatjuk, hogy kezd stagnálni, de májusban elérte az 5,1% is, ami meghaladja az összes kötvény hozamát, de az éves átlag is valószínűleg 4% felett lesz. Tehát nekünk mindenképpen olyan éves hozamot kell választanunk, ami bőven 6-7% fölött van, ha nyerni is akarunk valamit. (www.allampapir.hu, 2021) Ekkor viszont már a tőzsdére kell mennünk, de erre jól fel kell készülni, mivel a kockázatok is jóval magasabbak itt. Plusz, aki most 2021 vége fele akar befektetni, az nehéz

helyzetben van, mivel minden tőzsdei index a csúcson van, nagyon meg kell fontolnunk a döntéseinket, ezért is fogjuk elemezni, hogy mit várhatunk ezektől az indexektől a jövőben.

Terítéken lesz valamilyen S&P500 ETF, ami az Amerika legértékesebb 500 részvényét tartalmazza. Illetve az egy másik fontos index ETF-e a MSCI (Morgan Stanley Capital International), ami globálisan 1583 vállalat részvényét foglalja magába. Legnagyobb súlyban (65%) itt is amerikai részvények vannak, ezután jönnek más fejlett országok cégei. Az MSCI indexet negyedévente számolják újra a vállalatok piaci kapitalizációjuk alapján. Jól látható, hogy az amerikai részvények sokat tesznek a latba, ezért egyes részvényeket külön le fogok elemezni. Így a részvények bestsellerét az Apple és az Amazont mindenképpen leelemezem, illetve a Tesla részvény alakulására is kíváncsi vagyok.

2. TÖRTÉNETI ÉS SZAKIRODALMI ÁTTEKINTÉS

Mielőtt rátérek arra, hogy milyen módszerekkel fogom az elemzést végezni, szeretném felvázolni, pénzügyi elemzés és a mesterséges intelligencia alapjait. Első körben az elemzést két részre bontottam:

- Pénzügyi analízis.
- Előrejelzés mesterséges intelligencia algoritmussokkal.

Ezekhez a feladatokhoz válogattam meg a szükséges eszközöket és ezek alapján szeretném a szakirodalmi áttekintést is felosztani. Az első feladathoz egyszerű algoritmusokat fogunk használni a vállalat pénzügyi mutatóinak a kiszámolásához. Ahhoz, hogy értsük mit jelentenek majd a megkapott eredmények, szeretnék ebben a fejezetben egy rövid belátást adni, hogy milyen pénzügyi mutatószámok vannak, amik segíthetnek az értékpapír értékének az előrejelzésben. Mivel ezeket a papírokat mindig egy adott vállalat adja ki, azért a vállalatnak az értéke nagyban befolyásolja az adott értékpapír értékét is. Bár ez nem teljesen egyértelmű, hiszen ezek statikus pénzügyi mutatók a múlt eredményeit mutatják be. Jó példa erre a Tesla, hiszen ott nem a vállalat piaci értéke határozta meg a részvény értéket, hanem egyszerűen népszerű lett a befektetők között. Egy időben mindenki megörült a Tesla részvényért, túlértékelődött, a szakzsargon erre azt mondja, hogy egy lufi lett belőle, ami egy idő után kipukkant. Így történt az, hogy egy hónap leforgása alatt egy részvény ára 895\$-ról 563\$-ra esett, azaz -37% zuhant a részvény értéke. Ahhoz, hogy az ilyesfajta beleseteket elkerüljük, tudnunk kell a céget értékelni, hogy tudjuk mikor van az adott részvény túlárzva, illetve mikor van alulértékelve, hogy a megfelelő időpontban tudjunk beszállni az adott részvénybe. A

tőzsdéről még annyit kell tudni, hogy folyamatosan bika és medve piacok váltják egymást. Bikapiacnál az árak felfelé mennek, medvepiacnál pedig az árak esnek. Érdemes tehát a piac viszontagságait is figyelni. Befektetni akkor kell, amikor medve piac van, így egységnyi pénzért több részvényt vehetünk. Ezután pedig megvárjuk a bikapiac végét és akkor adjuk el amikor az árak az egekben vannak.

2.1. VÁLLALATOK PÉNZÜGYI ANALÍZISE

A vállalatok értékelést szinte minden elemző a múltbéli elért teljesítmény alapján kezdi meg. Az elemzést a vállalatok beszámolójából, mérlegéből kinyert adatokból végzik, és legtöbbször valamilyen arányszámot származtatnak belőlük. Ezekből az arányszámokból következtetéseket lehet levonni a vállalat:

- Pénzügyi
- Vagyoni
- Jövedelmezőségi
- Hatékonysági helyzeteiről.

Ha ezeket a mutatószámokat egy adott évre nézzük, akkor nem látjuk a köztük lévő összefüggéseket, ezért mindenképp legalább 3 évre visszamenőleg számoljuk ki. Így látni fogjuk azt, hogy milyen változásokon ment keresztül a vállalat és milyen tendenciák vannak a vállalat életében. Az még a jó ezekben a mutatókban, hogy gyakran ki vannak számolva elemzői oldalakon, így web scraping programokkal könnyedén hozzájuk férhetünk, ezekre később térek ki. (Takács-1, 2015)

2.1.1. PÉNZÜGYI HELYZET ELEMZÉSE

Ezek a mutatószámok a vállalkozás adósságállományára hosszú, illetve rövid lejáratú fizető- és hitelképességére, likviditására ad betekintést.

Adósságállomány – itt az egy évet meghaladó hosszú lejáratú kötelezettségeket értjük.

Értéke: a Hátrasorolt kötelezettségek + Hosszú lejáratú kötelezettségek összege lesz.

Adósságállomány aránya – mint a neve elárulja, az adósságállomány és a saját tőke viszonyát fejezi ki. Ez azt jelenti, hogy a vállalat milyen mértékben finanszíroz hitelből, illetve saját tőkéből.

Értéke: $\text{Adósságállomány} / (\text{Adósságállomány} + \text{Saját tőke})$.

Saját tőke aránya – a mutató azt vizsgálja, hogy hogyan aránylik a saját tőke az adósságállományhoz.

Értéke: Saját tőke / (Saját tőke + Adósságállomány).

Adósságállomány fedezettsége – azt kapjuk meg, hogy a saját tőke hogyan fedezi az adósságállományt.

Értéke: Saját tőke / Adósságállomány.

Likviditási mutató (current ratio) – azt fejezi ki, hogy a vállalat éven belül képes-e eleget tenni a rövidlejáratú kötelezettségeinek. A mutatótól mindenképpen elvárhatjuk, hogy 100% fölötti értéket produkáljon, hiszen forgóeszközöknek mindenképpen fedeznie kell a rövid lejáratú kötelezettségeket.

Értéke: Forgóeszközök / Rövid lejáratú kötelezettségek

Likviditási gyorsráta (quick ratio) - ez egy szigorúbb mutató a Likviditási rátánál, itt a forgóeszközökből kivonjuk a készleteket, hiszen ezeket időigényes pénzzé tenni.

Értéke: (Forgóeszközök–Készletek) / Rövid lejáratú kötelezettségek. (Takács-2, 2015), (Bíró – Kresalek – Pucsek – Sztanó-1, 2016)

2.1.2. VAGYONI HELYZET ELEMZÉSE

A vagyoni helyzet elemzése a vállalkozás eszköz, forrás szerkezetére ad betekintést, ezeket a mutatókat mérleg adataiból származtatjuk. Legfontosabb mutatószámok a következők:

Befektetett eszközök aránya – azt vizsgálja, hogy hogyan aránylanak a befektetett eszközök az összes eszközhöz.

Értéke: Befektetett eszközök / Összes eszköz.

Forgóeszközök aránya – azt vizsgálja, hogy hogyan aránylanak a forgóeszközök az összes eszközökhöz.

Értéke: Forgóeszközök / Összes eszköz.

Saját tőke növekedési mutató – azt vizsgálja, hogy hogyan nő a saját tőke a jegyzett tőkehez viszonyítva.

Értéke: Saját tőke / Jegyzett tőke.

Tőkeellátottság – a saját tőke arányát mutatja be az összes forráshoz viszonyítva.

Értéke: Saját tőke / Források összesen.

Céltartalékok aránya – fontos mutató lehet számunkra, hiszen láthatjuk, hogy hogyan alakul a céltartalékok aránya az összes forráshoz.

Értéke: Céltartalékok / Források összesen. (Takács-3, 2015), (Bíró – Kresalek – Pucsek – Sztanó-2, 2016)

2.1.3. JÖVEDELMEZŐSÉGI HELYZET ELEMZÉSE

Ez egy fontos pillére az elemzésnek, hiszem itt információt kapunk a bevételekről, ráfordításokról, a költségekről és a hozamokról. Főként egy nagyobb időhorizonton vett eredmények fontosak, így egymáshoz tudjuk viszonyítani, hogy hogyan változott/fejlődött a cég.

Árbevétel-arányos üzemi eredmény – azt fejezi ki, hogy az árbevételének hány százaléka jelent meg profit formájában. Fontos, hogy ebből levonjuk a költségeket, de még a pénzügyi tevékenység eredményét nem számoljuk hozzá.

Értéke: Üzemi (üzleti) eredmény / Árbevétel

Árbevétel-arányos adózás előtti eredmény – itt már figyelembe vesszük a pénzügyi tevékenységek eredményét is (nagy eltérés lehet a két eredmény között).

Értéke: Adózás előtti eredmény / Árbevétel.

Eszközarányos megtérülés (ROA) – arra szolgál, hogy az eszközök értékére számoljuk ki a jövedelmezőséget.

Értéke: Adózott eredmény / Eszközök összesen.

Saját tőke-arányos megtérülés (ROE) - ez a mutató azt vizsgálja, hogy hogyan alakul a jövedelmezőség a saját tőkéhez viszonyítva. A ROA és ROE mutatók egymáshoz viszonyíthatók, mivel jól szemléltetik, hogy hogyan alakul a finanszírozási mód - saját tőkéből vagy hitelből finanszíroz inkább a cég.

Értéke: Adózott eredmény / Saját tőke. (Musinszki,2017), (Takács-4, 2015).
(www.szamvitelez.hu/pont.hu, 2021)

2.1.4. HATÉKONYSÁGI HELYZET ELEMZÉS

Úgyis tekinthetjük, hogy ezek a mutatószámok magyarázzák a vagyoni helyzet alakulását, ugyanis itt a hozamot vetítik le a felhasznált erőforrásra.

Tárgyi eszközök hatékonysága – ez azt mutatja, hogy egy egységnyi lekötött eszköz hány egységnyi árbevételt termelt az adott időszak során.

Értéke: Árbevétel / Tárgyi eszközök.

Munkaerő hatékonysága – azt fejezi ki, hogy 1 főre vagy 1 egységnyi jövedelem ráfordításra jutó árbevételt mutassa meg, azaz egy fő mennyi értéket termelt a vállalatnak.

Értéke: Árbevétel / Létszám v. személyi jellegű ráfordítás

Készletek forgási sebessége – arra a kérdésre válaszol, hogy milyen szintű készlet szükséges az adott eredmény elérésére, minél gyorsabb a forgási sebesség, annál jobban gazdálkodnak a készletekkel.

Értéke: Éves nettó árbevétel / Éves átlagos készletszint. (Takács-5, 2021)

2.1.5. RÉSZVÉNYEK FŐBB MUTATÓSZÁMAI ÉS KÖVETKEZTETÉSEK

Szeretnék meg kitérni pár mutatószámra, amik az értékpapír értékelésében töltenek be fontos szerepet.

EPS – ezzel a mutatóval azt vizsgálhatjuk meg, hogy mennyi az egy részvényre eső adózott nyereség. Ha az értéke évről évre stabil vagy emelkedő, akkor az egy pozitív jel lehet számunkra.

Értéke: Adózott nyereség / Részvények száma.

P/E – ez egy nagyon fontos mutató: azt mondja meg, hogy részvény árához képest hogyan alakul a vállalat nyeresége. Arra használható, hogy képet kapjunk a részvény árazásáról, viszont itt érdemes szektoron belül valamilyen konkurenshez vagy szektor átlagához viszonyítani a megkapott eredményt.

Értéke: részvény ár / EPS. (www.iocharts.io, 2021)

Kapitalizáció – a részvény aktuális árfolyam szorozva a kibocsájtott részvények számával. Segítségével a vállalat teljes piaci értékét kapjuk meg.

Mint látható rengeteg információ származtatható a múltbéli gazdálkodás átvizsgálatával, bár fontos kiemelni ismét, hogy önmagában nem elegendő a vállalat pénzügyi helyzetének az átvizsgálása. A részvény piacot nem mindig érdeklik ezek a mutatók, viszont segíthetnek, főként a hosszútávú döntések meghozatalában. Kihat a vállalat megítélésére, hogyha folyamatosan hiteket vesz fel, és például abszolút nem likvid, akkor kérdéses, hogy kisebb megbotlás esetén, képes-e osztalékot fizetni. Még mindig nem ez lenne a legnagyobb probléma, hanem ennek a híre egyszerűen olyan nagy pánikot szokott kelteni, hogy egy hónap leforgása



2. Ábra: „Hírek hatása az árfolyamokra”

alatt szinte mindenki megszabadul az adott vállalat részvényétől, és emiatt a részvény árfolyam elkezd zuhanni. Sajnos nem minden hírt lehet előre jelezni, erre egy jó példa a BitCoin vs Elon Musk incidens. Musk azt írta twitteren, hogy a Bitcon-t többé nem fogadja el fizetési eszközként környezetvédelmi aggályok miatt. Tehát pár twitt hatására 63 ezer dollárról 29 ezer dollárra esett az árfolyam, mint a 2. ábra is mutatja. Ezt senki se tudta volna előre jelezni. Szerencsére ilyen katasztrófák a részvény piacokon ritkán történnek meg, hiszen itt a cég pénzügyi helyzete árulkodó lehet.

2.2. MESTERSÉGES INTELLIGENCIA ALAPOK

Manapság a mesterséges intelligencia (MI) egy széleskörben használt technológia, szinte nincs olyan ipar, aminek valamilyen szegmensébe ne lehetne használni. A dolgozatomban is fogok MI algoritmusokat használni, segítségükkel megpróbálom előre jelezni az árfolyamok mozgását. Sajnos a népszerűségének köszönhető az is, hogy sokan félre definiálják, keverik gépi tanulással vagy épp magát a technológiát túlmagasztalják. De az is előfordul, hogy valaki

attól tart, hogy mesterséges intelligencia leigázza majd az emberiséget. Öntsünk akkor tiszta vizet a pohárba és definiáljuk, hogy mi is az MI:

„Az MI kifejezés gépek mögött futó programokra utal, amelyek segítségével a gépek absztrakt, kreatív és deduktív emberi gondolkodásból származó döntéseket tudnak imitálni. Az MI a számítógépek digitális és bináris logikáján keresztül megvalósított matematikai-informatikai módszerekkel dolgozik. Ezek alapja a statisztika, vagyis a nagy adathalmazokban fellelhető összefüggések, anomáliák és tipikus mintázatok megkeresése” (Rajczy,2019)

Ez a fogalom nagyon jól körülírja, hogy mi is az MI és hogyan működik. A mesterséges intelligencia algoritmusok kategorizálására rengeteg mód létezik, az egyik legnépszerűbb a következő:

- **Reaktív gépek:** ez az egyik legegyszerűbb MI, hiszen mint a neve is elárulja, környezetére tud reagálni, egy inputra ad egy választ. Működésébe nem építi bele a múlt tapasztalatait, csak az előre definiált szabályok mentén hozza meg mindig a legjobb döntését. Jó példa erre a Deep Blue az IBM által fejlesztett MI, amit megtanítottak sakkozni és legyőzte az akkori sakk világbajnokot, vagy pl. a Google által fejlesztett AlphaGo.
- **Korlátozott memória:** ez a fajta algoritmus képes a múlt tapasztalatait, adatait beépíteni a döntéseibe. Ezeket az adatokat, tapasztalatokat korlátozott ideig tárolja, innen is ered a neve. A legáltalánosabb példa erre az önvezető autó, pl.: tárolja azt, hogy mennyi a megengedett sebesség, de ha észlel egy új táblát, ami felülírja ezt, akkor az előzőt már elfelejti és az aktuális sebességi korlátnak megfelelően fog viselkedni. Elemzésekhez ilyen algoritmusokat fogok fejleszteni, hiszen az elmúlt évek tőzsdei adataiból előrejelzéseket is csinálhatunk.
- **Az elme elmélete:** ez már egy nagyon fejlett fajtája az MI-nek, ami képesnek kell hogy legyen megérteni az emberi érzelmeket és ezek szerint kell hogy viselkedjen és kommunikáljon az emberekkel. Ez a fajta MI még fejlesztés alatt áll, de az egyik legkiemelkedőbb humanoid a Hanson Robotics által fejlesztett Sophia, aki egyben Szaúd Arabiai állampolgár is. Aki érdeklődik a robotok iránt, annak javaslom, hogy nézzon pár videót Sophiáról, nagyon érdekes ahogy viccelődik, és kommunikál az emberekkel, pár éve Budapesten is járt.
- **Önismeret:** jelenleg ez az MI még nincs kifejlesztve. Ettől az MI-től azt várjuk, hogy képes legyen önmaga döntésekre, önálló gondolatokra, érzelmekre. A tudósok a

szuperintelligencia létrejöttét 2050-re saccolják. Elon Musk és Bill Gates is tart a kifejlesztésétől: úgy gondolják, hogy veszélyeztetheti az emberi létet is, hiszen gondolkodhat úgy is, hogy emberek ártalmasak vagyunk, pusztítjuk a saját környezetüket, és egy digitalizált világban egy MI-nek minden eszköze meglesz az emberek ellen. Had jegyezzem meg, hogy viszont Mark Zuckerberg pozitívan látja a jövőt és úgy gondolja, hogy emberek és droidok közösen fognak élni. (Reynoso, 2019)

2.2.1. MESTERSÉGES INTELLIGENCIA TÖRTÉNELMI HÁTTERE

Szeretnék kitérni még egy pillanatra, hogy hogyan is jutottunk el ide, hogy egyes emberek félnek a MI kifejlesztésétől, szeretnék rávilágítani, hogy milyen momentumok vezettek ide.

