

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Klemen Rizman

**Primerjava uspešnosti metod
poslovne inteligence ter metod
tehnične analize za napovedovanje
gibanja cen vrednostnih papirjev**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Tomaž Hovelja

Ljubljana, 2017

Fakulteta za računalništvo in informatiko podpira javno dostopnost znanstvenih, strokovnih in razvojnih rezultatov. Zato priporoča objavo dela pod katero od licenc, ki omogočajo prosto razširjanje diplomskega dela in/ali možnost nadaljne proste uporabe dela. Ena izmed možnosti je izdaja diplomskega dela pod katero od Creative Commons licenc <http://creativecommons.si>

Morebitno pripadajočo programsko kodo praviloma objavite pod, denimo, licenco *GNU General Public License, različica 3*. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Kandidat naj predstavi metode poslovne intelligence, predvsem naj se usmeri na tiste, ki so primerne za napovedovanje cen vrednostnih papirjev. Kandidat naj predstavi metode tehnične analize, ki jih finančna stroka tradicionalno uporablja za napovedovanje cen vrednostnih papirjev. Kandidat naj razvije program za analizo cen vrednostnih papirjev in primerja natančnost metod poslovne intelligence z dejanskim stanjem ter metodami tehnične analize.

Najprej bi se zahvalil doc. dr. Tomažu Hovelji za mentorstvo in pomoč pri izdelavi diplomske naloge. Največja zahvala gre mojim staršem, bratu Tilnu in vsem prijateljem, ki so mi bili v oporo v času študija in ob izdelavi diplomske naloge.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Poslovna inteligenca	5
2.1	Poslovna inteligenca	6
2.2	Strojno učenje	9
2.3	Polinomska regresija	10
2.4	RMSE (root mean square error)	12
3	Najbolj uporabljene metode tehnične analize vrednostnih pa- pirjev	15
3.1	Drseče sredine	16
3.2	Oscilatorji	19
4	Program za napovedovanje cen vrednostnih papirjev	23
4.1	Pregled uporabljenih tehnologij	23
4.2	Program	24
5	Podatki	31
5.1	Opis	31
5.2	Pridobivanje podatkov	31
5.3	Urejanje podatkov	32

5.4	Analiza rezultatov	33
6	Primerjava metod tehnične analize in polinomske regresije	39
6.1	Polinomska regresija	39
6.2	Tehnična analiza	39
6.3	Primerjava	40
7	Zaključek	45
8	Priloga	49

Seznam uporabljenih kratic

kratica	angleško	slovensko
BI	business intelligence	poslovna inteligenca
RMSE	root mean square error	kvadratna napaka
SMA	simple moving average	enostavna drseča sredina
EMA	exponential moving average	eksponentna drseča sredina
RSI	relative strength index	indeks relativne moči
MACD	moving average convergence divergence	

Povzetek

Naslov: Primerjava uspešnosti metod poslovne inteligence ter metod tehnične analize za napovedovanje gibanja cen vrednostnih papirjev

Diplomska naloga obravnava področje poslovne inteligence (BI). Poslovna inteligenca se ukvarja z orodji in algoritmi, ki se izvajajo nad neobdelanimi podatki, ki jih skušajo spremeniti v uporabne informacije za poslovne analize in napovedi. Tehnologija, ki jo uporabljajo v poslovni inteligenci, je največkrat namenjena obdelavi ogromne količine neurejenih podatkov (big data) ter smiselni in čim bolj enostavni obrazložitvi pridobljenih rezultatov.

V svoji nalogi sem uporabil polinomske regresije, ki se uporabljajo v poslovni inteligenci, in jo prilagodil za napovedovanje gibanja cen vrednostnih papirjev. Dobljene rezultate sem primerjal z metodami, ki jih uporabljajo v finančni industriji za napovedovanje cen vrednostnih papirjev.

Ključne besede: poslovna inteligenca, polinomska regresija, tehnična analiza.

Abstract

Title: Successfulness Comparison between Business Intelligence Methods and Technical Analysis Methods for Stock Price Forecasting

The graduation thesis deals with the area of business intelligence (BI). Business intelligence deals with tools and algorithms, which are used on raw data in order to change them into useful information for business analyses and forecasts. Technology used in business intelligence is mostly used for big data processing. The results we get from these methods need to be processed logically and as simply as possible.

In my thesis, I used polynomial regression, which is used in business intelligence. I modified it for forecasting of stock's price movement. I compared the obtained result to different methods used in financial industry for forecasting stock's price movements.

Keywords: business intelligence, polynomial regression, technical analysis.

Poglavje 1

Uvod

Poslovna inteligenca in analiza velikih podatkov sta postali v zadnjih dveh desetletjih zelo pomembni področji v akademskih in poslovnih krogih. Analize kažejo, da kar 97% podjetij, ki imajo letno vsaj 100 milijonov dolarjev prihodkov, uporablja vsaj nekaj orodij poslovne inteligenca. Poročilo, ki ga je leta 2011 naredil ameriški institut McKinsey Global Institute [8], predvideva, da bo do leta 2018 samo v Združenih državah Amerike potreba po kar 140.000-190.000 novih strokovnjakov na področju poslovne analize in kar 1,5 milijona podatkovnih skrbnikov. Priložnosti, ki so povezane s podatki in njihovo analizo na drugih področjih, so pripomogle k povečanju zanimanja za poslovno inteligenca in analizo podatkov. Z uporabljenimi metodami, analizami, programi in pridobljenimi izkušnjami poslovna inteligenca pomaga podjetjem, da bolje razumejo svoje stranke in trg, na katerem poslujejo, ter na podlagi rezultatov izberejo boljše dolgoročne odločitve. [1].

Poslovna inteligenca vsebuje orodja za analizo podatkov, s katerimi predstavi kompleksne in pomembne informacije ljudem, ki se ukvarjajo z odločanjem ter načrtovanjem pomembnih odločitev. Poslovna inteligenca se uporablja za razumevanje zmogljivosti podjetja, poslovnih trendov, strateških odločitev na trgu, tehnologije, okolja, v katerem podjetje deluje, in analize odločitev konkurenčnih podjetij.[9]

Tehnična analiza je orodje, ki ga uporabljajo strokovnjaki za napovedo-

vanje cen vrednostnih papirjev na podlagi preteklih vrednosti. Za razliko od temeljne analize, ki napoveduje poštene vrednosti delnic, se tehnična analiza osredotoča na dejanske vrednosti delnic in na podlagi teh napoveduje prihodnje dvige ter padce vrednosti delnic. Metode tehnične analize delujejo za napovedovanje vrednosti vseh trgovalnih instrumentov, kjer vrednost cene instrumenta določa trg s povpraševanjem in dobavo želenega instrumenta.

V zadnjih letih so strokovnjaki na področju tehnične analize razvili mnogo novih indikatorjev za napovedovanje cen vrednostnih papirjev v upanju, da izboljšajo natančnost napovedovanja cen. Nekateri indikatorji so primarno osredotočeni na identifikacijo trenutnih trendov na trgu, medtem ko so drugi osredotočeni predvsem na moč smeri, v kateri se gibljejo trendi.

Tehnični analitiki obravnavajo različne časovne okvirje. Del tehničnih analitikov se ukvarja s kratkoročnimi naložbami, zato analizirajo predvsem kratke časovne intervale, kot so minutni, urni in štiriurni intervali. Analitiki, ki se ukvarjajo z dolgoročnimi naložbami, pa spremljajo predvsem daljše časovne okvirje, kot so teden, meseci in leto.[16]

V diplomskem delu se bom seznanil s poslovno inteligenco in tehnično analizo, saj sta to področji, ki se ukvarjata z napovedjo cen vrednostnih papirjev. Motivaciji te diplomske naloge sta ugotoviti uspešnost metod poslovne inteligence in zmožnost napovedovanja cen vrednostnih papirjev, s katerimi lahko bralec sam analizira trg ter na podlagi analize uspešno investira v vrednostne papirje.

V diplomskem delu želim ugotoviti, katere metode poslovne inteligence so primerne za analizo vrednosti gibanja cen vrednostnih papirjev, katere tehnične metode so najbolj uveljavljene, kako uspešne so posamezne metode v primerjavi z dejansko uspešnostjo sklada SPDR 500 (SPY) in katere metode so za napovedovanje gibanja cene delnic najbolj primerne? Ključni cilj, ki sem si ga zadal za diplomsko delo, je narediti nov sklad, ki bo imel višjo donosnost kot sklad SPDR 500 (SPY).

Da bi te cilje dosegel, sem diplomsko delo strukturiral na naslednji način. V prvem poglavju sem predstavil poslovno inteligenco in metode, ki jih upo-

rabljamo. Ker poslovna inteligenca uporablja metode strojnega učenja, sem bolj podrobno predstavil še področje strojnega učenja in uporabljene metode. V drugem poglavju sem opisal področje tehnične analize in najbolj uporabljene metode, ki se uporabljajo pri tehnični analizi. Naslednje poglavje je namenjeno predstavitvi mojega programa za analizo vrednosti cen vrednostnih papirjev. V predzadnjem poglavju sem opisal uporabljene podatke in njihovo obdelavo za izdelavo napovednega modela. V zadnjem poglavju pa sem med seboj primerjal dobljene rezultate in jih primerno pokomentiral.

Poglavje 2

Poslovna inteligenca

Sodobne računalniške in komunikacijske tehnologije omogočajo zbiranje, shranjevanje ter hiter dostop do velikih količin podatkov. Obstoječe baze podatkov pa ostajajo v veliki meri neobdelane in neizkoriščene. V njih je še veliko neodkritih in potencialno uporabnih informacij, do katerih bi lahko prišli le z ustrezno analizo podatkov. Razkorak med velikim obsegom uskladiščenih podatkov in njihovo skromno izkoriščenostjo je spodbudil razvoj novega področja v računalništvu, t. i. podatkovnega rudarjenja (data mining), imenovanega tudi odkrivanje znanja v podatkovnih bazah. Področje se ukvarja z odkrivanjem implicitnih, doslej neznanih in potencialno uporabnih informacij v podatkih. Ključni korak v tem pristopu je uporaba strojnega učenja (machine learning) na podatkih. Metode strojnega učenja iz obravnavanih podatkov izpeljejo posplošene opise v različnih oblikah, kot so npr. odločitvena drevesa ali klasifikacijska pravila. Ti rezultati omogočajo modeliranje, klasifikacijo in napovedovanje, za razliko od rezultatov statističnih metod pa dajejo tudi boljši vpogled v obravnavano problematiko ter s tem pripomorejo k njenemu boljšemu razumevanju. [5]

2.1 Poslovna inteligenca

Po nekaterih ocenah se količina podatkov, shranjenih v podatkovnih bazah po vsem svetu, podvoji vsakih dvajset mesecev. Podatke s svojimi aktivnostmi ustvarjamo vsi subjekti, od posameznikov do največjih korporacij, saj pri današnjem načinu življenja in poslovanja skoraj ni več aktivnosti, ki ne bi bila tudi računalniško podprta. Shranjevanje podatkov je enostavno in poceni, kar velja tako na nivoju posameznega osebnega računalnika kot globalnega medmrežja. Ob nenehnem napredovanju pomnilniških zmogljivosti in globalizaciji procesov, ki podatke ustvarjajo, eksponentni rasti obsega shranjenih podatkov ni videti konca. Ob tem pa se povečuje razkorak med količino ustvarjenih podatkov in količino podatkov, ki jih ljudje obvladujemo, razumemo ter koristno uporabljamo. Naše tovrstne zmogljivosti napredujejo zelo počasi, zato se z večanjem obsega uskladiščenih podatkov delež razumljenih in koristno izrabljenih drastično manjša. [5]

Korak k zmanjševanju te razlike je bil storjen z začetkom razvoja metodologije odkrivanja znanja v podatkovnih bazah okrog leta 1990. Metodologija se loteva reševanja problemov z analizo podatkov, ki so že shranjeni v podatkovnih bazah. Njeno bistvo je v iskanju takšnih vzorcev (povezav, odvisnosti) v podatkih, ki na ekspliciten način povzemajo v podatkih vsebovane zakonitosti, omogočajo sklepanje in pojasnjevanje ter so uporabni pri nadaljnjih odločitvah. [5]

Odkrivanje znanja v podatkovnih bazah pomeni nadaljevanje in razširitev koncepta strojnega učenja. Strojno učenje je tehnična (algoritmična) osnova, ki jo je novo področje izpopolnilo in prilagodilo za obravnavanje velikih baz podatkov, s kakršnimi se srečujemo v praksi. Zanje so poleg kompleksnosti ponavadi značilni še redundantnost zapisov, šum v podatkih, manjkajoče vrednosti in podobne, za formalno obravnavo neugodne lastnosti, ki pa se jim v praksi le stežka izognemo. [5]

Odkrivanje znanja v podatkih je proces, ki vsebuje postopke izbire podatkov, predprocesiranja podatkov, transformacijo podatkov, analizo podatkov in interpretacijo ter ovrednotenje podatkov. [5]

Izbira podatkov pomeni določitev tistih podatkov iz množice razpoložljivih, ki jih bomo analizirali. Izbiro lahko opravimo tako na nivoju podatkovnih zapisov kot na nivoju komponent zapisov. Dobljeni podmnožici podatkov pravimo ciljni podatki. Za izbiro podatkov obstajajo različne metode in pristopi. Izbira najprimernejše metode je odvisna od primera do primera, ugotovimo pa lahko, da ne obstaja metoda, ki bi bila optimalna za vsak izbrani primer. Med najbolj razširjene pristope za iskanje optimalnega nabora podatkov spadajo ANOVA (Analysis of variance), LDA (Linear discriminant analysis) in Pearsonova korelacija. [4]

Predprocesiranje podatkov običajno pomeni nadaljnje krčenje ciljnih podatkov. V tem koraku ugotavljamo prisotnost šuma v podatkih, kot so napačne ali manjkajoče vrednosti. Napake so lahko sintaktične ali semantične. Pojavijo se pri vnašanju, prenosu ali predhodnih obdelavah podatkov. Sintaktične napake zaznamo z leksikalnim pregledovanjem podatkov, semantične (npr. vrednost podatka izven pričakovanega obsega vrednosti) pa s statističnim preverjanjem. Takšne zapise popravimo, če je to mogoče, v nasprotnem primeru pa jih izločimo iz nadaljnje obravnave. Manjkajoče vrednosti podatkov še ne pomenijo nujno, da je neki podatkovni zapis neuporaben za analizo. Mnoge metode strojnega učenja namreč znajo obravnavati tudi manjkajoče vrednosti. Pri predprocesiranju si veliko pomagamo z grafičnimi prikazi podatkov. Odvisno od vrste spremenljivke uporabljamo različne tipe grafov, kot so histogrami, grafikon kvartilov, lijakasti grafikon, stopničasti grafikon in ostali. [17]

Transformacija podatkov je namenjena neposredni pripravi predprocesiranih podatkov za analizo z izbrano metodo strojnega učenja. Podatke iz obstoječega formata pretvorimo v format, kakršnega zahteva izbrana metoda. S tem iz predprocesiranih podatkov dobimo transformirane podatke. Cilj transformacije je pretvoriti vse podatke v numerično obliko. [17]

Ključni korak v postopku odkrivanja znanja v podatkovnih bazah je njihova analiza oz. obdelava s strojnimi učenjem. Zanj uporabimo primerno metodo, ki ustreza tipu zastavljene naloge (klasifikacija, razvrščanje v sku-

pine, regresijsko modeliranje) in lastnostim podatkov (zvezni ali diskretni, urejene ali neurejene vrednosti, prisotnost neznanih vrednosti ipd.). Analiza odkrije relevantne vzorce v podatkih. Ti so lahko predstavljeni na različne načine, npr. kot odločitvena drevesa, pravila tipa če – potem (angl. if–then), odločitveni sezname, regresijska drevesa itd. Oblika je pogojena z vrsto problema in uporabljenim algoritmom strojnega učenja, pa tudi z namenom uporabe.

Obdelavi podatkov sledita interpretacija in ovrednotenje odkritih zakonitosti. Dobljene rezultate pojasnimo v kontekstu obravnavane problematike in jih primerjamo z obstoječim znanjem ter morebitnimi rezultati drugih metod. S testi ovrednotimo tudi zanesljivost rezultatov. Podamo jo kot pričakovano klasifikacijsko točnost, napako napovedovanja ipd., odvisno od uporabljene metode in oblike rezultata. [5]

Poslovna inteligenca uporablja metode, ki jih razvijajo na področju strojnega učenja in podatkovnega rudarjenja. Poznamo dve vrsti strojnega učenja, ki se razlikujeta po rezultatih, ki jih napovedujemo. [2] Pri nadzorovanem učenju izhodne vrednosti določa učitelj, kar pomeni, da so izhodne vrednosti, oziroma zaloga vrednosti izhoda, vnaprej določene. Nadzorovano učenje vsebuje učno množico, ki vsebuje bazo podatkov in želeni rezultat, ki ga želimo napovedati. Algoritem nato analizira podatke in ustvari funkcijo, na podlagi katere lahko vhodnim podatkom, ki nimajo končnih napovedi, določimo najbolj primeren rezultat.

V nadzorovano učenje spadata klasifikacija in regresija. Cilj klasifikacije je razvrstiti vhodne primere na podlagi vrednosti podatkov v posamezne razrede. Pri klasifikaciji algoritmu kot učno množico podamo primere, ki so že razvrščeni v razrede. Algoritem na podlagi vrednosti podatkov ustvari model, ki vsebuje pravila oziroma zakonitosti za pripadnost primera k določenemu razredu. Nato kot vhodni parameter modela pripravimo primere, ki jih želimo razvrstiti, model pa nam vrne seznam primerov in razred, v katerega jih je določil.