Számomra hatalmas meglepetés volt, amikor láttam, hogy a Turing tesztet, gépi tanulás fogalmát, Alan Turing már 1950-ben bevezette, MI alapjait már 1943-ben Warren McCulloch és Walter Pitts lerakta, hiszen művükben „A Logical Calculus of the Ideas Immanent in Nervous Activity” -ben leírták, hogyha megfelelően van kiépítve egy neurális hálózat, akkor a rendszer képes a tanulásra. Viszont a MI atyjának mégis John McCarthy-t tartják, hiszen az Artificial Intelligence fogalmat ő vezette be 1956-ban. Ezek a dátumok azért voltak számomra hihetetlenek, mert az első teljesen elektronikus számítógépet csak 1939-ben fejlesztették ki. Ebből is látható a két technológia erősen összefonódik és kéz a kézben járnak. (Russel-Norvig, 2005)

1980-ban már ipari felhasználásra is került MI a Digital Equipment Corporationnél, ahol az R1 program számítógépes rendszerek megrendeléseit konfigurálta és 1986-ra évi 40 millió dollárt spórolt a cégnek. Bár az igazán nagy robbanást 2009-ben a Google által megalkotott önvezető autó és a 2011-ben bemutatott IBM Watson jelentette. Már csak a Watson megérne egy külön dolgozatot, hiszen rengeteg szegmense van, így például IBM Watson for Oncology - egy olyan MI program, ami a rák gyógyításban vállal hatalmas szerepet és képes egy kezelési javaslatot felállítani az összes világon elérhető kutatási eredmények és a beteg kórtörténete alapján. Nem a Watson az egyetlen ilyen program, ami az orvoslásban hatalmas szerepet kap, hiszen a Konvolúciós Neurális Hálózatok (CNN) a fotókról 95% sikerrel tudja megállapítani a melanómát, míg 17 ország 58 orvosa mindösszesen 86,6%-ban volt képes erre. (www.webbeteg.hu, 2021)(Hegyeshalami, 2018). Manapság pedig már széles körben elterjedt az MI, szinte mindenhol ott van: YouTube ajánló rendszerétől kezdve, képfelismerés programok (pl.: facebook feltöltött képnél felajánlja kiket jelöljünk meg) vagy a digitális

Assistensek, akik képesek a beszéd felismerésre és ez alapján parancsokat tudnak teljesíteni. Ilyen az Apple Siri is, ami már 2011 óta működik, de azóta már rengeteg Sirihez hasonló program van - például a Samsung Bixby vagy az Amazon Alexa. Az adatelemzésben is egyre elterjedtebb, gondoljunk csak az SAP HANA-ra, ami valós időben képes elemezni a játékosok teljesítményét, ennek segítségével tudta a német válogatott megnyerni a 2014-es VB-t. (Russel-Norvig, 2005) (www.computerworld.hu, 2015)

2.2.2. ELEMZÉSHEZ HASZNÁLT MI ALGORITMUSOK

Dolgozatom során több MI algoritmust is használni fogok, szeretném ezeket bemutatni, hogy jobb betekintést nyerjünk a működésükben. Az adatelemzésben a legelterjedtebbek az úgynevezett gépi tanuló algoritmusok. Segítségükkel mintákat, esetünkben trendeket lehet felismerni az adatokban, ezekből egy modellt állíthatunk össze, ami alapján előrejelzéseket végezhetünk. Nagy előnye, hogy az adatok és a tapasztalatok alapján folyamatosan tanítja önmagát, így az eredmények egyre pontosabbak. A gépi tanuló algoritmusoknak több megvalósítása is létezik, mindegyiknek megvan a maga előnye/hátránya. Munkám során én egy lineáris regressziós modellt és neurális hálózatot fogok kiépíteni.

2.2.2.1. *Lineáris Regresszió*

A lineáris regresszió nem más, mint amikor azt feltételezzük, hogy az adatok között valamilyen összefüggés van, tehát ha az adatok egyik halmaza független, a másik halmaz pedig ettől a független változótól függ. A függő változó értéke a valós számok halmazán bármilyen értéket felvehet. Legyen az egyik adathalmaz értéke (x_1 változó), amit a szakirodalom magyarázó vagy független változónak hív, ezt megszorozzuk egy konstans értékkel (legyen B_1), ami matematikailag értendő meredeksége az egyenesnek, majd hozzá adunk egy tetszőleges konstans értéket (legyen B_0) és a hibatagot (jelöljük ξ , hiszen egyetlen esetben sem lehet 100%-ban magyarázni az y -t) - ekkor megkapjuk a másik adathalmaz függő, magyarázandó értékét (y). Tehát a kétváltozós lineáris regresszió egyenlete így néz ki:

$$y = B_0 + B_1 \cdot x_1 + \xi.$$

Ha elhagyjuk a hibatagot, akkor megkapjuk a regressziós egyenes egyenletét, ezt nevezhetjük trend vonalnak is:

$$y = B_0 + B_1 \cdot x_1.$$

Minden esetben arra törekedünk, hogy az egyenes a legjobban illeszkedjen a pontokra, ehhez pedig a legkisebb négyzetek módszere a leggyakrabban használt módszer. Ennek a mérésére szolgál az R^2 mutató, ami 0-1 közötti értékeket vehet fel, de százalékos formában értelmezzük. Értelmezése azt takarja, hogy független változó hány százalékban magyarázza a függőt. (www.variance.hu, 2017)

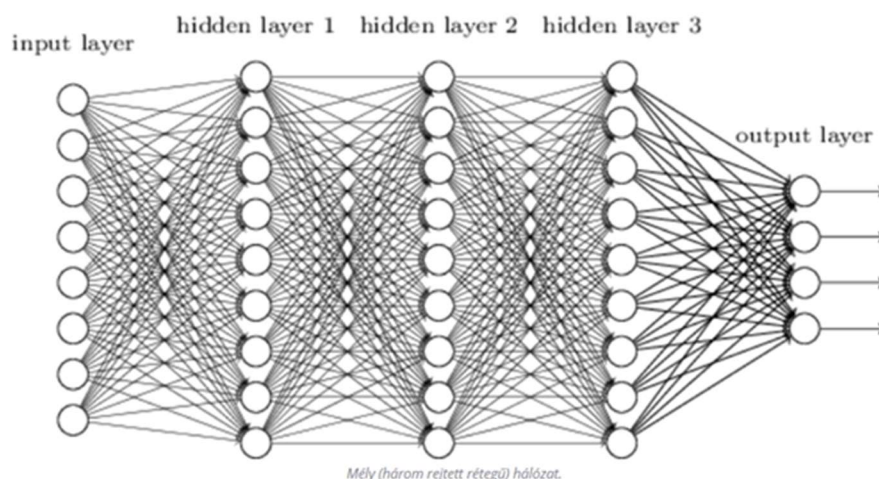
2.2.2.2. *Mesterséges neurális hálózatok*

Az emberi agy információfeldolgozását próbálják utánózni. A neuron egy olyan agysejt, aminek a feladata az elektromos jelek gyűjtése, feldolgozása és tovább küldése. A mesterséges neuronok egy vagy több egységből állnak, ezek hálózatba rendeződnek és egymással kommunikálnak. A kommunikáció során az élek súlyozva vannak, az egymásnak küldött jelek 0/1 lehetnek. Fontos különbség az emberi neuron és mesterséges neuronok között az, hogy az emberi neuronok és szinapszisok teljesen párhuzamosan működnek, míg a mesterséges neuronok legtöbb esetben szekvenciálisan, de esetenként párhuzamos végrehajtásra is lehetőség van, ez nagyjában függ a számítógép processzorától. Természetesen a mesterséges neuronokban a számítások sokkal gyorsabban hajtódnak végre. Ezek a hálók több rétegből épülnek fel és az információ bemeneti rétegtől a kimeneti rétegig haladnak (egy irányúan), de lehet visszacsatolás is a hálózatban. A 3. ábrán a működése jól megfigyelhető. Egy fontos

tulajdonsága még a neurális hálónak az, hogy képes a tanulásra. Ez alapján két típusát tudjuk megfigyelni: a felügyelt tanulást és a felügyelet nélküli tanulást. A hálózatok tanulása alatt legtöbbször a súly tényezők hangolását vagy a struktúra módosítását értjük. (Aradi, Gräff, Lipovszki, 2015)

A neuron kimentei értékének általános képlete:

$$y = f \left(\sum_{i=0}^d w_i a_i \right). \text{ (Farkas, 2021)}$$



3. Ábra: „Neurális háló példa”

Ezzel a résszel le is zárjuk az elméleti áttekintést. Most már szinte minden fontosabb témakört átbeszéltünk, megismertük az alapfogalmakat, így a következő fejezetben már áttérünk a gyakorlatra. Első körben az adatokat fogjuk megszerezni az elemzéshez, majd ezekből az adatokból származtatunk pénzügyi mutatókat, vagy eleve a pénzügyi mutatókat fogjuk kinyerni.

3. VÁLLALATOK PÉNZÜGYI ELEMZÉSE

A vállalatok pénzügyi elemzését már a megismert módon fogjuk végezni (lásd 2.1 pontban). Ehhez a cégek pénzügyi adataira lesz szükségünk, hiszen a mutatóinkat ezekből tudjuk származtatni. Régebben ezeket az adatokat cégjegyzék tárból lehetett megszerezni, de mára már szinte mindent megtalálunk az internetet böngészve. Így van ez a céges adatokkal is: rengeteg olyan oldal van, ahol nem csak a cégadatokhoz férünk hozzá ingyenesen, de előre kiszámolnak nekünk mutatókat, segítik a döntésünk meghozatalát ábrákkal, tanácsokkal. A legnagyobb nehézséget szerintem mégis az adatok feldolgozása jelenti, hiszen mára annyi

adathoz, információhoz jutunk hozzá, hogy szinte lehetetlen feldolgozni teljesen. Ezért is szeretném ezt az egészet automatizáltan elvégezni. A legnépszerűbb ilyen weboldal a Yahoo Finance (<https://finance.yahoo.com/>), ahol a részvények értékeinek változását lehet nyomon követni: én is itt szoktam, de amióta jobban belemerültem a dolgokba, megismerkedtem a Stock Analysis (<https://stockanalysis.com/>) weboldallal is, ahol sokkal több historikus adathoz férünk hozzá. Mindkét weboldal hasonlóan működik: a keresőjébe azt a részvényt kell beírni, ami minket érdekel és máris láthatjuk a fontosabb információkat az adott vállalatról. Ami számomra igazán érdekes - az a Financials menü alatt lévő Income, Balance Sheet, Cash Flow és Ratios.

Home » Stocks » AAPL » Financials » Ratios

Apple Inc. (AAPL)
NASDAQ: AAPL · IEX Real-Time Price · USD

142.81 -0.09 (-0.06%) **142.90** -0.39 (-0.27%)
After-hours: Oct 8, 2021 7:59 PM EDT At close: Oct 8, 4:00 PM

Overview Financials Statistics Dividends Profile Chart

Income Balance Sheet Cash Flow Ratios Annual Quarterly Trailing

Ratios and Metrics (Quarterly)

Market cap in millions USD. Fiscal year is October - September.

Export Financials

Quarter Ended	2021-06-26	2021-03-27	2020-12-26	2020-09-26	2020-06-27	2020-03-28	2019-12-28	2019-09-28	2019-06-29
Market Capitalization	2,221,291	2,034,885	2,243,728	1,920,273	1,532,751	1,083,981	1,287,658	988,887	908,887
Market Cap Growth	44.92%	87.72%	74.25%	94.19%	68.31%	21.02%	73.69%	-9.30%	
Enterprise Value	2,304,470	2,110,918	2,318,148	1,999,613	1,602,084	1,152,502	1,346,861	1,046,775	908,887
PE Ratio	25.59	26.67	35.10	33.45	26.23	18.95	22.38	17.90	
PS Ratio	6.40	6.25	7.63	7.00	5.60	4.04	4.81	3.80	
PB Ratio	34.56	29.41	33.88	29.39	21.20	13.82	14.38	10.93	
Debt / Equity Ratio	1.89	1.76	1.69	1.72	1.56	1.40	1.21	1.19	
Current Ratio	1.06	1.14	1.16	1.36	1.47	1.50	1.60	1.54	
Earnings Yield	0.98%	1.16%	1.28%	0.66%	0.73%	1.04%	1.73%	1.38%	
FCF Yield	0.86%	1.07%	1.57%	0.98%	0.96%	1.06%	2.21%	1.73%	
Dividend Yield	0.16%	0.17%	0.15%	0.18%	0.23%	0.31%	0.26%	0.35%	
Payout Ratio	16.80%	14.50%	12.10%	27.40%	31.50%	30.20%	15.30%	25.40%	30.20%
Buyback Yield / Dilution	1.03%	0.88%	1.10%	0.87%	1.04%	1.67%	1.61%	1.72%	
Total Shareholder Return	1.20%	1.05%	1.26%	1.05%	1.27%	1.98%	1.87%	2.08%	

4. Ábra: „Stock Analysis interfész”

Mint a 4. ábrán is látható, hogy csak a Ratios menü alatt hány mutatószám fellelhető, amit csak értelmeznünk kell. Azt könnyű belátni, hogy ezeket a mutatókat értelmezni még csak egy cég esetén is rengeteg idő, és akkor diverzifikációról nem is beszélünk. Mindenképpen valamilyen automatizált folyamatra lenne szükség, ami egyszerre több cég adatait le tudja szedni és ezeket az adatokat ki tudja értékelni. Ehhez első sorban ezek az adatok kellenek és

erre írtam egy programot, ami ezeket az adatokat szedi le Excelbe, majd a kiértékelést már Excel fájlban fogom elkészíteni.

3.1. VÁLLALATI ADATOK AUTOMATIZÁLT KINYERÉSE

Azt az eljárást, amikor egy adott weboldalról automatizáltan nyerünk ki adatokat - Web Scrapingnek nevezik. Én is egy ilyen programot írtam Python nyelven. Ahhoz, hogy ilyen programot tudjunk írni vagy futtatni, először le kell töltenünk a Pythont, azaz magát a kódsort értelmező programot, majd még egy IDE-t is. Az IDE - integrált fejlesztői környezetet jelent, segítségével tudjuk szerkeszteni a programkódunkat és tudjuk azt futtatni. Én ehhez az egyik legegyszerűbb és legnépszerűbb IDE-t használtam - a Jupiter Notebookot. Lássuk akkor a programot:

```
import os
import selenium #ha nem menne : !pip install selenium
from selenium import webdriver
#beállítom az alapértelmezett könyvtárat.
ldir = os.chdir(r'C:\Users\Dell 5300\Szakdolgozat_gyakorlat')
import pandas as pd
from numpy import random
from time import sleep

#Szüksége van Chromedriver.exe-re abba a könyvtárban, ahol a script is van.
print ('A modulok betöltöttek a program elindult: ')
```

A fenti kódsor lényege az, hogy beimportálom azokat a könyvtárakat, amit a programomban szeretnék használni. Itt négy fontos könyvtárat szeretnék megemlíteni, amelyekből az egyik a selenium, ami egy olyan könyvtár, amivel egy böngészőt lehet szimulálni és automatizálni. Én arra fogom használni, hogy töltsse be a kívánt linkeket és kattintson a Stock Analysis interfészen a negyedéves adatok fülre. A pandas az egyik legnépszerűbb adatelemző/adatfeldolgozó könyvtár, itt én arra fogom használni, hogy az interfészen megjelenő adatokat lementsem egy program változóba, majd kiírom Excel fájlba. De a pandas sokkal több ennél - különböző adat transzformáló műveletet és elemzéseket lehet vele csinálni, emiatt nagy hasznát fogom venni a mesterséges intelligencia programom írásakor is. A numpy a Python nyelvnek a matematika könyvtára, mi itt épp random számot fogunk vele generálni, a time-al meg egyszerűen csak egy időzítést fogunk tenni a programba, ha majd oda érünk, akkor érthetővé válik miért.

A Chromedriver igazából egy olyan eszköz, ami lehetővé teszi a GoogleChrome böngésző szimulálását. Ezt le kell tölteni az internetről:

<https://chromedriver.storage.googleapis.com/index.html?path=94.0.4606.61/>.

Letöltés után be kell másolni abba a könyvtárba, ahol a forrás kódunk is van. Javaslatom mindenki számára, hogy informáljuk a felhasználót arról, hogy éppen mi történik. Én is tettem a kódba egy egyszerű kiíratást, hogy minden modul sikeresen betöltődött, hiszen az informatika világára nagyon igaz Murphy törvénye „ami elromolhat, az el is romlik”, és az már egy nagyon nagy segítség, hogy tudjuk mi romlott el.

```
#Beállítom, hogy mi legyen az Excel file neve.
xlwriter = pd.ExcelWriter('Penzugyi_adatok.xlsx', engine='xlsxwriter')
#Egyszer hozzá kell adni egy chore dirvert a selliniumnak, ezt a motort fogja
használni a script a böngészéshez
link = f'https://stockanalysis.com/stocks/'
driver.get(link) #Betölti a program a Stock Analysis oldalát
sleeptime = 7 #Várjon a program 7 mpt
driver.find_element_by_id('accept-
choices').click() #Mivel a weboldal Cookikat használ ezért megkerestem a Cooki
#elfogadásának a gombját és megkértem a seleniumot, hogy kattintson rá nekem.
driver.find_element_by_xpath("//option[text()='10000']").click() #Kiválasztottam, hogy egy la
pon 10.000 értékpapírt mutasson.
newDF = pd.DataFrame() #készítettem egy DataFrame változót.
newDF = pd.read_html(driver.page_source) #Beleolvastam a változóba az értékpapírokat.
df = newDF[0].dropna(axis=0, thresh=4) #Töröltem a nullokot.
df.to_excel(xlwriter, sheet_name='Vezerlo', index=False) #Lementettem az adatokat a vezerlo
nevű Sheetre.
print('Értékpapírok sikeresen letöltöttek') #Felhasználó tájékoztatása.
```

A fenti kódrészletben azt csinálom, hogy megadom a létrehozandó Excel fájl nevét és megmondom a programnak, hogy honnan tudja elindítani a Chrome drivert. Ezután felmegy a program a Stock Analysis főoldalára és letölti az összes értékpapírt. Így az Excel fájlomban 5000 darab értékpapír neve, szimbóluma, szektora és piaci kapitalizációja lesz elérhető.

```
#Ezeknek cégeknek szedi le a script az adatait, ehhez bármennyit hozzá lehet tenni
cegek = ['AAPL','F','TSLA','AMZN']
#Minden cégen átiterálunk
for c in cegek:
    print(f'{c}-adatok letöltése:')
    #Különböző adatok
különböző elérési utakon vannak, ezért mindet hozzáférhetővé kell tenni a programnak
    urls = {}
    urls['income'] = f'https://stockanalysis.com/stocks/{c}/financials/'
    urls['balance sheet'] = f'https://stockanalysis.com/stocks/{c}/financials/balance-sheet/'
    urls['cash flow'] = f'https://stockanalysis.com/stocks/{c}/financials/cash-flow-statement/'
    urls['ratios'] = f'https://stockanalysis.com/stocks/{c}/financials/ratios/'
```

A fenti kódrészletben betöltöttem a selenimuba a Google Chrome driverét, a „cegek” változóba betöltöttem azoknak a cégeknek a részvényeinek a szimbólumait, amik érdekelnek

engem. Ezeket a szimbólumokat a neten kell megkeresni, erre egy jó oldal a: <https://www.nasdaq.com/market-activity/stocks/screener>. Itt szűrő feltételeket is lehet beállítani, de ezek a szimbólumok nálam már meglesznek az Excelben, és a jövőben onnan fogom kimásolni őket.

Pár szóban elmagyaráznám azt is, hogy hogyan működnek az urls változók is. Erre pedig nézzünk meg egy példát, hogy hogyan épül fel a StockAnalysis elérési útvonala:

<https://stockanalysis.com/stocks/aapl/financials/cash-flow-statement/>

- <https://stockanalysis.com> – ez lesz a weboldal címe, de ezzel nem mondtam újat. Jelöljük * ezt.
- */stocks - a weboldal ezen a elérési útvonalon „tárolja” a értékpapírokat.
- */stocks/aapl - ez már érdekesebb, azt látjuk, hogy az elérési útvonalban paraméterként szerepel a részvény szimbóluma. A programom ez alapján működik, ezt a szimbólumot cserélgetem a „cegek” változóban lévő szimbólumokra. Így át tudok iterálni az összes cégen.
- */stocks/aapl/financials/ - ha jól megfigyeljük a 4. Ábrát akkor látni fogjuk, hogy minden részvénynek lesz egy Overview, Financials, Statistics, stb. füle, amin a megfelelő adatok szerepelnek a cégről vagy a részvényről. Ez is átadható paraméterként az útvonalban - így, ha a „Financials” szót írjuk az útvonalba a szimbólum után, akkor a pénzügyi mutatókat kapjuk meg és így tovább.
- */stocks/aapl/financials/cash-flow-statement – így már érhető lesz, hogy a Financials alatt lesz még cash-flow-statement, és még sok minden más, szintén a 4. Ábrának megfelelően.

Jól látható, hogy én csak annyit tettem, hogy kiszedtem a Financials lap alól az összes elérési útvonalat és a szimbólum paramétert ciklikusan kicseréltem az én „cegek” változóban lévő szimbólumra. Itt arra kell figyelni, hogy Python nyelvben nincsenek pontosvesszők a parancsok lezárására, hanem alapesetben az új sor kezdte jelenti a sor végét. Ciklusoknál pedig figyelni kell a behúzásokra, hiszen ez mondja meg a Pythonban a ciklusnak, hogy milyen kód tartozik a ciklusba és melyik kód nem a ciklushoz tartozik.

#A „start” azért van, hogy követni tudja a script, hogy melyik sorba kell írni ép az adatot, majd az Excel fájlba, nehogy felülírjam véletlenül az egyik adattal a másikat.

```
start=0
```

```
#Minden linken átiterálunk
```

```
for key in urls.values():
```

```
    #Felhasználónak információ, hogy épp melyik linknél jár
```



```

print(key, end=" ")
#Betöltjük a linket a böngészőbe.
driver.get(key)
#Random időkre altatjuk a programunkat, hogy úgy tegyünk mintha egy ember csinálná,
így nem fogják bannolni az ip címünket – nem minden cég szereti, ha scrapelik.
sleeptime = random.uniform(2, 5)
print("alvás:", sleeptime, "másodpercig")
sleep(sleeptime)

```

A fenti rész már nagyon fontos - ha mindet jól csinálunk, akkor a program megnyitja a Google Chrome-ot és egyesével átlépked az „urls” változóban definiált url-eken, mindezt úgy csinálja, mintha egy ember lenne, azaz minden oldalon eltölt egy bizonyos időt, ezért kellett a time modul. Ezt ciklikusan hajtja végre az éppen adott szimbólumra.

#Utasítom a programot, hogy kattintson a Quarterly gombra, hogy megkapjam a negyedéves adatokat

```
driver.find_element_by_xpath("//span[text()='Quarterly']").click()
```

Lehet, hogy bonyolultnak látszik, de nem az: igazából annyi történt, hogy megnéztem a Stock Analysis forráskódját (Microsoft Edge-nél jobb klikk, Vizsgálat) és megkerestem, hogy mi a neve a Quartely gombnak (Quartely volt). Még azt is le kellett írni, hogy milyen HTML elembe van beágyazva, az pedig egy span tag volt. Mondhatnám azt is, hogy azt mondtam a programnak, hogy keressen nekem egy olyan span taget, aminek a neve 'Quartely'

```

#Csinálok egy új üres dataFrameet
newDF = pd.DataFrame()
#Be olvasom a dataFrame-be az adatokat
newDF = pd.read_html(driver.page_source)
#Mivel alapértelmezetten listába tölti le az adatokat, ezért törlöm az üres adatokat, és egy új dataFrameet csinálok belőle, aminek a Type-a már tényleg dataframe
df = newDF[0].dropna(axis=0, thresh=4)
#Beírom az adatokat az excelbe
df.to_excel(xlwriter, sheet_name=c, startrow=startr, index=False)
#Beállítom a következő tömbnek a kezdő sorszámát, hogy ne írja felül az adatokat a program
startr=startr+df[df.columns[0]].count()+1

#Mentem az Excel fájlt
xlwriter.save()
#Bezárom a böngészőt
driver.close()
#Tájékoztatom a felhasználót.
print('Sikeresen letöltöttek az adatok')

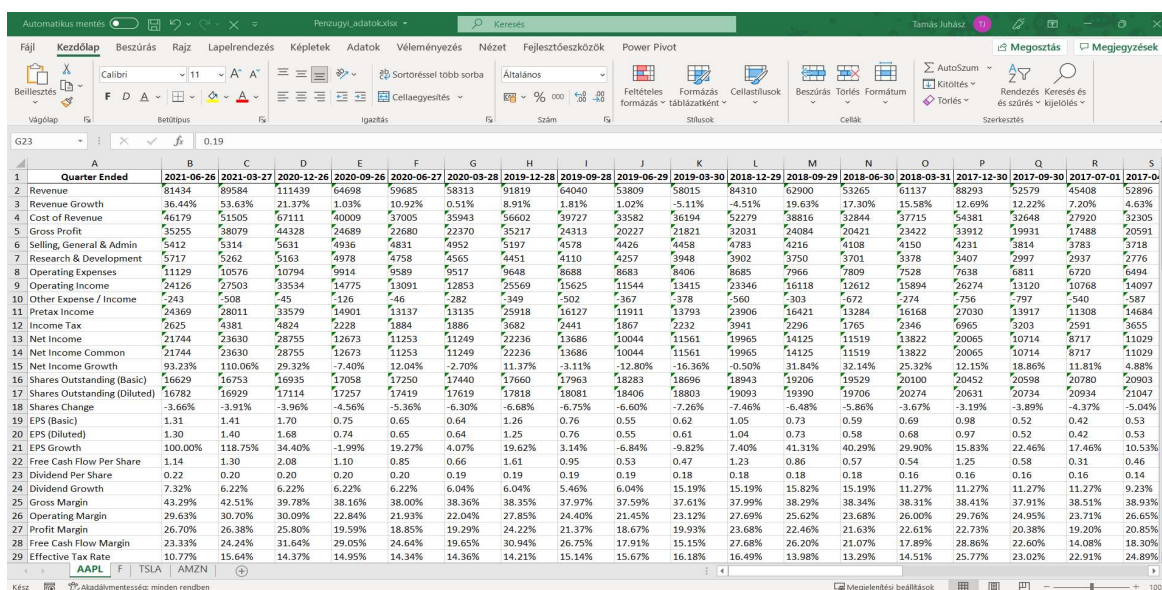
```


A fenti kódrészlet pedig azt tartalmazza, hogy készítettem egy változót és abba beletöltöttem a weboldalon megjelenő adatokat. Kiemelném, hogy mennyire intelligens a pandas, `pd.read_html` paranccsal azonnal kiolvasta az összes adatot. Látható, hogy semmilyen HTML taget nem adtam meg neki, hogy honnan olvassa ki, és mégis sikeresen kiolvasta a teljes adathalmazt. Ezután még végeztem egy adatkonverziót, mert amikor lejöttek az adatok, egy lista típusú változóba lettek beleírva, de a lista objektumnak nincsenek olyan függvényei/eljárásai, hogy `to_excel`, ezért egy `DataFrame` nevezetű adat típusba kellett konvertálni. Ez az adattípust arra használják, hogy adattáblákat tároljanak le benne. Ez nagyon kézre eső, mivel teljes adattáblás műveletek kapcsolódnak hozzá, olyanok, mint a `to_excel` és még sok-sok más, amit később használni is fogunk. A konvertálást úgy lehetett megcsinálni, hogy töröltük a Null értékeket (`newDF[0].dropna(axis=0, thresh=4)`). Majd a létrejött DataFramen meghívom a `to_excel` eljárást, amivel az adatok beleíródnak az Excel fájlba.

A ciklikusságról még nem esett szó. Két egymásba ágyazott for ciklus működteti a programot. Az első ciklusban kiveszi azt a részvény szimbólumot, aminek az adataira kíváncsiak vagyunk, és erre a szimbólumra regenerálja a kód az összes Financials fül alatt lévő elérési utat (Income, Balance Sheet, Cash Flow, Ratios lásd 4. Ábra.). Ezeket egy listában tárolom. Majd a második ciklus ezen a listán iterál át, és minden egyes elérési útvonalat megnyitok a seleniumnal, és letárolom az ott lévő adatokat, majd megyek a következő linkre. Azt még kiemelném, hogy `df.to_excel(xlwriter, sheet_name=c, startrow=starttr, index=False)` - azt jelenti, hogy amikor mentem az adatokat, minden egyes részvénynek létrehozok egy saját lapot. Ha elfogynak a linkek megyek a következő részvény szimbólumra és így tovább. A program végén mentem az Excel fájlt, bezárom a Google Chrome-ot és tájékoztatom a felhasználót, hogy minden sikerült.

Így tehát sikerült letölteni a Stock Analysis összes részvényének a nevét, szimbólumát, azt, hogy melyik szektorban van és hogy milyen kapitalizációval rendelkezik. Továbbá a program arra is képes, hogy egyesével szedje ki a részvények statikus pénzügyi mutatóit, mérlegét és egyéb más információkat a részvényre nézve. Ez rengeteg adatot jelent, ami látható az 5. Ábrán.

Kezdődhet az elemzés!