Regresija se ukvarja z oceno razmerja med spremenljivko, ki jo želimo

napovedati, in ostalimi odvisnimi ter neodvisnimi atributi. Kot učno množico algoritmu podamo primere in njihovo napovedno vrednost, ki je predstavljena kot zvezna spremenljivka. Algoritem ustvari polinomske funkcije, v katero nato vstavimo vhodne parametre, funkcija pa nam izračuna končno vrednost.

Druga skupina strojnega učenja pa je nenadzorovano učenje. Nenadzorovano učenje poizkuša v množici podatkov najti skupine primerov, ki imajo podobne lastnosti, ter jim na podlagi teh lastnosti določiti pravila oziroma zakonitosti. Segmentacija spada v skupino nenadzorovanega učenja, kjer algoritem skuša najti skupine podobnih primerov s skupnimi lastnostmi. Segmentacija je koristna v poslovni uporabi za ustvarjanje novih produktov, kjer algoritem najde nove skupine uporabnikov, strokovnjaki pa na podlagi lastnosti posameznih skupin kreirajo nove produkte, ki so personalizirani za nove skupine uporabnikov.

2.2 Strojno učenje

Pod pojmom učenje razumemo pridobivanje znanja, bodisi preko neposrednega sprejemanja od učitelja bodisi s samostojnim študijem ali skozi izkušnje. To pojmovanje se seveda nanaša na učenje pri človeku, ko pa govorimo o strojnem učenju, ki naj bi ga izvajal stroj (računalnik), ta definicija ne zadošča.[5] Zato v splošnem kot učenje opredelimo tiste aktivnosti, ki pripomorejo k temu, da neki sistem bolje opravlja svoje naloge. [5] Obstaja več vrst učenja, ki se razlikujejo po tem, kakšni sta vlogi sistema, ki se uči, in sistema, ki poučuje. V primeru enostavnega učenja sistem, ki se uči, zgolj sprejema znanje od učitelja, ne da bi ga sam preverjal ali dopolnjeval. Pri človeku je to pomnjenje podanih dejstev, pri računalniku pa sprejemanje navodil za delovanje (programiranje). Drugi skrajni primer je samostojno odkrivanje zakonitosti brez sodelovanja učitelja. To je najvišja oblika učenja pri človeku, računalniki pa je v splošnem še ne obvladajo. Vmes med obema pristopoma je učenje na osnovi primerov, imenovano tudi induktivno učenje. V tem primeru sistemu, ki se uči, posredujemo učne primere, ki predstavljajo rešene

primere z nekega področja, sistem pa primere posploši tako, da iz njih izpelje splošna pravila ali zakonitosti, ki omogočajo nadaljnje reševanje problemov na tem področju.

Strojno učenje je nastalo kot alternativa statistični analizi podatkov. Njegov namen ni zgolj graditi čim bolj zanesljive modele iz podatkov, temveč morajo biti modeli tudi takšni, da jih je moč pojasniti in razumeti. Prav transparentnost rezultatov je bistvena novost, ki jo prinašajo sistemi strojnega učenja v primerjavi s statističnimi metodami. [5]

2.3 Polinomska regresija

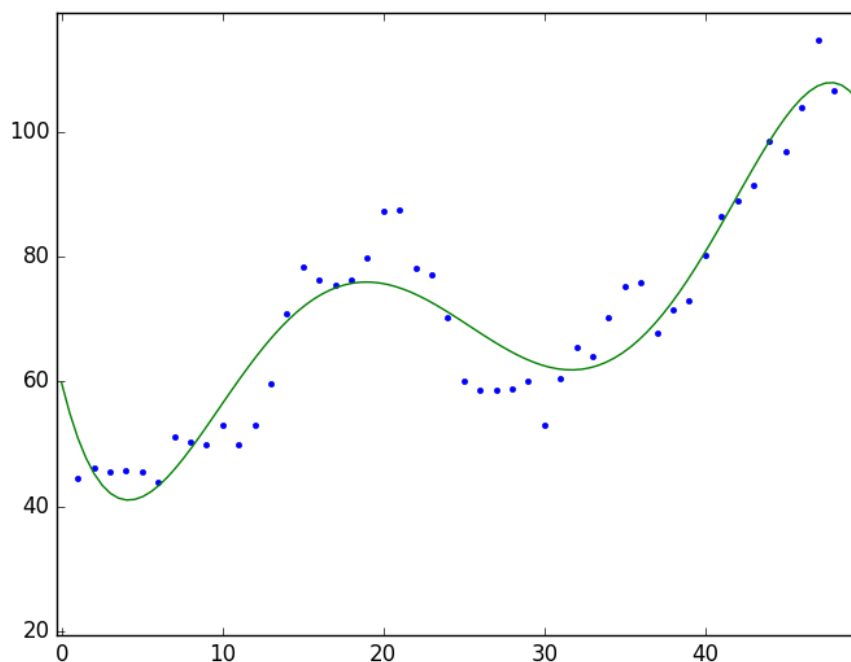
V izbranem članku [11] navajajo, da so med seboj primerjali linearne in polinomske regresijske modele ter ugotovili, »da je linearna regresija manj občutljiva pri nenadnih spremembah«. To pomeni, da se linearna regresija manj prilagaja vrednosti delnic in kaže trend delnice. V diplomskem delu se mi zdi pomembno, da algoritem zaznava nenadne spremembe v vrednosti delnic, zato sem se odločil za uporabo polinomske regresije. Za polinomske regresije sem se odločil, ker jo lahko uporabimo tudi kot linearno regresijo in nam daje več možnosti za raziskovanje. Polinomska regresija je vrsta regresije, v kateri je razmerje med neodvisno spremenljivko x in odvisno spremenljivko y predstavljeno kot n -ta stopnja polinoma. Cene vrednostnih papirjev konstantno padajo in se dvigajo, zato jih bomo lažje simulirali s polinomske kot linearno regresijo, saj premica pri linearni regresiji lahko samo konstantno pada oziroma se konstantno dviga. Linearna regresija ne zaznava vmesnih padcev oziroma dvigov cen, zato pri nenadnem visokem dvigu oziroma padcu vrednosti delnice z linearno regresijo dosegamo visoke napake. Kljub temu pa lahko polinomske regresije uporabimo kot linearno regresijo, če uporabimo prvo stopnjo polinoma, kar nam daje dodatne možnosti pri raziskovanju, in imamo pokrito celotno področje regresije.

$$y = a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n + \varepsilon_i (i = 1, 2, \dots, n)$$

V našem primeru y predstavlja našo krivuljo (vrednosti delnic), x pred-

stavlja mesece, ki so predstavljeni s števkami od 1 do 60 ($60 = 5 * 12$; 5 let, 12 mesecev), in n , ki predstavlja stopnjo polinoma.

Vendar pa je pri polinomske regresiji zelo pomembno, da določimo pravilno stopnjo polinoma, saj lahko pride do pretiranega prilagajanja (overfitting) ali pa preveč nenatančne napovedi. Pretirano prilagajanje je pojav, ki se pojavi, ko naš model dosega visoko natančnost na učnih podatkih in nizko natančnost na testnih podatkih. To se zgodi v primerih, ko se naš napovedni model preveč prilagaja učnim podatkom. Ker ima vsaka delnica različno krivuljo, sem na učnih podatkih za posamezne delnice določil stopnjo polinoma, ki sem jih ocenil z RMSE (root mean square error) z oceno (root mean square error). (slika: 2.1.)



Slika 2.1: Primer dobre polinomske regresije

2.4 RMSE (root mean square error)

Pri regresiji za merjenje uspešnosti uporabljamo različne ocene napak. Najbolj uporabljeni napaki sta RMSE (root mean square error) in MSE (Mean squared error). Obe napaki merita iste parametre, razlika je le v tem, da je MSE (Mean squared error) kvadratna vrednost RMSE, kar pomeni, da je RMSE lažja za razumevanje, saj je končni rezultat v istih enotah kot vhodni podatki, zato je uporabniku ta vrednost bolj razumljiva. Koren povprečne kvadratne napake oziroma RMSE uporabljamo za merjenje povprečja od napake. To pomeni, da nam RMSE pove razliko med napovedanimi in realnimi podatki. Za čim boljše napovedi želimo čim manjši RMSE, kar pomeni, da bi bil v primeru točne napovedi brez napak naš RMSE enak nič.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

V našem primeru n predstavlja število mesecev, ki jih napovedujemo, y_i predstavlja realno vrednost, ki se jo hočemo napovedati za posamezni mesec, \hat{y}_i pa predstavlja našo napovedano vrednost za posamezni mesec. Prva štiri leta pridobljenih podatkov sem razdelil na učno množico (prva tri leta) in testno množico (zadnje leto v učni množici). To pomeni, da y_i predstavlja testno množico podatkov in \hat{y}_i predstavlja napovedane podatke za četrto leto. Nato sem za prvih devet stopenj polinoma izdelal polinomske regresije in izračunal njeno napako s pomočjo RMSE. Tako sem dobil za vsako delnico različno polinomske regresije z različno stopnjo. Izmed vseh izračunanih vrednosti RMSE sem izbral polinomske stopnje, ki je imela najnižjo vrednost RMSE in jo uporabil za stopnjo pri izbranem vrednostnem papirju.

Iz tabele 2.1 lahko vidimo, da višja kot je stopnja polinoma, manjša je natančnost. Iz tega lahko sklepamo, da se pri višjih stopnjah model bolj prilagaja podatkom, kar pomeni, da prihaja do pretiranega prilagajanja. Ker ima tretja stopnja najnižjo vrednost RMSE, bomo to stopnjo uporabili za naše napovedi. Prva in druga stopnja imata višjo vrednost RMSE od tretje stopnje. Iz tega lahko sklepamo, da prva in druga stopnja ne zaznavata nenadnega dviga oziroma padca vrednosti delnice.

Tabela 2.1: Primer RMSE za različne stopnje polinomske regresije delnic podjetja CenterPoint Energy

stopnja polinoma	vrednost RMSE
1	26.564
2	26.428
3	2.022
4	41.150
5	311.794
6	18.765
7	35662.659
8	153807.813
9	2699720.289

Poglavje 3

Najbolj uporabljene metode tehnične analize vrednostnih papierjev

Tehnična analiza je analiza obnašanja trga na podlagi uporabe grafov za napovedovanje prihodnosti trendov vrednosti delnic. Izraz »obnašanje trga« vsebuje tri vire informacij, dostopnih za tehnično analizo (vrednost, obseg in obresti). Drugi izraz »obnašanje vrednosti cene«, ki se ga velikokrat omejnja, se zdi preveč ozek, ker večino tehničnih analitikov uporablja obseg in obresti kot del njihovih analiz trga.

Koncept trenda je bistvenega pomena za tehnični pristop. Dokler oseba, ki se želi ukvarjati s tehnično analizo, ne sprejme premise, da se trg giblje v trendih, mu tehnična analiza ne bo v pomoč. Celoten namen grafičnega prikaza nihanja vrednosti cene vrednostnega papirja je identifikacija trenda v zgodnjih fazah razvoja za namen trgovanja v smereh trenda. Večina tehnik, uporabljenih pri tehnični analizi, je narejena na podlagi sledenja trendom za namen, da se jih identificira in sledi. [12] Na podlagi Newtonovega prvega zakona je izpeljan koncept, ki pravi, da je bolj verjetno, da se bo trend v gibanju nadaljeval v smeri gibanja, kot pa obrnil, kar pomeni, da se bo trend gibanja vrednosti cene delnic nadaljeval, dokler se trend ne bo obrnil.

Veliko tehničnih analiz in analiz trga se nanaša na analizo človeške psihologije. Vzorci, ki jih najdemo na grafu, so bili prepoznani in kategorizirani v preteklih 100 letih. Ti vzorci so še vedno aktualni in jih najdemo na današnjih grafih ter razkrivajo dvig ali padec v obnašanju trga. Ker so ti vzorci delovali dobro v preteklosti, lahko predpostavimo, da bodo dobro delovali tudi v prihodnosti, saj delujejo na podlagi človeške psihologije. Vse to lahko opišemo kot ponavljanje zgodovine, ki je ključna za razumevanje prihodnosti.

V naslednjih poglavjih bom predstavil najbolj pogoste metode tehnične analize, ki so bile predstavljene v knjigi *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*, ki je ena izmed napogosteje citiranih knjig na področju tehnične analize.

Za primerjavo in analizo sem vzel vrednost ene enote sklada. S polinomsko regresijo tretje stopnje sem izračunal, da bo vrednost enote sklada 1. 8. 2016 znašala \$236.17, kar pomeni, da je napovedana vrednost večja od realne za \$18.93.

Tabela 3.1: Vrednost ene enote sklada na dan 1. 8. 2016

Opis	Vrednost
Open	\$217.19
High	\$219.6
Low	\$214.25
Close	\$217.38
Napovedana vrednost	\$236.31

3.1 Drseče sredine

Drseča sredina je orodje za sledenje trendu. Njen namen je prepoznati in opozoriti, da se je začel nov trend oziroma se je trend končal. Drseča sredina ne napoveduje obnašanja trga, kot to počnejo standardne analize grafov,

temveč sledi aktualnim trendom trga, zato lahko rečemo, da je drseča sredina sledilec in ne vodja obnašanja trga.

Drseča sredina je orodje za glajenje, saj s tem, ko naredi povprečje vrednosti cene delnice, izrisuje gladko črto, ki je enostavnejša za interpretacijo trendov. Zaradi svoje narave drseča sredina deluje z zamikom, ki ga lahko zmanjšamo, a bo vedno prisoten. Zamik prilagajamo z manjšanjem ali večanjem časovnega okvirja. Izbira časovnega okvirja je odvisna od trga, ki ga analiziramo, saj so krajši časovni okvirji bolj občutljivi na spremembe v vrednosti cene delnice, daljši pa manj.

Pri uporabi drseče sredine uporabljamo končno ceno vrednosti, ki je najbolj uporabna vrednost trgovalnega dne, vendar nekateri analitiki uporabljajo tudi druge vrednosti v dnevu. Za drsečo sredino lahko uporabimo tudi povprečje vseh vrednosti v trgovalnem dnevu ali pa samo začetno in končno vrednost, ki pa delujeta kot nevtralno območje v vrednosti cene delnic. [12]

3.1.1 Enostavna drseča sredina

Enostavna drseča sredina ali povprečna vrednost je najpogosteje uporabljena metoda tehnične analize. Nekateri analitiki dvomijo v njeno uporabnost zaradi dveh omejitev. Prva omejitev enostavne drseče sredine je omejenost na časovni okvir, saj ga moramo izbrati in analizirati znotraj izbranega okvirja. Druga omejitev enostavne drseče sredine je, da so vsi dnevi znotraj časovnega okvirja enako pomembni. To pomeni, da prvi dan v 10-dnevnom časovnem okvirju doda 10% h končni vrednosti, tako kot zadnji dan v istem časovnem okvirju, zato nekateri analitiki menijo, da bi bilo bolje, če bi posamezne dneve različno otežili.

Enostavna drseča sredina je neuravnoteženo povprečje prejšnjih n vrednosti. Če želimo izračunati 20-dnevno enostavno drsečo sredino in vrednosti delnic označimo z $\rho_M, \rho_{M-1}, \dots, \rho_{M-19}$, potem je naša formula [6]:

$$SMA = \frac{1}{n} \sum_{i=0}^{n-1} \rho_{M-1}$$



Slika 3.1: 20-dnevni SMA za Microfostove delnice

3.1.2 Eksponentna drseča sredina

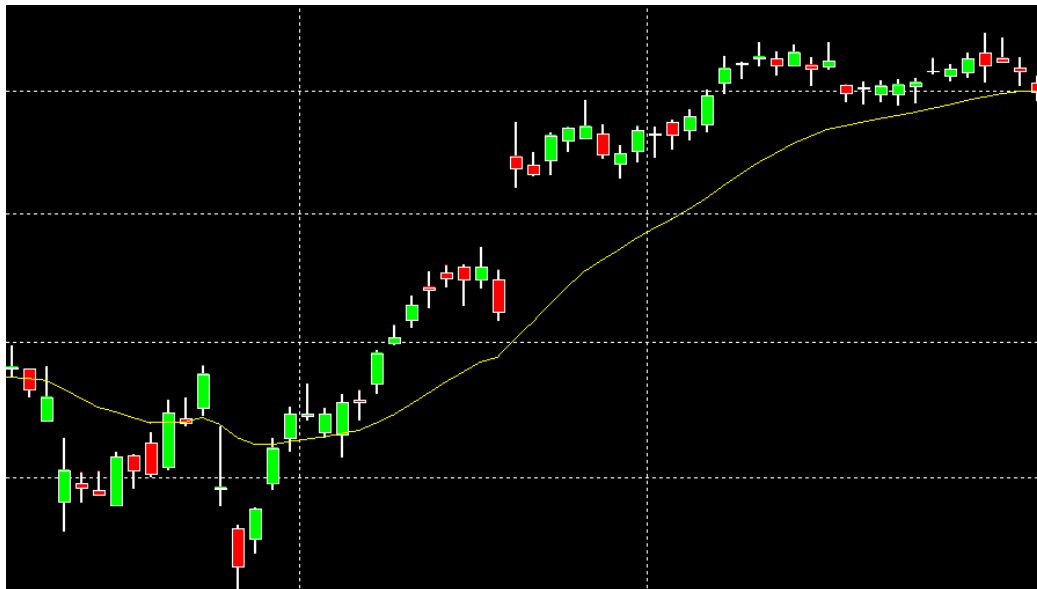
Eksponentna drseča sredina se sooči z obema omejitvama, ki jih ima enostavna drseča sredina. Utežitev časovnih intervalov reši tako, da novejšim podatkom dvigne vpliv, medtem ko ga starejšim zmanjša. Ta način je realiziran tako, da najnovejši vrednosti dodeli delež, ki se doda k deležu od predhodne vrednosti. Ker vrednostim dodamo uteži, dosežemo, da so starejši podatki manj pomembni od novejših, zato eksponentna drseča sredina vzame vse vrednosti, ki obstajajo za opazovani predmet, s čimer odpravimo omejitev enostavne drseče sredine. Končna prednost eksponentne drseče sredine je hitrejša reakcija na zadnje spremembe v vrednosti delnic. [12]