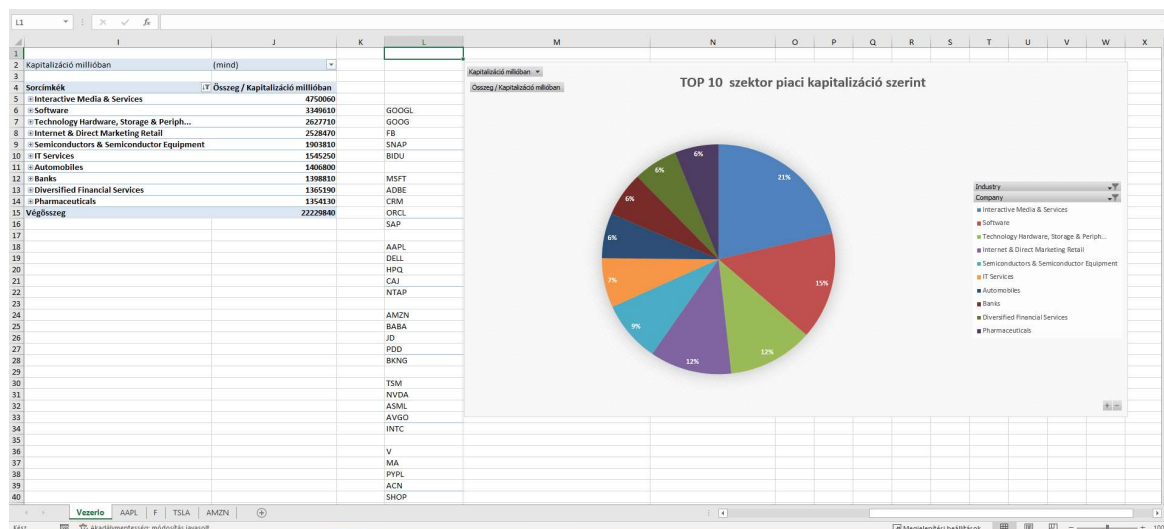
	2021-06-26	2021-03-27	2020-12-26	2020-09-26	2020-06-27	2020-03-28	2019-12-28	2019-09-28	2019-06-29	2019-03-30	2018-12-29	2018-09-29	2018-06-30	2018-03-31	2017-12-30	2017-09-30	2017-07-01	2017-0
1 Quarter Ended																		
2 Revenue	61434	89584	111439	64698	59685	58313	91819	64040	53809	58015	64310	62900	53265	61137	88293	52579	45408	52896
3 Revenue Growth	36.44%	53.63%	21.37%	1.03%	10.92%	0.51%	8.91%	1.81%	1.02%	-5.11%	-4.51%	19.63%	17.30%	15.58%	12.69%	12.22%	7.20%	4.63%
4 Cost of Revenue	46179	51505	67111	40009	37005	35943	56602	39727	33582	36194	52279	38816	32844	37715	54381	32648	27920	32305
5 Gross Profit	35255	38079	44328	24689	22680	22370	25172	24313	20227	21821	12031	24084	20421	23422	33912	19931	17488	20591
6 Selling, General & Admin	5412	5314	5631	4936	4831	4952	5197	4578	4426	4458	4783	4216	4108	4150	4231	3814	3783	3718
7 Research & Development	5717	5262	5163	4978	4758	4565	4451	4110	4257	3948	3902	3750	3701	3378	3407	2997	2937	2776
8 Operating Expenses	11129	10576	10794	9914	9589	9517	9648	8688	8683	8406	8685	7966	7809	7528	7638	6811	6720	6494
9 Operating Income	24126	27503	33534	14775	13091	12853	15569	15625	11544	13415	33346	16118	12612	15894	26274	31310	10768	14097
10 Other Expense / Income	-243	-508	-45	-126	-46	-282	-349	-502	-367	-378	-560	-303	-672	-274	-756	-797	-540	-587
11 Pretax Income	24369	28011	33579	14901	13137	13135	15918	16127	11911	13793	23906	16421	13284	16168	27030	31917	11308	14684
12 Income Tax	2625	4381	4824	2228	1884	1886	3682	2441	1867	2232	3941	2296	1765	2346	6965	3203	2591	3655
13 Net Income	21744	23630	28755	12673	11253	11249	22236	13686	10044	11561	19965	14125	11519	13822	20065	10714	8717	11029
14 Net Income Common	21744	23630	28755	12673	11253	11249	22236	13686	10044	11561	19965	14125	11519	13822	20065	10714	8717	11029
15 Net Income Growth	93.23%	110.06%	29.32%	-7.40%	12.04%	-2.70%	11.37%	-3.11%	-12.80%	-16.36%	-0.50%	31.84%	32.14%	25.32%	12.15%	18.86%	11.81%	4.88%
16 Shares Outstanding (Basic)	16629	16753	16935	17058	17250	17440	17660	17963	18283	18896	18943	19206	19529	20100	20452	20598	20780	20903
17 Shares Outstanding (Diluted)	16782	16929	17114	17257	17419	17619	17818	18081	18406	18803	19093	19390	19706	20274	20631	20734	20934	21047
18 Shares Change	-3.66%	-3.91%	-3.96%	-4.56%	-5.36%	-6.30%	-6.68%	-6.75%	-6.60%	-7.26%	-7.46%	-6.48%	-5.86%	-3.67%	-3.19%	-3.89%	-4.37%	-5.04%
19 EPS (Basic)	1.31	1.41	1.70	0.75	0.65	0.64	1.26	0.76	0.55	0.62	1.05	0.73	0.59	0.69	0.98	0.52	0.42	0.53
20 EPS (Diluted)	1.30	1.40	1.68	0.74	0.65	0.64	1.25	0.76	0.55	0.61	1.04	0.73	0.58	0.68	0.97	0.52	0.42	0.53
21 EPS Growth	100.00%	118.75%	34.40%	1.99%	19.27%	4.07%	19.62%	3.14%	-6.84%	-9.82%	7.40%	41.31%	40.29%	29.90%	15.83%	22.46%	17.46%	10.53%
22 Free Cash Flow Per Share	1.14	1.30	2.08	1.10	0.85	0.66	1.61	0.95	0.53	0.47	1.23	0.86	0.57	0.54	1.25	0.58	0.31	0.46
23 Dividend Per Share	0.22	0.20	0.20	0.20	0.20	0.19	0.19	0.19	0.19	0.18	0.18	0.18	0.18	0.16	0.16	0.16	0.16	0.14
24 Dividend Growth	7.32%	6.22%	6.22%	6.22%	6.22%	6.04%	6.04%	5.46%	6.04%	15.19%	15.19%	15.82%	15.19%	11.27%	11.27%	11.27%	11.27%	9.23%
25 Gross Margin	43.29%	42.51%	39.78%	38.16%	38.00%	38.36%	38.35%	37.97%	37.59%	37.61%	37.99%	38.29%	38.34%	38.31%	38.41%	37.91%	38.51%	38.93%
26 Operating Margin	29.63%	30.70%	30.09%	22.84%	21.93%	22.04%	27.85%	24.40%	21.45%	23.12%	27.69%	25.62%	23.68%	26.00%	29.76%	24.95%	23.71%	26.65%
27 Profit Margin	26.70%	26.38%	25.80%	19.59%	18.83%	19.29%	24.22%	21.37%	18.67%	19.93%	23.68%	22.46%	21.63%	22.61%	22.73%	20.38%	19.20%	20.85%
28 Free Cash Flow Margin	23.33%	24.24%	31.64%	29.05%	24.64%	19.65%	30.94%	26.75%	17.91%	15.15%	27.68%	26.20%	21.07%	17.89%	28.86%	22.60%	14.08%	18.30%
29 Effective Tax Rate	10.77%	15.64%	14.37%	14.95%	14.34%	14.36%	14.21%	15.14%	15.67%	16.18%	16.49%	13.98%	13.29%	14.51%	25.77%	23.02%	22.91%	24.89%

5. Ábra: „WebScraping eredménye”

3.2. ADATOK FELDOLGOZÁSA

Miután meglelt ez a rengeteg adat, valahogyan fel is kellett dolgozni, ehhez az Excel beépített függvényeit használtam. Először is azt tettem, hogy a letöltött kapitalizációt visszavezettem millióba, mert rövidítve voltak megadva az értékek, pl. 245B vagy 150M. Ezeket egységesíteni kellett, ezért levágtam mindenhol a karaktereket és betettem egy külön oszlopba. Ezután ahol M ott simán millióval szoroztam, ahol B volt ott pedig 1000 milliót vettem. A következő művelet az volt, hogy összeállítottam az eredményből egy pivot táblát. A pivot táblára készítettem pár szűrő feltételt. A logikáját a következő részben szeretném részletezni, elég most annyit tudni, hogy engem csak azok a részvények érdekelnek, aminek a szektora eléri a 1.000.000M dollárt, ezek közül a szektorok közül pedig csak Top legnagyobb kapitalizációju cég érdekel engem. A cégeket még megszűrtem úgy, hogy csak azok a cégek maradjanak, amiknek a kapitalizációja eléri a 2 milliárd dollárt. Majd ezeket csökkenő sorrendbe rendeztem. Így csoportosítva kaptam egy listát a legnagyobb kapitalizációjú szektorokról és részvényekről. Majd a következő oszlopba beállítottam egy X keres függvényt a cégnevére. A függvényt úgy paramétereztem, hogy adja vissza a sorban lévő cég szimbólumát a táblámból. Sajnos még ilyen szigorú paraméterek mellett is majdnem 70 szektor volt, amit figyelni kell. Ezután még tovább gondoltam az egészet. Tettem pár segéd oszlopot és minden cég mellé kitétem azt, hogy mennyi a szektorának a teljes kapitalizációja, és mellé egy

másikban megnéztem, hogy ebből hány százalékot tesz ki az adott cég. Ezt úgy hoztam létre, hogy egy segéd pivot táblába kiszámoltattam minden egyes szektornak a teljes kapitalizációját,



6. Ábra: „Dashboard”

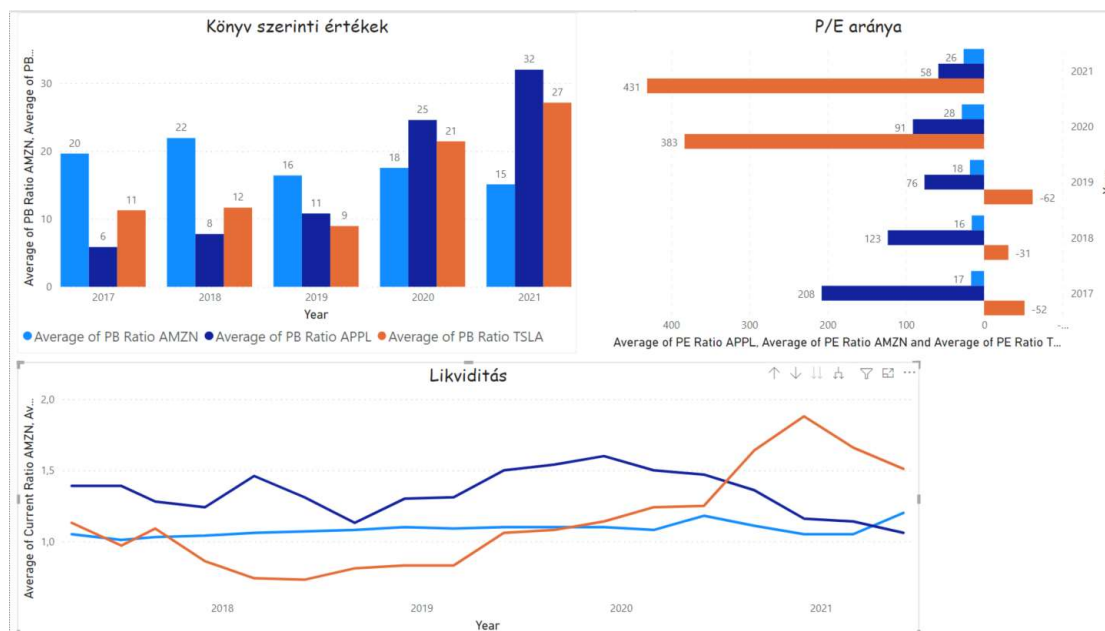
majd visszaraktam az értékeket a vezérlő lapra. Majd egy Xkres függvénnyel megkerestem az adott cég szektorát az én szektor / szektor kapitalizáció táblámban, és a visszaadott érték a szektor kapitalizációja volt. Ebből az egészből készítettem magamnak egy vezérlőpultot ez látható a 6. Ábrán.

3.3. ADATOK ELEMZÉSE

Ebben a részben akkor kitérnék arra, hogy miért is pont azokat a filtereket állítottam be az Excel fájlomban. Ehhez Benjamin Graham könyvét vettem alapul, akit a világ minden idők egyik legnagyobb befektetőjének tartanak. Az ő tanítványa Warren Buffet, aki a világ egyik leggazdagabb embere, és a vagyonának a nagyrészt pénzügyi befektetésekből építette fel. Az első szabály, amit Graham szerint be kell tartanunk az az, hogy kisméretű vállalatokat ki kell szűrni, ezért a biztonság kedvéért én is kiszűrtem azokat és azokat is, amelyeket én „kicsi”-nek gondolok. A kisvállalatokat a 2 milliárdos határnál húztam meg, az ennél kisebb kapitalizációjú részvények kikerültek az elemzésemből. Ez azt jelentette, hogy a 5500 darab vállalatból már csak 1800 vállalat maradt. Warren Buffet egy interjúban kifejtette, hogy ez azért szükséges, hogy az esetleges válságok idején a cég legyen elég nagy ahhoz, hogy a károkat kezelni tudja. Ezt láttuk a COVID pandémia alatt is - a kis cégek sajnos nem tudtak életben maradni. (Graham-1,2011)

A következő tanácsa Grahamnak az, hogy legyen stabil pénzügyi helyzete a cégnek. A gyakorlatban ezt azt jelenti, hogy keressük azokat a vállalatokat, amik legalább 2 likviditási aránnyal rendelkeznek, illetve szintén fontosnak tartja azt, hogy stabil nyeresége legyen a vállalatnak. Az ajánlás itt úgy szól, hogy az elmúlt 10 évben a nettó árbevétele legyen nagyobb nullánál. Ezek a feltételek a későbbiekben lesznek fontosok. Még megemlítenék két ilyen arányt, amit szintén megemlít Graham a könyv szerinti értékre és a P/E mutatóra. Az javasolja, hogy 1,5 legyen az árfolyam és a könyv szerinti értéke (P/B). Itt meg kell jegyezni azt is, hogy manapság magasabb átlagáron jegyzik a könyv szerinti értékeket, így a Morgen Stanley szerint 2,5 arány is elég szigorú. Illetve a másik az az, hogy törekedjünk a minél kisebb (20) P/E arányú cégek vásárlására, erős korreláció van a magas P/E ráta és az alacsony hozam között: minél magasabb az adott cég P/E rátája - annál kisebb hozamra számíthatunk a jövőben. Érdekes lehet szektoronként meghatározni a szektorátlaghoz az adott vállalat rátájának az arányát, így tudni fogjuk, hogy szektorhoz képest mennyivel drágább az adott részvény. (Graham-2,1973) (Nagy Attila-1, 2021)

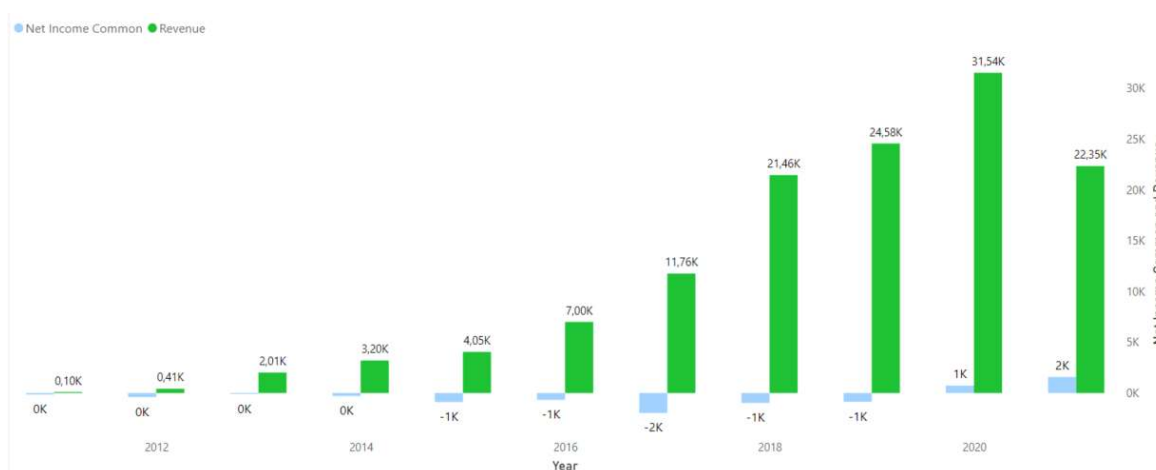
A másik filter, amit beállítottam, arról szólt, hogy az adott szektorban hány százalék a cég kapitalizációja a teljes szektor kapitalizációhoz képest. Ezt szintén Warren Buffet egyik interjújában magyarázta el, miszerint ő preferálja a monopóliumban lévő cégeket. Ezt azzal a kérdéssel tette fel, hogy ha tippelnünk kellene, hogy 10 év múlva melyik kóla gyártó fogja



7. Ábra: „Likviditás, P/B, P/E értékek”

produkálni a top eladásokat a Coca Cola vagy számunkra jelenleg még egy ismeretlen cég? Valószínű, hogy a Coca Cola lesz az, de ha nem is az első, akkor is benne lesz a top 5-ben. Mivel ez egy olyan cég ami, úgymond már lejátszotta a versenyét és már régóta a piacon van, szinte bebetonozva a sikerét. (Graham-3,1973)

Ha ezek az ajánlások alapján kezdem értékelni a részvényeket - már megbukott volna az összes részvény (lásd a 7. Ábrát, amit a dolgozatom elején felvetettem). Hiszen mind az Apple mind az Amazon részvénye túlértékeltnek tűnik hiszen, PB már mindegyiknek 20 fölött van. Illetve mindkét cég likviditási rátája 1 körül mozog. A Tesla részvényről nem is beszélek, aki jelenleg ott tart, hogy a P/E rátája 431, azaz a részvény értékének a megtérülése 431 év múlva fog megtörténni. A jövedelmezőségi mutatószámok az Amazon-nál és Apple-nél rendben vannak, viszont számomra ami hihetetlen volt, az a Tesla pénzügyi mutatószámai. A cég évek óta



8. Ábra: „Tesla bevétel, nyereség aránya”

veszteségesen működik, a 2020. év volt az első olyan év, amikor jövedelmező volt és akkor is az árbevétel arányos nyereség nem érte el a 9%-ot, a 2021 évben pedig már csökkent is a bevétele. Nem valami fényes a helyzet, persze szokták mondani, hogy árazza a jövőt a piac, de itt valószínűleg egy hatalmas nagy lufi van. Ezt nem csak én gondolom így, több elemző is 35% korrekciót vár legalább az árfolyamától, azaz 1200 dollár körül mozog a részvény ár nagyjából 780 dollárra fog visszaesni egy éven belül. Viszont azt meg kell hagyni, hogy hatalmas ütemben fejlődik a cég: a piaci kapitalizációja 2019-ben 75,402M dollár volt, 2020-ban már 668,900M dollár, ami 787% növekedést jelent. Arra nagyon kíváncsi vagyok, hogy mennyi marad a lufi kidurranása után. Szóval, sajnos a kezdeti részvényeim mind drágának bizonyulnak jelenleg, de az Excelemmel nem csak ezt derítettem ki, hanem azt is, hogy mik azok a részvények, amik

jó áron lehetnek, illetve jó piaci helyzetben vannak. Kicsit sikerült tovább is fejlesztenem az Excelt: az Adatok fülön találhatunk egy olyan gombot, hogy Részvény és ezt alkalmaztam. Úgy kellett, hogy egyszer kijelöltem a Szimbólumokat és rákattintottam erre a gombra, majd az Excel automatikusan felismerte a szimbólumukat, és mint hivatkozás ezek után tudtam kezelni a részvényre vonatkozó paramétereket úgy, mint ár, kapitalizáció, P/E mutató. Be is húztam ezeket a mezőket a táblázatomba. Ezek után beállítottam egy sorrendet, miszerint először azt nézze, hogy a teljes piaci kapitalizációból hány százalékot tesz ki az adott részvény, a másodlagos sorrend a szektor kapitalizációja volt, utána a részvény kapitalizációja, végül pedig a P/E mutató. Az Excel file már hihetetlenül interaktív lett, és rálátást adott azokra a részvényekre, amik tényleg a piac csúcsán helyezkednek el és monopol helyzetben vannak és

Symbol	Company	Industry	Market Cap	Kapitalizáció milliárd USD	Szektor	Kapitalizáció %	Price	P/E
329,98	Meta Platforms, Inc.	Technology	329,98	329,98	Technology	27,64	27,64	23,12
...