$$EMA_n = EMA_{n-1} + \alpha * (\rho_n - EMA_{n-1})$$

$$\alpha = \frac{2}{1+n}$$

3.1.3 Kazalnik MACD

MACD je razlika med dvema eksponentnima drsečima sredinama. V praksi se največkrat uporablja razlika med 12-dnevno in 26-dnevno eksponentno drsečo



Slika 3.2: 20-dnevni EMA za Microfostove delnice

sredino, ki ustvari novo MACD-črto. Na grafu MACD prikaže še počasnejšo, signalno črto, ki predstavlja eksponentno drsečo sredino z obdobjem 9 period, narejeno na podlagi MACD-črte. Avtor, Gerald Appel, priporoča, da se ne uporablja privzetih vrednosti (9, 12, 26) za nakup in prodajo vrednostnih papirjev, vendar večina strokovnjakov uporablja privzete vrednosti (9, 12, 26).

Na podlagi narejenih signalnih črt MACD uporabnik lahko vidi, kdaj je primeren čas za prodajo oziroma nakup delnic. Signal, ki sporoči nakup delnic, se sproži, ko se MACD-črta dvigne nad počasnejšo signalno črto, medtem ko se signal, ki signalizira prodajo delnice, sproži, ko se MACD-črta spusti pod počasnejšo signalno črto. [12] (Slika: 3.3).

3.2 Oscilatorji

Oscilatorji so zelo uporabni pri analizi trgov, ki se ne gibljejo v trendih, vendar je gibanje cen nepredvidljivo, zato so analize, ki sledijo trendom, neučinkovite. Oscilator analitikom omogoči pomoč pri ustvarjanju zaslužka



Slika 3.3: MACD za 12-dnevno in 26-dnevno EMA

na trgih, ki se ne gibljejo v trendih.

V navezi z grafi pri analizi trgov, ki se gibljejo v trendih, je oscilator zelo pomemben, saj analitika opozori, če prihaja do pojava precenjenosti ali podcenjenosti, na katere mora analitik hitro reagirati. Oscilator analitika opozori tudi, če trend izgublja momentum, še preden ta postane očitno za ostale trgovalce, ali pa če se trend bliža koncu.

Oscilator je sekundarni indikator v smislu, da dopolnjuje analizo trendov trga, vendar je v določenih primerih bolj informativen od analize trendov. Na začetku analize trga so oscilatorji zelo zavajajoči in ne pomagajo pri analizi, vendar so proti koncu analize trga zelo pomembni. [12]

3.2.1 Indeks relativne moči

Eden izmed dveh glavnih problemov pri kreaciji grafa za momentum je neredno gibanje, ki nastane zaradi ostrih sprememb. Oster dvig ali padec pred desetimi dnevi lahko povzroči nenadne spremembe v grafu momentuma, čeprav je sprememba v trenutni vrednosti delnice minimalna. Zaradi tega potrebujemo glajenje grafa, da zmanjšamo te nenadne spremembe. Drugi problem je potreba po konstantni spremembi območja, ki ga potrebujejo za primerjavo. RSI-indikator ne samo poskrbi za potrebno glajenje grafa, temveč tudi reši problem z definiranjem območja med 0 in 100.

Izraz »relativna moč« je velikokrat narobe interpretiran, predvsem med analitiki trga vrednostnih papirjev. Relativna moč pomeni razmerje med dvema različnima atributoma, ki pa ima pri tem indikatorju drugačen pomen. V praksi se največkrat uporablja 14-dnevni časovni interval za dnevni graf in 14-tedenski interval za tedenski graf. Ker se indeks giblje med 0 in 100, sta zelo pomembni meji 30 in 70. Če ima indeks vrednost nad 70, pomeni, da je vrednost delnice precenjena in bo sledil padec cene, če pa je indeks pod 30, pomeni, da bo zaradi podcenjenosti sledil dvig cene. Največkrat se uporablja 14-dnevni RSI, kot je razvidno s spodnje slike. [12]

$$U = close_{danes} - close_{ceraaj}$$

$$D = 0$$

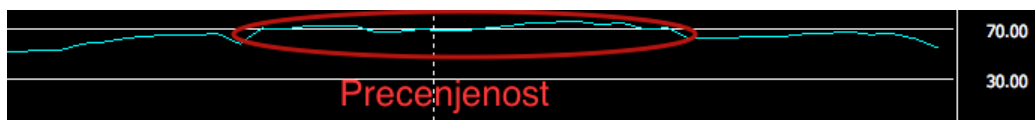
ali

$$U = 0$$

$$D = close_{vceraaj} - closedanes$$

$$RS = \frac{SMMA(U,n)}{SMMA(D,n)}$$

$$RSI = 100 - \frac{100}{1+RS}$$



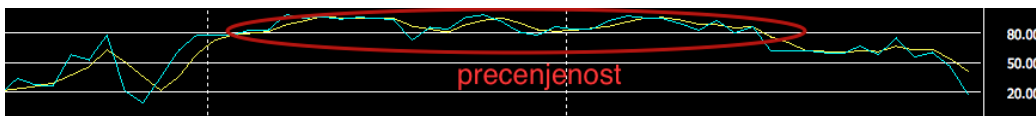
Slika 3.4: 14-dnevni indeks relativne moči

3.2.2 Stohastični oscilator

Stohastični oscilator temelji na opazki, da se zaradi dviga cene delnice zaključna vrednost cene delnice bliža vrhnjemu delu območja vrednosti cene delnice ali pa da se zaradi padca cene delnice zaključna vrednost cene delnice bliža minimalni vrednosti. Namen oscilatorja je ugotoviti, kje se nahaja trenutna vrednost delnice v primerjavi z zaključno vrednostjo.

Stohastični oscilator je sestavljen iz %K- in %D-grafov. %K je graf, ki bolj občutljiv na spremembe v vrednosti cene in je izračun po spodnji formuli. Spremenljivka C predstavlja zadnjo zaključno vrednost cene delnice, L14 predstavlja najnižjo vrednost cene delnice v zadnjih 14 dnevih (ali tednih) in H14 predstavlja najvišjo vrednost cene delnice v zadnjih 14 dnevih (ali tednih).

Drugi graf pa predstavlja drsečo sredino za časovno obdobje treh period in se zato imenuje počasni stohastični oscilator. Če drseči sredini dodamo dodatno obdobje treh period, bo graf posledično bolj gladek, zato ga imenujemo počasni stohastični oscilator. Stohastičen oscilator se nahaja na območju med 0 in 100 z določenima mejama 20 in 80. Tako kot pri RSI-indikatorju tudi stohastični oscilator zazna, ko prihaja do precenjenosti ali podcenjenosti. Če se graf %D nahaja nad 80, potem je treba prodati delnice, medtem ko če se graf %D nahaja pod 20, je treba delnice kupiti.[12]



Slika 3.5: 14-dnevni stohastični oscilator

Poglavje 4

Program za napovedovanje cen vrednostnih papirjev

4.1 Pregled uporabljenih tehnologij

4.1.1 Python

Program sem izdelal v programskem jeziku Python [13], različice 2.7.2. Python je objektno usmerjen visoko nivojski programski jezik, ki je neodvisen od strojne in programske opreme. Programski jezik ima avtomatizirano delo s pomnilnikom in dinamične podatkovne tipe. Za Python sem se odločil, ker ima preprosto sintakso in veliko dobrih zunanjih knjižnic, ki sem jih uporabil pri izdelavi programa.

4.1.2 NumPy

Numpy [10] je odprtokodna Python knjižnica za znanstveno računanje. Numpy je enostavna in hitra za uporabo. Pri obdelavi velike količine podatkov je zelo pomembno, da so funkcije dobro in hitro implementirane, saj vsaka nepotrebna operacija vpliva na končni čas računanja. Pri Numpy sem uporabil funkciji `polyfit()` in `poly1d()` za računanje regresije.

4.1.3 Scikit

SciKit [14] je Python knjižnica, ki se uporablja za strojno učenje. Scikit vsebuje veliko funkcij za strojno učenje in grafično predstavitev. Pri strojnem učenju je pomembno, da so funkcije čim bolj optimizirane, da se zagotovi hitro in zanesljivo delovanje. Pri Scikit sem uporabil funkcijo `mse()`, ki sem jo nato korenil, da sem dobil želeno vrednost RMSE (root mean square error).

```
from sklearn.metrics import mean_squared_error as mse
```

4.1.4 Matplotlib

Matplotlib [7] je Python knjižnica, ki se uporablja za risanje grafov, histogramov in ostalih matematičnih 2D-grafik. Matplotlib poskuša enostavne stvari grafično prikazati in zahtevne stvari narediti čim lažje. Matplotlib uporabljam za izris polinomske regresije in grafični prikaz podatkov.

```
from matplotlib import pyplot as plt
```

4.1.5 Xlrd

Xlrd[18] je Python knjižnica za uvod in branje `.xml`-datotek. Xlrd je enostavna knjižnica, ki omogoča branje, pisanje in ustvarjanje `.xml`-datotek.

```
import xlrd
```

4.2 Program

V spodnjih podpoglavjih bom opisal delovanje posameznih funkcij, ki so del programa. Program je razdeljen na šest funkcij oziroma delov, kjer se v vsakem delu programa izvede pomemben del napovedovanja cen vrednosti delnic. Na koncu programa je koda, ki zažene posamezne funkcije za izračun končnih napovedi.

4.2.1 Branje podatkov

S funkcijo `podatki()` program prebere podatke in jih vrne v obliki slovarja, kjer je ključ kratica podjetja, vrednost posameznega ključa pa je tabela vrednosti delnic. Program vrne dva različna slovarja. Tabela slovar vsebuje vrednosti delnic, tabela `slovarSt` pa vsebuje število delnic v skladu.

S spletne strani `yahoo.finance.com` [19] smo prenesli Excelovo datoteko, ki vsebuje seznam vseh delnic, ki so v portfelju sklada SPDR 500 (SPY) [15]. Za branje Excelove datoteke bomo uporabili knjižnico `xlrd`, ki je namenjena branju in zapisovanju Excelovih datotek. Funkcija najprej prebere Excelovo datoteko in v posamezne tabele shrani kratico podjetja (`imeDelnice[]`), lokacijo, kjer je posamezna `.csv`-datoteka z vrednostmi delnic (`potDelnice[]`), in število delnic, ki jih ima sklad (`stDelnic[]`).

Ko imamo potrebne podatke shranjene v tabelah, s pomočjo `csv`-bralnika preberemo vrednosti delnic, ki so zapisane v `.csv`-datotekah. Za branje `.csv`-datotek bomo uporabili knjižnico `csv`, ki je namenjena za obdelavo datotek tipa `.csv`. Podatki so v datoteki shranjeni tako, da je v prvi vrstici zapisana najmlajša vrednost, v zadnji vrstici pa je shranjena najstarejša vrednost. Na grafu želimo vrednosti podatkov predstaviti kronološko, zato se nam pojavi potreba, da zamenjamo vrsti red podatkov v tabeli, tako da bo prva vrednost v tabeli najstarejša in zadnja vrednost najmlajša. S tem bomo lahko vrednosti na grafu predstavili od leve proti desni, oziroma od najstarejše do najmlajše vrednosti. To rešimo tako, da s pomočjo pomožne tabele na koncu zamenjamo vrsti red vrednosti v tabeli in dobimo tabelo v želeni obliki. Na koncu želimo združiti oba slovarja s kraticami delnic, da bomo vedeli, katera vrednost pripada kateri delnici. Za združevanja bomo uporabili funkcijo `zip`, s pomočjo katere bomo v slovar z imenom `slovar` združili kratice delnic (`imeDelnic[]`) in vrednosti delnic (`vrednosti[]`), ter v `slovarSt` združimo kratice delnic in število delnic (`stDelnic[]`) in jih vrnemo kot rezultat funkcije.

Končni rezultat funkcije `podatki()` sta dva slovarja, v katerem prvi vsebuje število delnic posameznega podjetja, ki jih vsebuje sklad SPDR 500 (SPY) [15], drugi pa vsebuje vrednost delnic posameznega podjetja v izbra-

nem mesecu.

4.2.2 Urejanje podatkov

V enem izmed prvih korakov je treba določiti pravila, skozi katera moramo filtrirati podatke, preden jih uporabimo za testno in učno množico. Na podlagi pregleda porazdelitve števila vnosov pridobljenih podatkov sem ugotovil, da ima 98% podjetij v naboru podatkov vsaj 52 vnosa. Tako sem določil, da je minimalna meja za uporabo izbranega podjetja v napovedi vsaj 52 mesecev, kar pomeni, da mora podjetje biti delniška družba vsaj 52 mesecev.

V ta namen sem naredil funkcijo `odstraniDelnice()`, ki je namenjena odstranitvi podatkov, ki ne zadoščajo minimalnim standardom za končno napoved. Funkcija za vhodni parameter potrebuje slovar (dictionary) delnic in njihove vrednosti. Če je dolžina posamezne tabele znotraj slovarja manjša od 52 mesecev, program izpiše kratico podjetja, ki ne zadošča minimalnim standardom. Da sem lahko preveril dolžino tabele, sem moral tabelo spremeniti v posebno matematično (numpy) tabelo.

Ker je podjetij, ki ne dosegajo minimalnih standardov, zelo malo, sem zaradi optimizacije samo izpisal delnice in ročno popravil (odstranil delnice) `.xml`-datoteko.

4.2.3 Ustvarjanje učne in testne množice podatkov

V tej funkciji ustvarimo učno in testno množico podatkov, nad katerimi bomo kasneje izvajali polinomske regresije ter RMSE (root mean square error). Učno množico podatkov uporabimo za učenje polinomske regresije. To pomeni, da s pomočjo učne množice poiščemo optimalne parametre za polinomske regresije posamezne delnice. Podatke, ki so v testni množici podatkov, pa uporabljamo za preverjanje uspešnosti naše polinomske regresije, da preverimo uspešnost naših najdenih parametrov.

Funkcija vrne tabelo mesecev x , ki so predstavljeni s številkami (1 = prvi mesec, 2 = drugi mesec ...), tabelo y , ki je učna množica podatkov, in

ytestna, ki je tabela testnih podatkov.

Program najprej izračuna velikost učne in testne tabele ter ju nato napolni s podatki iz slovarja. Velikost testne in učne množice je odvisna od časovnega obdobja, v katerem bomo spremljali našo napoved. Če bomo napovedovali za eno leto vnaprej, potem bomo v učni množici uporabili 40 vnosov, v testni pa 12 vnosov, medtem ko bomo pri napovedovanju enega meseca uporabili 51 podatkov v učni množici in en vnos v testni množici. Program s preprosto zanko napolni tabelo mesecev in jo pretvori v posebno matematično tabelo za lažje kasnejše urejanje tabel. Funkcija kot rezultat vrne vse tri tabele v obliki tipa `np.array`.

4.2.4 Polinomska regresija

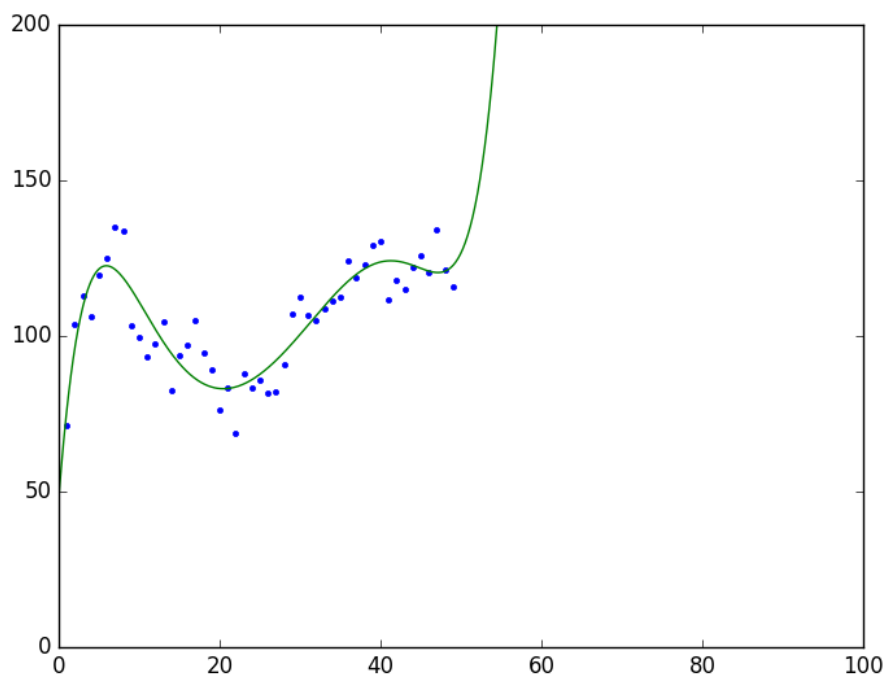
Funkcijo `regresija()` uporabljamo za izračun polinomske regresije. Funkcija kot vhodni parameter sprejme tabelo mesecev, tabelo učnih vrednosti, tabelo testnih vrednosti in stopnjo polinoma, vrne pa testne ter nove napovedane podatke.