9. Ábra: „Legjobb részvények”

már azt is látom az Excel fájlból, hogy hogyan vannak árazva ezek a cégek.

Így most már rögtön látom azt, hogy melyik cégnek éppen, hogy alakul a P/E rátája és árfolyama. Például a nem rég nevet változtatott Facebook, aki már Meta Platforms néven fut, árfolyama igen jó áron van most, a P/E rátája 23,12 ami számomra nagyon jónak tűnik, hiszen minden más tech óriás cég 25 fölött mozog, a Microsoft már lassan a 40-et veri, az Apple a kettő között van valahol, 26 pontjával, ami szintén átlagosnak mondható ebben a szektorban. Ha visszatérünk az autó iparra, akkor jól látható, hogy a Tesla birtokolja a teljes kapitalizáció 54%-át jelenleg, de nem szabad figyelem kívül hagyni a japánokat sem, hiszen a Toyota-é a piac 16%, de az ő P/E mutatójuk jóval kecsesebb számomra, hiszen az 10 körül mozog jelenleg. Ezek jó kiindulási pontot adhatnak, de nem szabad elfelejteni, hogy a jövedelmezőségi

elemzéseket is kell végezni ezeken a cégeken. Én a dolgozatomban elején említett cégekkel Tesla, Amazon, Apple fogok tovább menni a példa kedvéért. Plusz arra is kíváncsi vagyok, hogy egy mesterséges intelligencia algoritmus mire megy ezekkel az adatokkal, vajon ő is észre fogja-e venni ezeket az előrejelzéseket, pl. a Tesla árfolyamáról (35% esés).

4. RÉSZVÉNY ÁRFOLYAM ELŐREJELZÉS MESTERSÉGES INTELLIGENCIA PROGRAMOKKAL

A dolgozatomban elején a [2.2](#) pontban már írtam arról, hogy milyen algoritmusokat fogok használni az elemzéshez. Kicsit szeretném részletesebben kifejteni, hogy milyen implementációt fogok használni ezeknél az eljárásoknál. Az idősoros elemzéseknél jóval szélesebb körben használt módszer az ARIMA.

4.1. ARIMA MODELL

Az ARIMA egy mozaik szó, aminek a jelentése Autoregresszív Integrált Mozgóátlag. Ennek a modellnek az alapja, hogy a múltbeli értékek alapján próbálja meg megjósolni a jövőbeli értékeket. Ennek az egyik nagy előnye, hogy mozgóátlagokat használ a trend megértéséhez, de azt feltételezi, hogy a jövő hasonlítani fog a múltra. Ebből az következik, hogy a hosszú távú modellezések nem fognak ezen jól működni, hiszen a piaci hangulat egyik pillanatról a másikra változhat, illetve a népszerűségi trendeket, hullámokat nem tudja értelmezni. Tehát ha esetleg holnaptól medvepiac lesz, vagy egy kisebb válság, akkor teljes mértékben el fog térni a prediktált ár az aktuális ártól. Hiszen ennek a módszernek az a lényege, hogy a múltbeli adatok mozgóátlagára illeszt egy egyenest, és azt feltételezi a modell, hogy ez az egyenes a jövőben is így folytatódik majd, tehát a piac nagy változásait nem képes lekövetni (pl. válságok, csődök). A modellt 3 fő paraméter vezérli:

- p: késési időszak, (pl. ha értéke 3, akkor az azt jelenti, hogy 3 korábbi periódust használunk számításhoz).
- d: differencing mértéke, ezzel azt határozzuk meg, hogy az két érték között hányat kell lépni, hogy megkapjuk az átlagot.
- q: mozgóátlag mérete, az az hány elemen számolja a mozgóátlagot.

Fontos tudni az ARIMA-ról, hogy arra alapoz, hogy az előző érték befolyással van a következő értékre, és a két érték között valamilyen regressziós függvénnyel keresi a kapcsolatot. Ezt a modellt és a következőt is szintén Python nyelven implementáltam. (Beverin, 2021)

A programhoz a következő könyvtárak lesznek szükségesek:

```
import os
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#pip install statsmodels==0.11.0 csak ezzel megy újabbal nem fog működni.
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA
from pmdarima.arima import auto_arima
from sklearn.metrics import mean_squared_error, mean_absolute_error
import math
from pylab import rcParams
import yfinance as yf
print('Modulok betöltöttek')
```

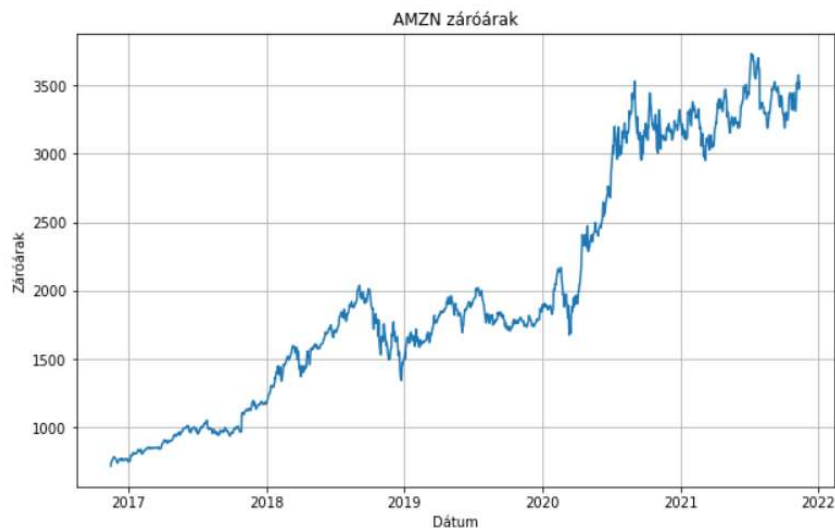
A modulok közül kiemelném a statsmodels könyvtárat – ezzel fogjuk elvégezni a főbb statisztikai műveleteket, és építjük majd fel az ARIMA modellt. Egy másik fontos pedig pmdarima könyvtár, aminek segítségével könnyedén meg lehet találni a legjobb (p, d, q) értékeket. Illetve találtam egy Yahoo Finance-os könyvtárat, amivel azt tudjuk megtenni, hogy csak megkell majd adnunk a részvény szimbólumát és a Yahoo Finance-ról leszedi a szimbólumhoz tartozó historikus adatokat. (Beverin-2, 2021)

```
#Adatok letöltése a megfelelő paraméterekkel
szimbolum = 'TSLA'
#itt megadjuk a paraméterkét, elmúlt 5 év minden napját kérem
adatok = yf.download(tickers=szimbolum,period='5y',interval='1d')
adatok
```

A fenti kód segítségével töltöm be az adatokat egy változóba.

```
#Adatok megjelenítése
plt.figure(figsize=(10,6))
plt.grid(True)
plt.xlabel('Dátum')
plt.ylabel('Záró árak')
plt.plot(adatok['Close'])
plt.title(szimbolum + " záró árak")
plt.show()
```


Ez a kódrészlet egy egyszerű vizualizáció, aminek a segítségével megjelenítem a záróárakat a adatok nevű változóból.



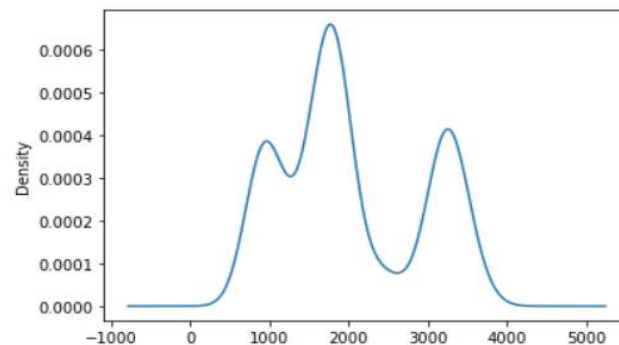
10. Ábra: „Amazon záróárfolyam”

Erről fentebb látható is az eredmény a 10. ábrán, amit a matplotlib.pyplot könyvtár segítségével jelenítettem meg. Ezen kívül beletettem jó pár vizualizációt az adatok jobb értelmezéséhez.

#Adatok eloszlása

```
zaro_arak = adatok['Close']  
zaro_arak.plot(kind='kde')
```

Ezzel a pársorral a záró adatokat jelenítettem meg egy haranggörbén, aminek az eredménye a következő volt (11. Ábra):



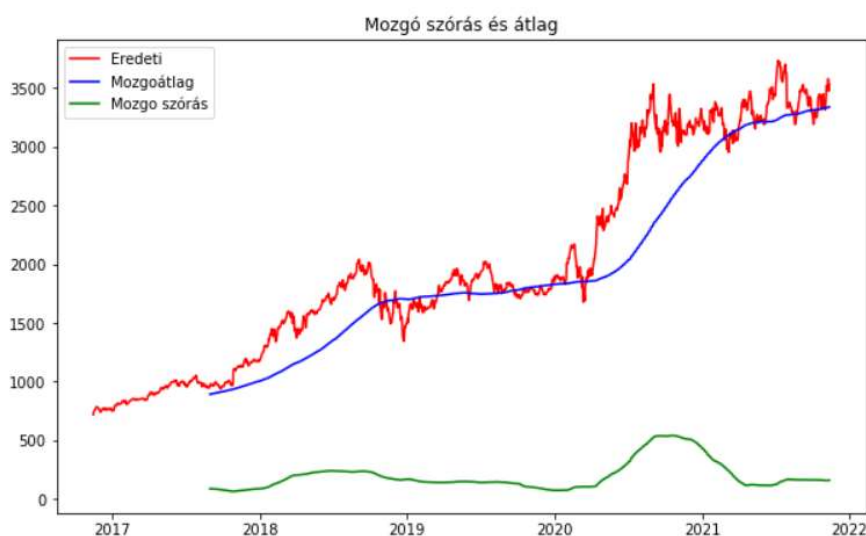
11. Ábra: „Amazon záróárfolyama normál görbére vetítve”

Számomra ez nagyon sokat mondó hiszen leolvashatom erről, hogy a legtöbb adat az nagyjából 2000 dolláros árfolyamon mozog, és az adatokra inkább jellemzőek a nagyobb árak mint az alacsonyok, de az is látszik, hogy hatalmas a szórás a kis és a nagy értékek között. A következő kódrészletben pedig már mozgóátlaggal fogunk játszani. Ezek a fajta átlagok alapul szolgálnak a technikai elemzésekhez. Nagy befektető cégek pl. akkor vásárolnak, amikor az árfolyam

felfelé átlépi a 200 napos mozgóátlagot, és akkor adnak el, amikor lefelé átlépi azt. (Beverin-3, 2021)

```
def dickey_fuller(timeseries):
    #Mozgó átlag és mozgó szórás meghatározása
    mozgo_atlag = timeseries.rolling(12).mean()
    mozgo_szoras = timeseries.rolling(12).std()
    #A mozgó átlag és szórás vizualizációja:
    plt.plot(timeseries, color='yellow',label='Eredeti')
    plt.plot(mozgo_atlag, color='blue', label='Mozgó átlag')
    plt.plot(mozgo_szoras, color='green', label = 'Mozgó szórás')
    plt.legend(loc='best')
    plt.title('Mozgó szórás és átlag')
    plt.show(block=False)
    print("Dickey Fuller teszt eredménye: ")
    difu = adfuller(timeseries,autolag='AIC')
    # erdm for dft will give us without defining what the values are.
    #hence we manually write what values does it explains using a for loop
    erdm = pd.Series(difu[0:4],index=['Test Statistics','p-value','No. of lags
used','Number of observations used'])
    for key,values in difu[4].items():
        erdm['critical value (%s)'%key] = values
    print(erdm)
dickey_fuller(zaro_arak)
```

A fenti kódrészlet nem csak a mozgóátlag definiálására szolgált, hanem egyben egy teszt is. Ezt a tesztet Dickey-Fuller-tesztnek is nevezik, arra használják, hogy van-e olyan tulajdonság, amiből statisztikai következtetéseket okozhat egy autoregresszív modellben (egységkört alkot). A teszt H_0 hipotézise: A sorozat egységkört alkot, H_1 : A sorozatnak nincs egységköre. Ennek



12. Ábra: „Amazon 200 napos mozgó átlag és szórás”

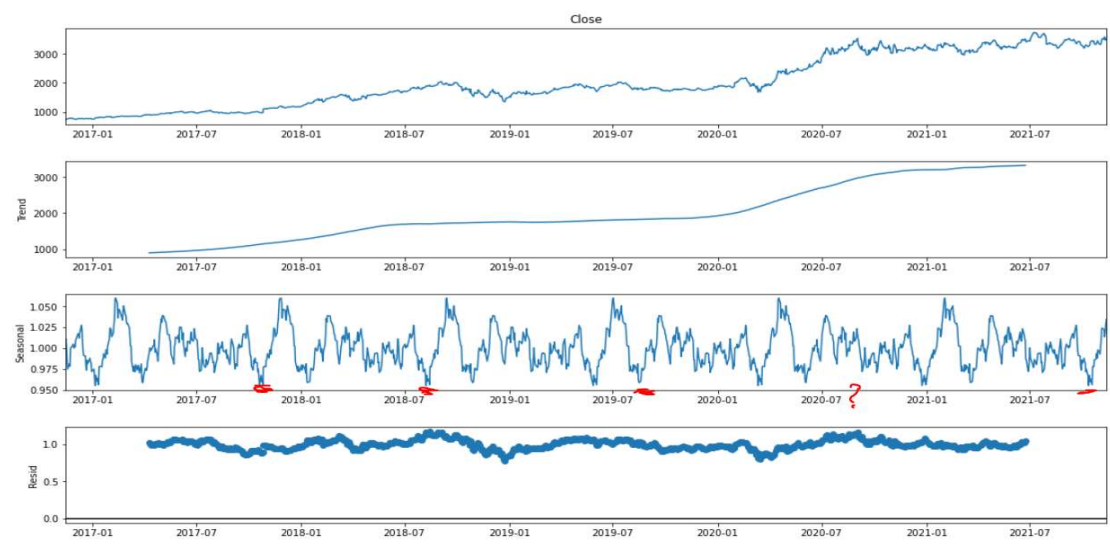
az eredménye látható a 12. ábrán. A teszt több fontos kritikus értéket is megjelenített, de ami a legfontosabb, hogy a teszt szignifikancia szintje 0.901764 volt. Ami azt jelenti az esetünkben, hogy elfogadjuk a null hipotézist - tehát a tesztelt értékek egységkört alkotnak. Különben, ha megnézzük a 200 napos mozgóátlagos kereskedési stratégiát az Amazonra, akkor ezzel is nagyon jól lehetett volna járni 2020. elején. Látható, hogy átlépte a mozgóátlagot az árfolyam ha akkor vettünk volna és egészen a lefelé való átlépésig tartjuk, ami 2021 eleje volt, akkor 1500 dollárról ment volna feljebb a részvény értéke egészen nagyjából 3000 dollárig, ami 100%-os növekedést jelentett volna. A probléma ebben csak az, hogy egy-egy ilyen trend nem tudjuk, hogy meddig tart az időben és ha sokszor veszünk és adunk el, akkor magasak lesznek a tranzakciós költségeink. Térjünk vissza a modellre. (Brownlee, 2017) (Nagy Attila-2, 2021)

A következő lépés a modell építésében, hogy az ilyen kimagasló értékeket, trendeket és szezonálisokat el kell különíteni a modellben, mert ha nem tesszük meg, akkor hibásan fog számolni, illetve kimagasló értékkel nem tud dolgozni ez a fajta modell. Erre a statsmodels könyvtárban van is egy függvény, aminek a neve `seasonal_decompose`.

#Szétválasztom a kimagasló értékeket a modelben

```
eredmenyek = seasonal_decompose(zaro_arak, model='multiplicative', freq = 200)
fig = plt.figure()
fig = eredmenyek.plot()
fig.set_size_inches(16, 9)
```

A fenti kódrészlet megcsinálja a trendek „ki simítását”, aminek az eredménye a görbéken is jól látszik (13. Ábra).