Najprej v funkciji pretvorimo tabelo testnih vrednosti v matematično tabelo vrednosti s plavajočo vejico. Nato s funkcijama `polyfit()` in `poly1d()` ustvarimo polinomsko regresijo in jo pretvorimo v enodimenzionalen razred, da lahko nad temi podatki izvajamo ostale izračune. Potem sledi polnjenje tabele z napovedmi, kjer zapišemo vrednost y glede na vpisano vrednost x .

V funkciji imamo možnost tudi izrisa posameznih regresij, vendar je ta del kode blokiran, saj bi izrisali preveliko število grafov in bi imeli težave z zapiranjem programa. Možnost je dodana samo za izris poljubnih regresij in lažjo predstavitev 4.1.

4.2.5 Določanje stopnje polinoma

Naloga funkcije `stopnjaPolinoma()` je določitev najboljše stopnje polinoma, ki jo uporabimo za polinomsko regresijo. Funkcija ne sprejme nobenih vhodnih parametrov, na koncu pa vrne slovar, ki vsebuje kratice podjetij in njihove



Slika 4.1: Primer izrisa polinomske regresije

stopnje polinomov.

Funkcija se sprehodi čez slovar vseh delnic in za vsako posamezno podjetje preveri devet različnih stopenj. Stopnje sem omejil na devet, saj nimamo podatkov za 15 let in več, da bi bilo vredno testirati višje stopnje. Za vsako posamezno podjetje program naredi učno in testno množico podatkov ter izračuna MSE (mean squared error). Nato program preveri, ali obstaja boljši rezultat od trenutnega ter v pozitivnem primeru zapiše vrednost MSE (mean squared error) in stopnjo. Na koncu program koreni MSE (mean squared error), da dobimo RMSE (root mean square error), in vrne slovar, ki vsebuje stopnje polinoma za posamezne delnice.

Tabela 4.1: Primer RMSE (root mean square error) za različne stopnje polinomske regresije delnic podjetja Microsoft

stopnja polinoma	vrednost RMSE (root mean square error)
1	4.460
2	3.089
3	11.916
4	34.598
5	13.862
6	10.643
7	40.344
8	181.545
9	1278.178

4.2.6 Izpis rezultatov

Zadnji del programa je namenjen izpisu rezultatov. Program najprej razvrsti delnice glede na pričakovani zaslužek, in sicer najbolj dobičkonosna delnica je na začetku vrste in najmanj dobičkonosna na koncu, nato pa program odstrani delnice, ki imajo negativni zaslužek. Ko imamo vrstni red delnic določen, program enakomerno porazdeli denar med delnice, kar pomeni, da program enakomerno porazdeli denar med vse delnice, ki so imele po enem letu višjo vrednost. Na koncu program izračuna pričakovano vrednost in izpiše index velikosti sklada, ki je večji od sklada SPDR 500 (SPY)[15].

```
velikost sklada: 15 ||| vrednost sklada: 195125861706.850$  
velikost sklada: 20 ||| vrednost sklada: 199991249073.156$  
velikost sklada: 21 ||| vrednost sklada: 200849605699.442$  
velikost sklada: 22 ||| vrednost sklada: 200906443658.676$  
velikost sklada: 23 ||| vrednost sklada: 199205320465.667$  
velikost sklada: 24 ||| vrednost sklada: 198478580454.294$  
velikost sklada: 25 ||| vrednost sklada: 197524655720.806$  
velikost sklada: 26 ||| vrednost sklada: 198523843174.906$  
velikost sklada: 27 ||| vrednost sklada: 197448931192.554$  
velikost sklada: 28 ||| vrednost sklada: 198708313694.555$  
velikost sklada: 29 ||| vrednost sklada: 198098438901.558$  
velikost sklada: 30 ||| vrednost sklada: 198476581301.309$  
velikost sklada: 31 ||| vrednost sklada: 197717066416.159$  
velikost sklada: 32 ||| vrednost sklada: 196212086542.314$  
velikost sklada: 33 ||| vrednost sklada: 195073000667.433$  
velikost sklada: 34 ||| vrednost sklada: 194684866205.664$  
velikost sklada: 35 ||| vrednost sklada: 194951901014.297$  
velikost sklada: 36 ||| vrednost sklada: 193917162611.840$  
velikost sklada: 38 ||| vrednost sklada: 193581799088.536$  
velikost sklada: 39 ||| vrednost sklada: 193685984426.272$  
velikost sklada: 42 ||| vrednost sklada: 194251290083.953$
```

Slika 4.2: Končni izpis

Poglavje 5

Podatki

5.1 Opis

Cilj diplomske je narediti sklad, ki bo boljši od sklada SPDR 500 (SPY) ETF [15], in rezultate primerjati z metodami, ki jih uporabljajo strokovnjaki pri tehnični analizi vrednostnih papirjev. Za to nalogo bom potreboval delnice, ki jih vsebuje SPDR 500 (SPY), jih uredil in poskušal napovedati njihove čim bolj realne vrednosti.

5.2 Pridobivanje podatkov

Za začetek sem na spletni strani sklada SPDR 500 (SPY) [15] dobil .xml-datoteko, ki sem ji moral odstraniti prvih in zadnjih nekaj vrstic, saj so bile nepotrebne. Tako sem dobil celoten seznam delnic in število posameznih delnic.

Ko sem imel seznam delnic, sem s spletne strani Yahoo finance[19] prenesel .csv-datoteko, ki je vsebovala mesečno vrednost (vrednost delnice prvega v mesecu) posamezne delnice zadnjih pet let. Tako je vsaka delnica vsebovala 60 vrednosti. Za mesečni prikaz sem se odločil, ker bi se količina podatkov znatno povečala in bi imel težave z računanjem regresije, saj bi potreboval veliko močnejši procesor.

Name	Identifier	Shares Held
Apple Inc.	AAPL	57923108.000
Microsoft Corporation	MSFT	83128220.000
Exxon Mobil Corporation	XOM	43854390.000
Johnson & Johnson	JNJ	29089952.000
Amazon.com Inc.	AMZN	4092288.000
Facebook Inc. Class A	FB	24451900.000
General Electric Company	GE	97240904.000
Berkshire Hathaway Inc. Class B	BRK.B	19818292.000
AT&T Inc.	T	65106370.000
JPMorgan Chase & Co.	JPM	38680200.000
Wells Fargo & Company	WFC	48862310.000
Procter & Gamble Company	PG	28152728.000
Alphabet Inc. Class A	GOOGL	3105934.000
Alphabet Inc. Class C	GOOG	3123586.000
Verizon Communications Inc.	VZ	43111010.000
Pfizer Inc.	PFE	64143570.000
Chevron Corporation	CVX	19932884.000
Merck & Co. Inc.	MRK	29273934.000
Intel Corporation	INTC	49939772.000
Coca-Cola Company	KO	41178300.000
Home Depot Inc.	HD	13154456.000
Bank of America Corporation	BAC	108809980.000
Cisco Systems Inc.	CSCO	53194050.000
Comcast Corporation Class A	CMCSA	25569066.000

Slika 5.1: Del Excelove datoteke

5.3 Urejanje podatkov

Med pregledovanjem podatkov sem ugotovil, da sklad vsebuje nekaj podjetij, ki so mlajša od petih let. Ker je teh podjetij malo in ker skupaj predstavljajo manj kot 0.5% sklada, sem se odločil odstraniti teh 50 podjetij.

Zaradi odstranitve nekaj podjetij posledično nisem mogel uporabiti vrednosti sklada, ki je zapisana na spletni strani, zato sem sam izračunal novo vrednost sklada, ki ne vsebuje podjetij, mlajših od 58 mesecev. Končna vrednost sklada 1. 9. 2015 je tako znašala \$168,972,164,389.874, leto kasneje pa je vrednost narasla na \$193,521,569,800.803.

Da način izbire ni pristranski, sem preveril še donosnost, ki zdaj znaša 14,52%, pred obdelavo podatkov pa je znašala 14,41%, kar ni tako velika sprememba, ki bi močno vpliva na končne rezultate.