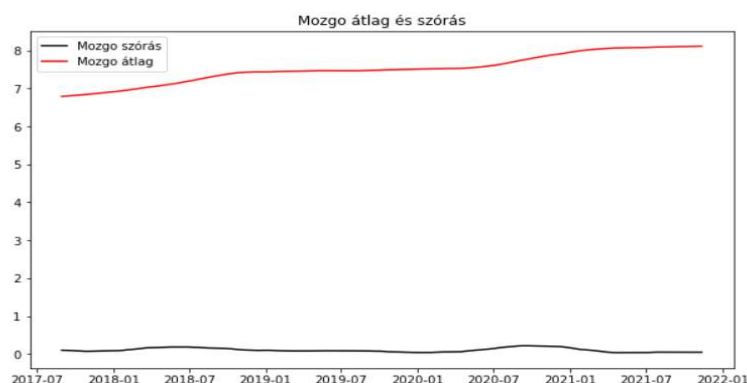
13. Ábra: „Amazon Szezonális kizárása”

Aláhúztam egy pár időszakot és szeretnék megint kitérni egy úgynevezett piaci anomáliára. A tőzsdén megfigyelhető több anomália is, az egyik ilyen a már említett P/E anomália, amit kifejtettem már az [3.3](#) részben, de ezen kívül ismerünk még naptári hatásokat is. A fenti jelölések számomra a Halloween effektust tükrözik vissza, aminek az alapja az, hogy a tőzsdén a november-április időszak, között jóval nagyobb hozamot lehet elérni, mint a május-október időszak alatt. Ezt a megfigyelést láthatjuk vissza köszönni a 13. ábrán is. A probléma ezzel csak az, hogy mivel itt is felmerülnek nagyobb tranzakciós költségek, ezért a hozamunkat ez nagyban árnyalni is fogja. Én még tettem egy kérdőjelet is az ábrára, amin látható, hogy a 2020 évben ez nem volt igaz, nem jött ki a Halloween effektus, mivel épp akkor egy kisebb csődöt éltünk meg a COVID miatt, de ennek hamar vége volt. A csőd alatt a S&P 500 index egészen 2700 pontig esett vissza márciusban. Érdekes lett volna akkor befektetni, hiszen most az S&P 500 az már a 5000 pontot éri. Térjünk vissza a 13. Ábrára - jól látható, hogy a függvény milyen szép munkát végzett a 3-4 ábrákon, most már szinte semmilyen kilengő érték nincs. (Nagy Attila-3, 2021)

#Ez függvény logaritmikus alakara hozza a kapott értékeket.

```
rcParams['figure.figsize'] = 10, 6
log_adatok = np.log(zaro_arak)
mozg_atlag = log_adatok.rolling(200).mean()
szoras = log_adatok.rolling(200).std()
plt.legend(loc='best')
plt.title('Mozgo átlag')
plt.plot(szoras, color="black", label = "Szórás")
plt.plot(mozg_atlag, color="red", label = "Átlag")
plt.legend()
plt.show()
```

A fenti kódrészlet ugyan azért kell, mint előző metódus, azt a funkció tölti be, hogy minden egyes értéknek megkeresi a logaritmikus alakját. Ez azért kell nekünk, mert, mint mondtam, az ARIMA nagyon érzékeny a nagy kilengésekre, eltérésekre, ezért visszavezetjük kis számokra,



14. Ábra: „Értékek logaritmusá”

hogy kicsik legyenek az eltérések, de az értékek között fenntartjuk az arányokat. Megcsináljuk ugyan ezt a szórásra és mozgóátlagra is, majd ennek az eredményét a 14. ábrán látjuk. Jól látható, hogy mind az árfolyam értékek, mind a szórás most már logaritmusos alakban vannak.

```
#Adatok két részre bontása próbára és tesztre
proba_df, teszt_df = log_adatok[3:int(len(log_adatok)*0.8)],
log_adatok[int(len(log_adatok)*0.8):]
plt.figure(figsize=(10,6))
plt.grid(True)
plt.xlabel('Dátum')
plt.ylabel('Záró árak')
plt.plot(log_adatok, 'green', label='Próba adatok')
plt.plot(teszt_df, 'blue', label='Teszt adatok')
plt.legend()
print(len(proba_df))
print(len(teszt_df))
```

A fenti kódrészletben pedig az adataimat bontom két részre - próbára és tesztre. Az adatok 80% lesz próba érték és a maradék 20% pedig teszt. Ez egyszerű szavakkal annyit jelent, hogy az adatok 80%-a alapján próbáljuk megbecsülni majd az adatok 20%-át. Jelen esetben 1004 próba és 252 teszt adatom volt. (Brownlee-2, 2017)

```
model_autoARIMA = auto_arima(proba_df, start_p=0, start_q=0,
                             test='adf',          # adf tesztet használtam a d érték
megkeresésére
                             max_p=3, max_q=3, # definiáltam a max p és q értékeket
                             m=1,             # ezzel a paraméterrel a szezonalitást
lehet beállítani, éves nem szezonális adatoknál alapértelmezett az 1.
```

```

d=None,          # nem adtam meg d értéket, tehát a
program fogja meghatározni
seasonal=False,  # nincs szezonális, azt már
kiszúrtuk.

start_P=0,
D=0,
trace=True,
error_action='ignore',
suppress_warnings=True,
stepwise=True)

print(model_autoARIMA.summary())
model_autoARIMA.plot_diagnostics(figsize=(15,8))
print(model_autoARIMA.get_params())
print(model_autoARIMA.get_params()['order'])
plt.show()

```

A fenti kódrészletben elkészítettem a modellt, ami majd prediktálni fog nekünk. A modellnek a `model_autoARIMA` nevet adtam. A modell a `proba_df` amibe a Yahoo Finance-ből kinyert zárási árakat tettem és még pár lépéssel ez előtt logaritmikus alakra hoztam. A modellt a fenti paraméterek alapján állítottam be, igyekeztem jól kommentálni. Annyit fűznék még hozzá, hogy az `auto_arima` függvény `adf` tesztként ismeri az Augmented Dickey–Fuller tesztet, ezt

```

Performing stepwise search to minimize aic
ARIMA(0,1,0)(0,0,0)[0] Intercept : AIC=10190.755, Time=0.07 sec
ARIMA(1,1,0)(0,0,0)[0] Intercept : AIC=10187.880, Time=0.15 sec
ARIMA(0,1,1)(0,0,0)[0] Intercept : AIC=10188.035, Time=0.59 sec
ARIMA(0,1,0)(0,0,0)[0] Intercept : AIC=10192.412, Time=0.05 sec
ARIMA(2,1,0)(0,0,0)[0] Intercept : AIC=10189.447, Time=0.14 sec
ARIMA(1,1,1)(0,0,0)[0] Intercept : AIC=10187.026, Time=1.19 sec
ARIMA(2,1,1)(0,0,0)[0] Intercept : AIC=10188.887, Time=1.21 sec
ARIMA(1,1,2)(0,0,0)[0] Intercept : AIC=10188.831, Time=1.38 sec
ARIMA(0,1,2)(0,0,0)[0] Intercept : AIC=10189.889, Time=0.66 sec
ARIMA(2,1,2)(0,0,0)[0] Intercept : AIC=10188.798, Time=1.95 sec
ARIMA(1,1,1)(0,0,0)[0] Intercept : AIC=10189.069, Time=0.31 sec

Best model: ARIMA(1,1,1)(0,0,0)[0] Intercept
Total fit time: 7.704 seconds

===== SARIMAX Results =====
Dep. Variable:          y      No. Observations:      1004
Model: SARIMAX(1, 1, 1)      Log Likelihood:    -5089.513
Date: Sun, 14 Nov 2021      AIC:      10187.026
Time: 20:25:42              BIC:      10206.669
Sample: 0                  HQIC:      10194.491
Covariance Type:          opg

=====
coef    std err      z    P>|z|    [0.025    0.975]
-----
Intercept    3.7543    1.896    1.981    0.048    0.039    7.469
ar.L1      -0.5960    0.144   -4.137    0.000   -0.878   -0.314
ma.L1       0.5252    0.154    3.421    0.001    0.224    0.826
sigma2     1496.3911   34.798   43.003    0.000  1428.189  1564.593
=====
Ljung-Box (Q):      115.12    Jarque-Bera (JB):      1251.36
Prob(Q):           0.00      Prob(JB):           0.00
Heteroskedasticity (H): 13.24      Skew:           0.14
Prob(H) (two-sided):  0.00      Kurtosis:          8.47
=====

```

15. Ábra: „Auto ARIMA eredmények”

használtuk már korábban. A teszt segítségével fogjuk megkeresni a megfelelő paramétert a `d` értéknek. A 15. Ábrán látható a modell kalkulációja és eredménye is. Látható, hogy a modell kiszámolta a függvény értékeit minden egyes paraméter változás esetén. Az a legjobb modell, aminek a függvény értéke a legalacsonyabb – a mi esetünkben ez (1,1,1) volt, de ez minden modellnél, részvényenél más lesz. Miután megkerestük a legjobban illeszkedő modellt az adatainkra, létrehozunk egy osztályt, ami úgymond ezeken az adatokon tanult.

```

#A legjobb modell létrehozása
model = ARIMA(proba_df, order=model_autoARIMA.get_params()['order'])
fitted = model.fit(dis=-1)
#n nap létrehozása
#A teszt_dfben talált utolsó dátum_indexhez hozzá adunk n napot
vegso_datum = teszt_df.last_valid_index() + datetime.timedelta(days=30)
#létrehozunk egy új indexet a teszt_df első indexétől az utolsó + n napig
datum_30 = pd.date_range(start = teszt_df.first_valid_index(), end =
vegso_datum, freq = '1D').strftime('%Y-%m-%d')
#létrehozunk egy listát az új index méretével
uj_tesztadatok_bov=[]
for szam in range(len(datum_30)):
    uj_tesztadatok_bov.append(szam)
#a listából datafremat csinálunk
bov_df = pd.DataFrame(uj_tesztadatok_bov)
#beállítjuk a generált dátum indexet a listának
bov_df = pd.DataFrame(bov_df,index=datum_30)
#sorozatot csinálunk a listából.
bov_df=bov_df.squeeze()
#Előrejelzett adatok
elorejelzett_arfolyamok1,elorejelzett_arfolyamok2,elorejelzett_arfolyamok3=fitted.forecast(len(bov_df), alpha=0.05)
#pandas sorozat csinálása a dict-ből
elorejelzett_arfolyamok = pd.Series(elorejelzett_arfolyamok1,
index=bov_df.index)
print('Az előrejelzés átlaga: ' + str(elorejelzett_arfolyamok1.mean()))
#Predikált értékek kiírása
print(elorejelzett_arfolyamok.tail(30))

```

Ebben a kódrészletben létrehoztam a pdq paramétereknek megfelelő modellt, amit majd prediktálásra fogok használni. Mivel már ezeket a paramétereket kiszámoltam a model_auto_ARIMA-ban, ezért csak át kell adnom a model_autoARIMA.get_params()['order']). A modell eredménye egy úgynevezett *dictionary* adattípusba van lementve, itt az adatok kulcs-érték párokba vannak rendezve, és jelen esetben a (p,d,q) sorrendet a 'order' kulcs alatt találjuk. A kommentek remélem beszédeselek, pár szóban összefoglalva létrehoztam egy új olyan sorozatot, ami n (jelen esetben 30) nappal nagyobb, mint az előző, és átadtam a modellnek, hogy predikálja meg az értékeket. (Brownlee-3, 2017)

```

#beírom listába az összefűzni kívánt df-eket
l1=[proba_df,elorejelzett_arfolyamok]
#összefűzöm a próba és előrejelzett értékeket
ujertekek = pd.concat(l1)
#tölcsér megcsinálása 95% konfidencia szinthez
lower_series = pd.Series(elorejelzett_arfolyamok3[:, 0], index=bov_df.index)
upper_series = pd.Series(elorejelzett_arfolyamok3[:, 1], index=bov_df.index)

```

```

# Diagram szerkesztése
plt.figure(figsize=(10,5), dpi=500)
plt.plot(ujertek , label='Tanulási adathalmaz')
plt.plot(teszt_df, color = 'blue', label='Aktuális ár')
plt.fill_between(lower_series.index, lower_series, upper_series, color='red',
alpha=.1)
plt.axhline(y=elorejelzett_arfolyamok1[len(elorejelzett_arfolyamok1)-1], color
= 'red', linestyle = ':', label = 'Előrejelzett érték:
{0}'.format(round(elorejelzett_arfolyamok1[len(elorejelzett_arfolyamok1)-1])))
plt.title(szimbolum+ ' Árfolyam előrejelzés')
plt.xlabel('Idő')
plt.ylabel('Értékpapír ár')
plt.legend(loc='upper left', fontsize=8)
plt.grid()
plt.show()

```

4.1.1. ARIMA EREDMÉNYEK

Ebben a kódrészletben pedig összeillesztem a próba, és a predikált adatokat. A többi pedig már a vizualizációnak a része, aminek az eredménye a 16. ábrán látható.



16. Ábra: „Amazon árfolyam előrejelzés”

Sajnos a grafikonról nehezen lehet leolvasni, de modell alapján 2021-11-22 : 3995 USD lesz a amazon árfolyama. Az már az ábráról is leolvasható, hogy az aktuális árfolyam jóval a előre jelezett árfolyam alatt van, ez gondolom, hogy a jövőben is így marad. Ezt november 22-én leellenőriztem de pont ekkor egy nagyobb korrekció volt így az árfolyam 3572 USD volt, de az ár is benne volt a 95% konfidenciaszintben. De nagyon sok múlik a modell alapbeállításain is, illetve azon, hogy milyen időszakban nézzük az adott részvényt. Mivel én 5 éves intervallumban tanítottam a programot, ezért a program az gondolja, hogy a következő 10 napra

is hatalmas növekedése lesz az árfolyamnak. Főként azért, mert ha megnézzük a 2020 elejétől 2021-ig, akkor látható, hogy majdnem 1000 USD-vel növekedett az árfolyam. De ha a programot az elmúlt egy év alapján tanítom, akkor már más a helyzet.



17. Ábra: „Amazon árfolyam előrejelzés elmúlt egy év alapján”

A 17. ábra már jóval pontosabb, mivel tanult adatoknál sem volt olyan nagy kilengés. Ezek az adatok alapján a mai (2021.11.15) napra az előre jelzett érték 3619 USD, ami jóval közelebb van a 3,525.15-hoz - ez sem pontos, de az árfolyam növekszik, közeledik a trendvonalhoz. E szerint a modell szerint az árfolyam 2021-11-22: 3672 USD lesz. Az is látható, hogy jóval nagyobb a 95% konfidenciaszint (rózsaszínnel jelölve) és az is, hogy az árfolyam ebben a konfidenciaszintben mozog. Ebből a modellből igazából azt tudjuk levonni, hogy a trend vonala éppen hogyan néz ki, és az látszik, hogy a jövőben is növekedésre számíthatunk. Ha a modelletem úgy állítom be, hogy prédikálja meg a következő 300 napot, akkor azt kapom, hogy 2022-09-08: - 4184.364947 USD lesz az árfolyam, jelenleg a Stock Analysis 47 elemzője szerint pedig 4066, 41 USD lesz egy év múlva, ami elég közel van a ARIMA modellel prediktált értékhez, nagyjából 100USD az eltérés, ami szerintem elég jó. Részletesen megvizsgáljuk még egy másik MI-val is, ami a neurális hálón alapszik. Annak a modellnek pedig RNN (Recurrent Neural Network) a neve.

4.2. RNN MODELL

Az RNN egy olyan neurális háló, ahol az előző lépés eredménye input lesz a következő lépésnek, szemben egy hagyományos hálóval ez hatalmas különbség, hiszen annál a kimenetek

és a bementeket teljesen függetlenek egymástól. Tehát az RNN egy speciális neurális hálózat, ahol az eredmények beépülnek a következő háló inputjába. Ez akkor hasznos, amikor sorozatokat elemzünk, manapság gyakori ez a fajta modell, hiszen használják szöveg, beszéd és kép felismeréshez is. A RNN fő jellemzője, hogy van benne egy olyan rejtett háló, ahol az előző sorozat eredményeire emlékszik a modell - ezt a részt nevezzük az RNN memóriájának. A memóriáról annyit kell tudni, hogy csak az előző outputot tudja megjegyezni, ezért nagy sorozatoknál ez nem használható. Az RNN-nek az egyik típusa az úgynevezett Long Short Term Memory, ami kiküszöböli ezt a problémát. Ez a fajta RNN modell hosszabb távon képes az adatokat eltárolni, illetve könnyedén programozható arra, hogy bizonyos eredményeket elfelejtsen. Ez nagyon jól jön az értékpapírok előrejelzésénél hiszen, ha végiggondoljuk azt, hogy a vállalatok a korai fázisban még gyorsan növekednek (éven belül akár 100% növekedés sem ritka a jó cégeknél), de amikor már nagy a vállalat, akkor már éven belül a 10% is soknak számít. (Ganegedara, 2020). Lássunk is hozzá a modell létrehozásához:

```
#pip install yfinance
import yfinance as yf
import numpy as np
from sklearn.preprocessing import MinMaxScaler
#pip pip install keras
#pip install tensorflow
from keras.models import Sequential
from keras.layers import Dense, LSTM
import matplotlib.pyplot as plt
print('Modulok betöltődtek')
szimbolum = 'AMZN'
#Adatok letöltése a megadott intervallumra
adatok = yf.download(tickers=szimbolum,period='5y',interval='1d')
type(adatok)
adatok.head()
```

A fenti könyvtárak lesznek szükségesek a programhoz, illetve szintén az Amazon részvény példáján fogom bemutatni a programot. (Ganegedara-2, 2020)

```
#A nyitási adatok áttöltése nyitar változóba
nyitar = adatok[['Open']]
#Adatok áttöltése egy listába
ls = nyitar.values
print(ls)
plt.plot(ls)
```

A továbbiakban lista adattípussal fogunk dolgozni, ezért át kellett konvertálni a változót.

```

#Adatok normalizálása, hogy minden adat 0 és 1 közzé essen az adatok között a
#változás megmaradjon.
#Ez azért kell, hogy a modell tanulni tudjon, mivel ha az adatok
#intervallumban mozognak akkor megzavarja a tanulási folyamatot
normalizalt = MinMaxScaler(feature_range=(0,1))
ls_lista = normalizalt.fit_transform(np.array(ls).reshape(-1,1))
print(len(ls_lista), len(ls))
#Megadjuk a proba és a teszt halmazok hosszát
proba_h = int(len(ls_lista)*0.70)
teszt_h = len(ls_lista) - proba_h
proba_h,teszt_h

```

A fenti kódrészletben megképeztük a próba és teszt halmazoknak a hosszát.

```

#Ketté vágjuk tanító és teszt adatokra a nagy listát
ls_proba, ls_teszt = ls_lista[0:proba_h,:], ls_lista[proba_h:len(ls_lista),:]
len(ls_proba),len(ls_teszt)
#Ez a függvény azt csinálja, hogy szétválogatja az adathalmazt két részre:
#próba részre és tesztre.
def gen_ls(adathalmaz,lepes):
    Xproba, Yproba = [], []
    for i in range(len(adathalmaz)-lepes-1):
        a = adathalmaz[i:(i+lepes), 0]
        Xproba.append(a)
        Yproba.append(adathalmaz[i + lepes, 0])
    return np.array(Xproba), np.array(Yproba)

```

Csinálok két listát a megadott hosszakkal és a következő lépésben pedig a gen_ls függvénnyel feltöltöm a listákat.

```

#Feltöltöm a proba és teszt adathalmazokat a függvénnyel
#Ez azt jelenti, hogy 100 adatból fog egyszer prediktálni, ez lesz a model
lépésköze. A modell 100 rekordot fog egynek nézni
idokoz = 100
X_proba, y_proba = gen_ls(ls_proba,idokoz)
X_teszt, y_teszt = gen_ls(ls_teszt,idokoz)
print(X_proba.shape,y_proba.shape)
print(X_teszt.shape, y_teszt.shape)

```

Miután megvannak a listáink megfelelő formára kell alakítani, amit majd át tudunk adni a modellnek.

```

#az adatokat megfelelo formaba alakitjuk
X_proba = X_proba.reshape(X_proba.shape[0],X_proba.shape[1] , 1)
X_teszt = X_teszt.reshape(X_teszt.shape[0],X_teszt.shape[1] , 1)
#Modell létrehozása
model = Sequential()
#Első LSTM háló létrehozása, aminek az indupt adata a X_proba

```

```

model.add(LSTM(units=50,return_sequences=True,input_shape=(X_proba.shape[1],1)
))
#Kiszűröm az adatok 0.2%kát a modellel(pontosítok)
model.add(Dropout(0.2))
#2. LSTM háló hozzáadása
model.add(LSTM(units=50,return_sequences=True))
model.add(Dropout(0.2))
#3. LSTM háló létrehozása az előző paraméterek szerint
model.add(LSTM(units = 50, return_sequences = True))
model.add(Dropout(0.2))
#4. LSTM háló
model.add(LSTM(units = 50))
model.add(Dropout(0.2))
model.add(Dense(units=1,activation='linear'))
model.summary()

```

A fenti kódrészletben már a modellt hoztam létre, 4 LSTM hálót adtam a modellhez és minden egyes lépésnél csökkentem bizonyos neuronok hatását 20%-al. Ezt azért szokták csinálni, hogy a nagy kilengéseket elhagyjuk a modellből. A Dense hálóban pedig megképezzük az eredményeket. (Ganegedara-3, 2020)

```

#Adatok tanulása
model.compile(loss='mean_squared_error',optimizer='adam')
model.fit(X_proba,y_proba,validation_data=(X_teszt,y_teszt),epochs=100,batch_size=64)
#Tanulási eredmény vizsgálata
loss = model.history.history['loss']
plt.plot(loss)

```

A modell alapján elindítjuk a tanulást. Ez egy kis időt igénybe vesz, minél nagyobb az adatmennyiségünk, annál több ideig fog futni a program.

```

#Előrejelzés a proba és teszt adatokra
proba_predict = model.predict(X_proba)
teszt_predict = model.predict(X_teszt)
#Vissza konvertáljuk a normalizált adatokat, a szebb vizualizációért
proba_predict = normalizalt.inverse_transform(proba_predict)
teszt_predict = normalizalt.inverse_transform(teszt_predict)

```

A fenti kódrészlet segítségével megcsináljuk az előrejelzést.

```

#Prediktált és a próba lista egyesítése
test = np.vstack((proba_predict,teszt_predict))
plt.plot(normalizalt.inverse_transform(ls_lista),color='blue',label='Eredeti árfolyam')
plt.plot(test,color='red',label='Teszt adat')
plt.legend(loc='best')

```

```
print(len(ls_teszt))
```

Itt pedig, mint a komment is jelzi, összerakjuk a prediktált és a próba listát. (Loukas, 2020)

```
#Előrejelezni a legutolsó 100 adatotból. Ha itt többet akarunk akkor lehet, de
#a While ciklusnál szintén kell
elorejelzes = ls_teszt[len(ls_teszt)-100:]
elorejelzes = elorejelzes.reshape(1,-1)
ideglenes = list(elorejelzes)
elorejelzes.shape
#Konvertálás listába
ideglenes = ideglenes[0].tolist()
```

Most a legutolsó 100 adatból megpróbáljuk előrejelezni a következő n napot.

```
#Előrejelezni a következő 30 adatot, 100 adatból
lst_er=[]
el_jel_hosz=100
n=30
i=0
while(i<n):

    if(len(ideglenes)>100):
        elorejelzes = np.array(ideglenes[1:])
        elorejelzes=elorejelzes.reshape(1,-1)
        elorejelzes = elorejelzes.reshape((1, el_jel_hosz, 1))
        yhat = model.predict(elorejelzes, verbose=0)
        ideglenes.extend(yhat[0].tolist())
        ideglenes = ideglenes[1:]
        lst_er.extend(yhat.tolist())
        i=i+1
    else:
        elorejelzes = elorejelzes.reshape((1, el_jel_hosz,1))
        yhat = model.predict(elorejelzes, verbose=0)
        ideglenes.extend(yhat[0].tolist())
        lst_er.extend(yhat.tolist())
        i=i+1
print(lst_er)
```

A fenti kódrészlettel már a következő n napot fogjuk prediktálni.

```
#Uj diagram készítése az adatok megjelenítéséhez, így lehet majd össze
csusztatni az adatokat
plot_new=np.arange(1,101)
plot_pred=np.arange(101,len(lst_er)+1)

plt.plot(plot_new, normalizalt.inverse_transform(ls_lista[len(ls_lista)-
100:]),label='Eredeti adatok')
```

```
plt.plot(plot_pred, normalizalt.inverse_transform(lst_er), label='Előrejelzett  
adatok')  
plt.legend(loc='best')
```

A fenti kódrészlet arra való, hogy megjelenítse a prediktált és teszt adatokat, de ezen még van csúsztatás. A következő kóddal ezt orvoslom.

```
#A előrejelzett és az eredeti adatok össze csúsztatása  
uj_df = ls_lista.tolist()  
uj_df.extend(lst_er)  
plt.plot(uj_df[len(ls_lista)-100:])
```

Ennél a kódrészletnél a komment eléggé beszédes. A dolgozatba már csak a végső eredményt fogom beilleszteni, amit ezzel a kódrészlettel állítok elő (Loukas-2, 2020):

```
#Végső grafikon, adatok vissza transzformálása eredeti értékekre  
final_graph = normalizalt.inverse_transform(uj_df).tolist()  
print(normalizalt.inverse_transform(lst_er))  
print(len(lst_er))  
#Adatok megjelenítése  
plt.figure(figsize=(10,5),dpi=500)  
plt.plot(final_graph, label='Árfolyam')  
plt.ylabel("Árak")  
plt.xlabel("Idő")  
plt.title("{0} árfolyam előrejelzés".format(szimbolum))  
plt.axhline(y=final_graph[len(final_graph)-1], color = 'red', linestyle = ':',  
label = 'Várható árfolyam:  
{0}'.format(round(float(*final_graph[len(final_graph)-1]),2)))  
plt.legend()
```

4.2.1. RNN EREDMÉNYEK

A RNN modellünk eredménye tehát a 18. Ábrán látható. Ez a modell kissé pesszimistább,



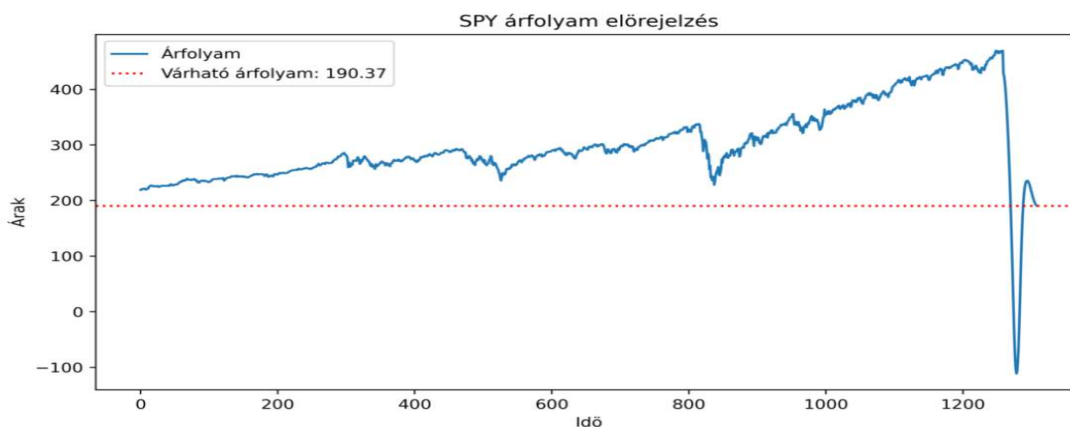
18. Ábra: „Amazon árfolyam előrejelzés elmúlt 100 nap alapján”

hiszen november 22.-re 3564 USD árfolyamot mondd. Míg a +30 napos előrejelzés alapján december 17.-én 3657 USD lesz az árfolyam. Ami még benne van az előző ARIMA modell konfidencia szintjében is. A két modell között a nagy különbség, hogy az ARIMA-ban egy lineáris előrejelzés van, itt pedig hullámszerű. Így ezzel akár trendeket is előre jelezhetünk. Nézzük meg ezekkel a modellekkel a többi részvényt is.

4.3. RÉSZVÉNY ELEMZÉK

4.3.1. S&P500 INDEX

Nézzük meg akkor a többi részvény előrejelzését is ezekkel a modellekkel, és még a dolgozatom elején ígértem, hogy megpróbáljuk az S&P 500 index értékének az előrejelzését.



19. Ábra: „S&P500 Index követő ETF előrejelzés, RNN”

Kezdjük is ezzel, hiszen ez egy részvény kosár, és a teljesítménye nagyban meghatározza a globális részvénytőzsi hangulatot. Érdekes eredményeket kaptam. Az RNN modellel próbáltam 5 év alapján előre jelezni a következő 50 napot. Az eredménye a 19. Ábrán látható.

Ezen a görbén jól látszik, hogy mennyire nem mindegy, hogy hogyan tanítjuk a hálót. Itt most ez hibásnak tűnik, mert negatív értékeket is előre jelzett, de az mindenképpen érdekes, hogy 190 USD árat jelzett elő. Ezért látszik, hogy nagyon nem mindegy, hogy mi lesz a tanuló, és mi lesz a teszt halmaz. Lefuttattam hasonló konfigurációkkal az ARIMA modellt is, ami szintén



20. Ábra: „S&P500 Index követő ETF előrejelzés, ARIMA”

egy korrekciót jelzett elő, ez a 20. ábrán látható. Megfigyelhető, hogy az alsó konfidencia szintje egészen 200 USD körül mozog, ebbe a sávba esik az LSTM modell előrejelzése is, ami szintén érdekes. Már csak az a kérdés, hogy lehetséges-e ez a hatalmas korrekció, vagy ilyen nagy hiba van a modellekben. Utánanéztem az adatoknak, kapóra jött, hogy nemrég kezdtek megjelenni a különböző beszámolók a nagy guruktól is. Warren Buffett és Charlie Munger ebben a negyedévben nagyon kevés részvényt vásárolt (azt is inkább pánik elkerülés végett) és inkább eladták a részvényeiket. Ez számunkra is azt jelenti, hogy nem most a 2021 év végén kell részvényt venni, hanem inkább eladni kell, és ez azt is jelenti, hogy ők egy kisebb korrekcióra számítanak. A kérdés már csak az lesz, hogy mekkora lesz a korrekció és mennyi ideig lesz medvepiac. Ehhez készítettem egy technikai elemzést is, ami a 21. ábrán látható. A



21. Ábra: „S&P500 Index Technikai elemzés”

2021 évre berajzoltam egy növekedési csatornát piros színnel. Felvettem még pár fontos indexet és mozgó átlagokat lilával a 200 napost, zölddel az 50 napost. A fenti alakzatokról, RSI-ről, MACD-ről annyit kell tudni, hogy ezeket mind indikátoroknak használják. Kékkel jelöltem egy úgynevezett ék alakzatot, ami egy trendfordító alakzat. Jól látható, hogy most már e körül táncol az árfolyam, ha beesik az árfolyam a piros növekedési csatorna alá akkor valószínűleg egy nagyobb korrekció is lesz, ehhez a csatorna nagyságát szokták venni, de ez akár a csatorna nagyságának a 2x 3x is lehet, amit lilával jelöltem. Ezt mutatja az RSI is, ami a 80.000 értéket súrolt és ez szintén egy erős eladási jel. Ugyan ez a helyzet a MACD indexnél is: ott jól látható, hogy a kék vonal fent keresztezte a narancssárga vonalat. Ezek mind amellet érvelnek, hogy most már eladni kell, és lekéstünk a nagy hozamokról.

4.4. TESLA

Nézzünk meg a Tesla részvényére a modellt, ami a 22. Ábrán látható. Engem elképeszt teljesen: akkorát emelkedett az elmúlt 1 évben hogy a még a programom 95% konfidencia szintjéből is

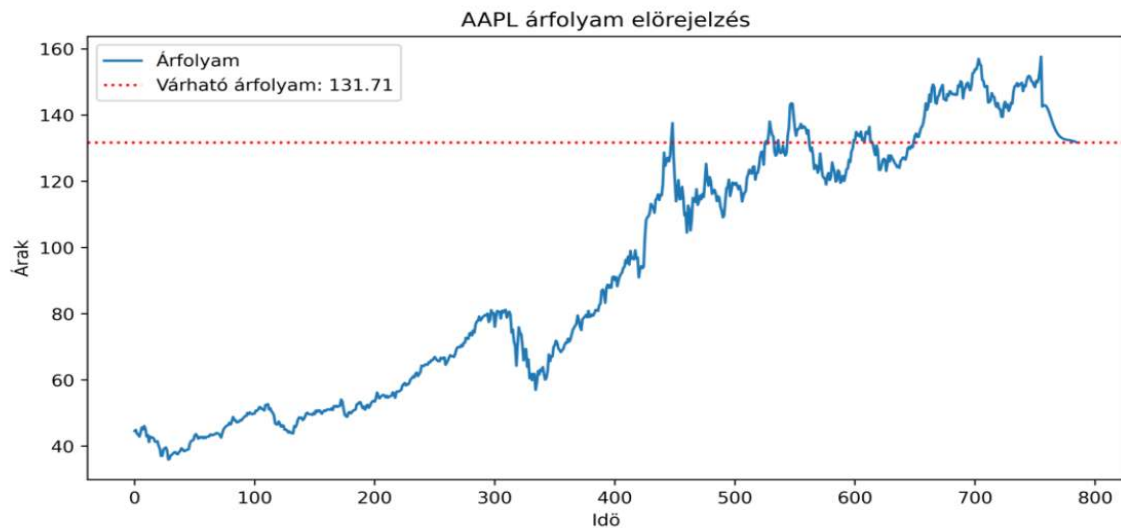


22. Ábra: „Tesla előrejelzés, ARIMA”

kilépett. Ez egy könnyen orvosolható hiba, azért fordul elő, mert a cégek kezdetben lineárisan majd exponenciálisan kezdenek növekedni, és ezt a lineáris modell nem annyira tudja értelmezni. Úgy kell ezért az adatokat behívni, hogy a leírt árfolyam görbe nagyjából lineáris legyen. Ezt meg is tettem, átállítottam a programot, hogy csak az elmúlt 2 évet nézze. Még így is kicsit kilépett a teszt az aktuális árfolyam konfidencia zónájából, de itt már elfogadható ez a kiugrás. Ez látható a 22. ábrán is. A jelenlegi árfolyam valahol 1137 körül mozog, tehát hosszú távon egy korrekciót vár a program is az árfolyamtól, ugyan ezt mondta a másik LSTM modell is, de ott a korrekció egészen 900 USD árfolyamra való ráállás lesz, ami még mindig a 95% konfidencia zónába esik.

4.5. APPLE

Ennél a részvélynél szintén mindkét program azt várja, hogy a jövőben csökkenni fog az árfolyama, itt az ARIMA modell az várja, hogy a következő 30 napban körülbelül 154 dollárra fog esni az árfolyama az Applenek, de elég nagy itt a konfidenciaszint - egészen 110 visszaeshet. Az LSTM szerint 30 nap múlva 131 USD lesz.



23. Ábra: „Apple előrejelzés, LSTM”

5. ÖSSZEGZÉS

A dolgozatom azzal kezdtem, hogy megpróbálok majd olyan programokat írni, amik segítenek eldönteni, hogy mibe fektessem a félretett pénzem. Úgy gondolom, hogy sikerült ezt a témát eléggé körbejárni, egészen statikus pénzügyi elemzésektől, mesterséges intelligencia programokig eljutottunk. A [3.](#) fejezetben készített program Excel fájlja már tartalmazza azon részvények adatait, amelyek elég jó pozícióban vannak piaci kapitalizáció, és ár/érték arány alapján. Számomra azért rendkívül hasznos ez az „adatbázis”, mert rengeteg olyan részvényt hozott a perifériámba, amiről eddig még csak fogalmam se volt, hogy jó befektetés lenne. A szűrők segítségével 5500 részvényből egy elmélet alapján sikerült 300 részvényt találni, ami jó vétel lehet. Köztük: Toyota, P&G, Philip Morris, Verizon, T-Mobil, AT&T, Veeva Systems, Alibaba, csak, hogy párat soroljak. Úgy ez az elmélet nem más, mint Warren Buffet és Benjamin Graham befektetési stratégiája az „érték alapú” befektetés. Különben Graham nem csak erre tér ki könyvében, hogy hogyan lehet a jó részvényeket megtalálni, de arra is, hogy mennyi legyen a befektetésünk korhoz, életkörülményekhez képest. Szintén a könyvében azt is tanácsolja, hogy ne tegyük minden pénzünket részvénybe, mert vannak olyan medvepiacok, amik évekig kitartanak, és azt nem biztos, hogy képes vagyunk túrni, hogy a pénzünk 30-40% eltűnik, ezért is tanácsolja azt, hogy maximum 60-70% legyen a pénzünknek részvényben, attól függően, hogy mennyire szeretjük a kockázatot, a többi az legyen állampapír.

A másik program, amit használtam - azok az előrejelző algoritmusok voltak. Ezt azért használtam, mert biztos akartam lenni magamban, hogy azok a részvények, amik bekerültek (vagy épp ki - lásd APPLE, AMZN, TSLA), azok egy másik „szemlélet mód” alapján is jók-e. Párat a „legjobb részvények” közül betettem a MI modellekbe és azt kaptam, hogy mind előre jelzett ár alatt van. Ami visszaigazolja azt is, hogy ezek jó áron vannak, másrészt azt is, hogy a jövőben növekedni fognak. Az szintén egy jó visszajelzés volt számomra, hogy más elemző cégek szerinti, az egy éven belüli árfolyam változás nagyjából egyezett az én előrejelzett értékeimmel. Illetve sokat kutakodtam ezenfelül is előrejelzésekért, és nagyon közeli értékeket kaptam minden más fórumon az én értékeimhez. Ezenfelül még elsajátítottam pár technikai elemzési alapot is, hogy visszaigazoljam ezeket az adatokat, majd azok is megerősítettek engem ezekben az állításokban, és egy visszacsatolást tudtak adni a munkámról. A másik ilyen, amit szintén nagy sikernek élek meg, hogy az előrejelzett értékeket közel visszakaptam az idő múlásával. Például készítettem egy előrejelzést az Amazonra, november 15-én, ahol november 22-re 3672 volt, és árfolyam november 22 ténylegesen így alakult. Engem ezek az adatok

nagyon megleptek, nem gondoltam, hogy ilyen pontosan lehet előrejelezni rövid távon, bár azt meg kell hagyni, hogy a konfidenciaszintek igen szélesek. Ami igazából tényleges probléma, hogy nem tudja kezelni a modell a piaci változásokat, tehát a modellekben a világ statikus, nincs válság, nincs külső információ. Ha esetleg most válság lenne a és részvények lejtmenetbe kapcsolnának, akkor ezt a modell biztosan nem látná. Arra viszont jó, hogy egy trendet meghatározzon, viszont a trendfordulásokra nekünk biztosan figyelniünk kell. A technikai elemzésekből ezek nagyjából láthatóak is.

Szerintem a teljes rendszeremnek nagy hátránya, hogy nem nézi a lehetőségeket, csak a múltba tekint és az alapján próbál jóslni. Sok esetben viszont így le lehet késni jó fogásokról. Ezt, szerintem, sajnos nem lehet semmilyen automatizmussal kiváltani, hiszen itt akár technológiai fejlettségről, akár vezetői képzettségről, rátermettségről van szó, csak sok kutatás árán lehet azokról meggyőződni. Ezek a tényezők szintén hatalmas befolyással vannak a részvény árfolyamára, hiszen azt az esetet, hogy Steve Jobs hogyan hozta vissza a csőd szeléről az Apple-t mindenki ismeri. Hogyha jó befektetéseket akarunk, akkor mindenképpen ilyen fajta szempontokra is figyelniünk kell.

Sajnos azt a kérdést, hogy mibe fektessem a pénzemet 2021 végén, nagyon egyszerűen meg lehet válaszolni - rövid lejáratú állampapírokba. A kutatásom során nagyon hamar világossá vált, hogy nem csak az Apple vagy a Tesla van túlárázva, de az egész piac is. Ezt a MI algoritmusok is látják, hiszen mindegyik egy erős korrekciót jelez előre. Szintén ezt lehet kiolvasni technikai elemzésekből és ha követjük Buffet befektetéseit, azok is erről árulkodnak. Hiszen ő is most csak készpénzt halmoz és részvényt ad el túlsúlyban, csak a saját részvényeit veszi vissza. Úgyhogy az én stratégiám is ez lesz: igyekszek minél több megtakarítást félre tenni, azokat kötvénybe akár MÁP+-ba és megfelelő időben pedig ezt a pénzt átcsoportosítom majd részvénybe. A módszer, ami alapján megteszem majd az átcsoportosítást már kész van. Ezt a korrekciót már több befektető is várja és ez valószínűleg a jövő év elején-közepén meg is fog történni. Azt azért szeretném kiemelni, hogy dolgozatom az aktualitását az idő múlásával elveszti. Korrekció után bármi jó lehet ismét. Akkor azt kell figyelni, hogy hogyan fognak olyan cégek árfolyamai mozogni, amik jelenleg túlárázottak. Korrekció után valószínűleg már nekem is lesz Tesla részvényem, úgyhogy én már nagyon várom azt az időt, amikor ezeket a módszertanokat bevethetem élesben is. Addig pedig az értékalapú befektetést szeretném részletesen kutatni, illetve el szeretnék jobban mélyülni technikai elemzésben is.

IRODALOMJEGYZÉK

- Benjamin Graham: Az intelligens befektető. Törökbálint: T.bálint kiadó, 2011.
Letöltve: 2021.11.20
- Dr. Aradi, Petra, Gräff, József, Dr. Lipovszki, György: Számítógépes szimuláció, Budapest, BME MOGI, 2015.
http://www.mogi.bme.hu/TAMOP/szamitogepes_szimulacio/ch04.html
Letöltve: 2021.11.20
- Dr. Bíró, T – Kresalek, P – Dr. Pucsek, J – Dr. Sztanó, 2012: A vállalkozások tevékenységének komplex elemzése, Budapest, Perfekt, 2012
Letöltve: 2021.11.20
- Dr. Musinszki Zoltán, 2017: Hagyományos pénzügyi mutatószámok. <https://gtk.unimiskolc.hu/files/11762/hagyom%C3%A1nyos+p%C3%A9nz%C3%BCgyi+mutat%C3%B3k.pdf>
Letöltve: 2021.11.20
- Farkas Richárd, 2021: GÉPI TANULÁS A GYAKORLATBAN. https://www.inf.u-szeged.hu/~rfarkas/ML20/deep_learning.html
Letöltve: 2021.11.20
- Hegyeshalmi Richárd, 2018: A mesterséges intelligencia jobb a bőrrák felismerésében, mint a szakorvos. https://index.hu/tudomany/egeszseg/2018/05/29/a_mesterseges_intelligencia_jobb_a_borrak_felismereseben_mint_a_szakorvos/
Letöltve: 2021.11.20
- Jason Brownlee, 2017: How to Create an ARIMA Model for Time Series Forecasting in Python. <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>
Letöltve: 2021.11.20

- Luka Beverin, 2021: Introduction to Time Series Forecasting of Stock Prices with Python: <https://www.datadriveninvestor.com/2020/07/07/introduction-to-time-series-forecasting-of-stock-prices-with-python/>
Letöltve: 2021.11.20
- Nagy Attila, 2021: A 200 napos mozgóátlag: Hogyan használd, mi a gond vele? <https://elemzeskozpont.hu/200-napos-mozgoatlag-hogyan-hasznald-mi-gond-vele>
Letöltve: 2021.11.20
- Nagy Attila, 2021: Halloween-effektus, halloween-stratégia: Van értelme használni? <https://elemzeskozpont.hu/halloween-effektus-halloween-strategia-van-ertelme-hasznalni>
Letöltve: 2021.11.20
- Nagy Attila 2021: P/E ráta, mutató használata, hatásosságát bizonyító 3 tanulmány áttekintése. <https://elemzeskozpont.hu/pe-rata-mutato-hasznalata-3-tanulmany-eredmenye>
Letöltve: 2021.11.20
- Rajczy István 2019: MESTERSÉGES INTELLIGENCIA. https://www.parlament.hu/documents/10181/1789217/Infojegyzet_2019_11_mesterseges_intelligencia.pdf/6ec90247-a26c-30ed-be63-c4e3f052b835
Letöltve: 2021.11.20
- Rebecca Reynoso, 2019: 4 Main Types of Artificial Intelligence. <https://www.g2.com/articles/types-of-artificial-intelligence>
Letöltve: 2021.11.20
- Stuart Russell, Peter Norvig: Mesterséges Intelligencia Modern megközelítésben, Budapest, Panem kiadó, 2005. <http://mialmanach.mit.bme.hu/aima/ch01s03>
Letöltve: 2021.11.20
- Serafeim Loukas, 2020: Time-Series Forecasting: Predicting Stock Prices Using An LSTM Model.

<https://towardsdatascience.com/lstm-time-series-forecasting-predicting-stock-prices-using-an-lstm-model-6223e9644a2f>

Letöltve: 2021.11.20

- Szendrei Ádám, 2021: ETF Útmutató: Befektetés olcsón, passzív alapokon keresztül.

<https://szendreiadam.hu/befektetes/etf-passziv-befektetesek/>

Letöltve: 2021.11.20

- Takács András: Vállalatok értékelése. Budapest: Akadémiai Kiadó, 2009.

https://saldokiado.hu/assets/res/download/_vallalatertekeles-magyar-szamviteli-kornyezetben.pdf

Letöltve: 2021.11.20

- Takács András: Modern vállalatok értékelése, Budapest: Akadémiai Kiadó, 2021.

https://mersz.hu/dokumentum/m827mv_1/

Letöltve: 2021.11.20

- Thushan Ganegedara, 2020: Stock Market Predictions with LSTM in Python:

<https://www.datacamp.com/community/tutorials/lstm-python-stock-market>

Letöltve: 2021.11.20

- www.allampapir.hu, 2021: Állampapírok.

<https://www.allampapir.hu/allampapirok/>

Letöltve: 2021.11.20

- www.computerworld.hu, 2015: A foci vébé technológiáját kínálja az SAP, már a cseh hokisok is ezt használják. <https://computerworld.hu/tech/a-foci-vebe-technologiajat-kinalja-az-sap-mar-a-cseh-hokisok-is-ezt-hasznaljak-162334.html>

<https://computerworld.hu/tech/a-foci-vebe-technologiajat-kinalja-az-sap-mar-a-cseh-hokisok-is-ezt-hasznaljak-162334.html>

Letöltve: 2021.11.20

- www.iocharts.io, 2021 Mi az EPS jelentése, mit mutat meg az EPS és a P/E mutató?

<https://iocharts.io/blog/eps-mutato-jelentese/>

Letöltve: 2021.11.20

- www.oktato.econ.unideb.hu, 2021:

<http://oktato.econ.unideb.hu/domician/Downloads/ppt/Ertekpapirook.pdf>

Letöltve:2021.11.20

- [www.szamvitelezz.hupont.hu](http://www.szamvitelezz.hupont.hu/51/penzugyi-mutato-szamok), 2021: Pénzügyi mutatószámok:
<http://szamvitelezz.hupont.hu/51/penzugyi-mutato-szamok>

Letöltve: 2021.11.20

- [www.variance.hu](https://variance.hu/2017/04/23/machine-learning-ml-ii-fejezet-linearis-regresszio/), 2017: Machine Learning (ML) II. fejezet – lineáris regresszió
[/https://variance.hu/2017/04/23/machine-learning-ml-ii-fejezet-linearis-regresszio/](https://variance.hu/2017/04/23/machine-learning-ml-ii-fejezet-linearis-regresszio/)

Letöltve: 2021.11.20

- [www.webbeteg.hu](https://www.webbeteg.hu/cikkek/pr/25409/mesterseges-intelligencia-a-rakgyogyitasban), 2021: Mesterséges intelligencia a rákgyógyításban:
<https://www.webbeteg.hu/cikkek/pr/25409/mesterseges-intelligencia-a-rakgyogyitasban>

Letöltve: 2021.11.20

ÁBRÁK JEGYZÉKE

1. „Infláció alakulása”, saját szerkesztés forrás KSH.
2. „Hírek hatása az árfolyamokra” - Elon Musk VS Bitcont:
<https://www.vox.com/recode/2021/5/18/22441831/elon-musk-bitcoin-dogecoin-crypto-prices-tesla>
3. „Neurális hálók példa” Forrás: https://www.inf.u-szeged.hu/~rfarkas/ML20/deep_learning.html
4. „Stock Analysis interfész”, saját szerkesztés forrás Stock Analysis
5. „WebSraping Eredménye”, saját szerkesztés, forrás Stock Analysis.
6. „Dashboard”, saját szerkesztés, forrás Stock Analysis.
7. „Likviditás, P/B, P/E értékek ”, saját szerkesztés, forrás Stock Analysis.
8. Tesla bevétel, nyereség aránya, saját szerkesztés, forrás Stock Analysis.
9. „Legjobb részvények”, saját szerkesztés, forrás Stock Analysis.
10. „Amazon záróárfolyam”, saját szerkesztés, forrás Yahoo finance
11. „Amazon záróárfolyama normál görbére vetítve”, saját szerkesztés, forrás Yahoo finance
12. „Amazon 200 napos mozgó átlag és szórás”, saját szerkesztés, forrás Yahoo finance
13. „Amazon Szezonalitás kizárása”, saját szerkesztés, forrás Yahoo finance
14. „Értékek logaritmus”, saját szerkesztés, forrás Yahoo finance
15. „Auto ARIMA eredmények”, saját szerkesztés, forrás Yahoo finance
16. „Amazon árfolyam előrejelzés”, saját szerkesztés, forrás Yahoo finance
17. „Amazon árfolyam előrejelzés elmúlt egy év alapján”, saját szerkesztés, forrás Yahoo finance
18. „Amazon árfolyam előrejelzés elmúlt 100 nap alapján”, saját szerkesztés, forrás Yahoo finance
19. „S&P500 Index követő ETF előrejelzés, RNN”, saját szerkesztés, forrás Yahoo finance
20. „S&P500 Index követő ETF előrejelzés, ARIMA”, saját szerkesztés, forrás Yahoo finance
21. „S&P500 Index Techniki elemzés”, saját szerkesztés, forrás investing.com
22. „Tesla előrejelzés, ARIMA”, saját szerkesztés, forrás Yahoo finance
23. „Apple előrejelzés, LSTM”, saját szerkesztés, forrás Yahoo finance