Preden pridemo do pomembnejšega odkritja v našem eksperimentu, je po-

membno, da lahko ponovimo svoje odkritje na večjem številu eksperimentov. Nekateri eksperimenti poskušajo kopirati vse pogoje eksperimenta, ostali pa spremenijo pogoje, ki niso relevantni za odkritje. Če s temi eksperimenti še vedno pridemo do enakih zaključkov, lahko rečemo, da je naš eksperiment dovolj robusten za potrditev naše teze. [20]

5.4 Analiza rezultatov

Rezultate bom razdelil v tri dele. V prvem delu bom predstavil rezultate, kjer sem napovedoval vrednosti od 1. 9. 2015 do 1. 9. 2016 in odstranil delnice, ki niso dobičkonosne. S tem načinom bom preveril uspešnost programa v daljšem časovnem obdobju. Tako bom poskusil dvigniti vrednost sklada samo z dobičkonosnimi delnicami in ugotovil, ali je program uspešen pri napovedovanju trenda delnic. V drugem delu bom primerjal rezultate, napovedane za isto časovno obdobje kot v prvem delu, le da bom vloženi denar enakomerno porazdelil med vse delnice. V zadnjem testu pa bom napovedoval vrednost delnic za zadnji mesec in tako preizkusil točnost programa v kratkem časovnem obdobju. S temi tremi testi bom preizkusil uspešnost programa v treh različnih scenarijih in ugotovil točnost napovedi pri različnih scenarijih delovanja.

5.4.1 Prvi del

V prvem delu sem napovedoval rezultate od 1. 9. 2015 do 1. 9. 2016. S seznama sem odstranil delnice, ki po napovedih v tem letu izgubijo vrednost. Delež delnic v portfelju sem pustil tako, kot ga ima sklad SPDR 500 (SPY), s čimer sem dobil sklad, ki vsebuje 302 različni delnici, in končno vrednost sklada \$184 milijard. Pričakoval sem višjo vrednost sklada, vendar sem ugotovil, da moram izboljšati porazdelitev delnic.

Tabela 5.1: Statistika prvega dela

opis	vrednosti
Število uporabljenih delnic	302
Število odstranjenih delnic	180
Vrednost sklada SPDR 500 (SPY) 1. 9. 2015	\$168.972.164.389,874
Vrednost sklada SPDR 500 (SPY) 1. 9. 2016	\$193.521.569.800,803
Vrednost napovedanega sklada 1. 9. 2016	\$183.987.846.139,663

5.4.2 Drugi del

V drugem delu sem tako kot v prvem odstranil vse delnice, za katere je polinomska regresija napovedala, da se jim v izbranem časovnem obdobju zmanjša vrednost. Razlika med prvim in drugim testom je v porazdelitvi vložnega kapitala, ki sem ga v drugem testu enakomerno porazdelil med vse dobičkonosne vrednostne papirje, kar pomeni, da sem v vse delnice vložil enako količino denarja. Rezultat je občutno boljši, saj sem povečal vrednost kapitala za kar \$9 milijard oziroma 4,9%.

Tabela 5.2: Statistika drugega dela

opis	vrednosti
Število uporabljenih delnic	302
Število odstranjenih delnic	180
Vrednost sklada SPDR 500 (SPY) 1. 9. 2015	\$168.972.164.389,874
Vrednost sklada SPDR 500 (SPY) 1. 9. 2016	\$193.521.569.800,803
Vrednost napovedanega sklada 1. 9. 2016	\$192.401.313.145,959
Količina denarja vložnega v delnico enega podjetja	\$561.369.316,91

5.4.3 Tretji del

V zadnjem delu sem napovedoval vrednost delnic samo v zadnjem mesecu, s čimer sem preizkusil uspešnost polinomske regresije v krajšem časovnem obdobju. Ker se napoved polinomske regresije slabša z velikostjo časovnega intervala, sem pričakoval, da bo tretja napoved najboljša. Podobno kot pri drugem delu sem enakomerno porazdelil denar med vse delnice. Moja pričakovanja so bila na mestu, saj sem vrednost sklada povečal še za dobre \$4 milijarde oziroma za 7,1 %, in tako dobro presegel dejansko realizirano vrednost sklada SPDR 500 (SPY).

Tabela 5.3: Statistika tretjega dela

opis	vrednosti
Število uporabljenih delnic	399
Število odstranjenih delnic	83
Vrednost sklada SPDR 500 (SPY) 1. 9. 2015	\$168.972.164.389,874
Vrednost sklada SPDR 500 (SPY) 1. 9. 2016	\$193.521.569.800,803
Vrednost napovedanega sklada 1. 9. 2016	\$196.633.499.356,004
Količina denarja vloženega v delnico enega podjetja	\$422.430.410,975

5.4.4 Povečanje dobička

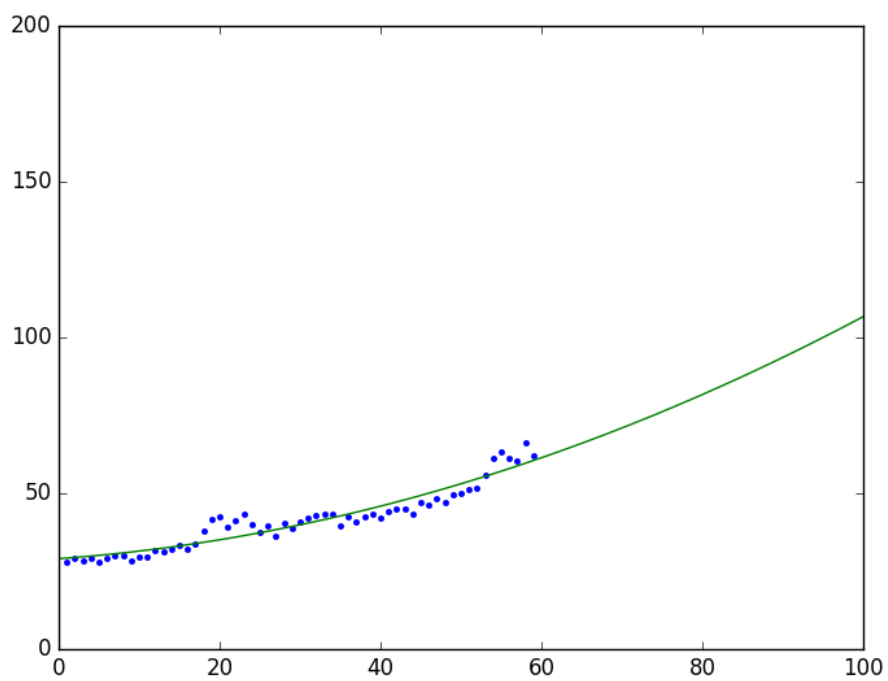
Na koncu sem poskušal poiskati najvišjo vrednost sklada tako, da sem z različnimi kombinacijami poskušal najti čim boljši sklad. Ker je količina podatkov prevelika, da bi poskusil vse možne kombinacije, sem pričakovane vrednosti delnic razvrstil od najbolj dobičkonosne do najmanj dobičkonosne. Na koncu sem sestavil sklad, ki ima 20 različnih delnic, skupna vrednost 1. 9. 2016 pa znaša kar \$209 milijard.

Moja končna ugotovitev je, da je polinomska regresija odlično napovedovala vrednosti za en mesec in slabše za eno leto. Napoved za eno leto je slabša

Tabela 5.4: Statistika polinomske regresije v primerjavi s skladom SPDR 500 (SPY)

Poiskus	Polinomska regresija
Prvi del	-4,91%
Drugi del	-0,6%
Tretji del	1,61%

tudi zato, ker regresija začne zelo hitro padati oziroma rasti in so rezultati posledično slabši.



Slika 5.2: Primer grafičnega prikaza polinomske regresije pri napovedi enega leta

Poglavje 6

Primerjava metod tehnične analize in polinomske regresije

V zaključku bom med seboj primerjal še napovedi za sklad. Med seboj bom primerjal indikatorje, ki sem jih opisal pri tehnični analizi, in svojo polinomsko regresijo. Za analizo sem uporabil vrednosti sklada zadnjih pet let, napovedoval pa bom za zadnji mesec, in sicer od 1. 8. 2016 do 1. 9. 2016.

6.1 Polinomska regresija

Iz svoje napovedi lahko razberem, da se je vrednost delnice dvignila. Poleg tega je vrednost RMSE (root mean square error) zelo nizka, kar pomeni, da ima krivulja zelo dobro ujemanje kljub visoki stopnji polinoma. Moja odločitev bi bila, da se splača vlagati v ta sklad.

6.2 Tehnična analiza

Enostavna drseča sredina se v obdobju od 1. 7. 2016 do 1. 8. 2016 enakomerno giblje, za kar lahko ugotovim, da je dober znak za vlaganje v sklad. Do podobne ugotovitve prihajam tudi pri eksponentni drseči sredini, ki se

Tabela 6.1: Polinomska regresija sklada

opis	vrednosti
Napovedana vrednost	\$219.860
RMSE (root mean square error)	2.480
Stopnja polinoma	9

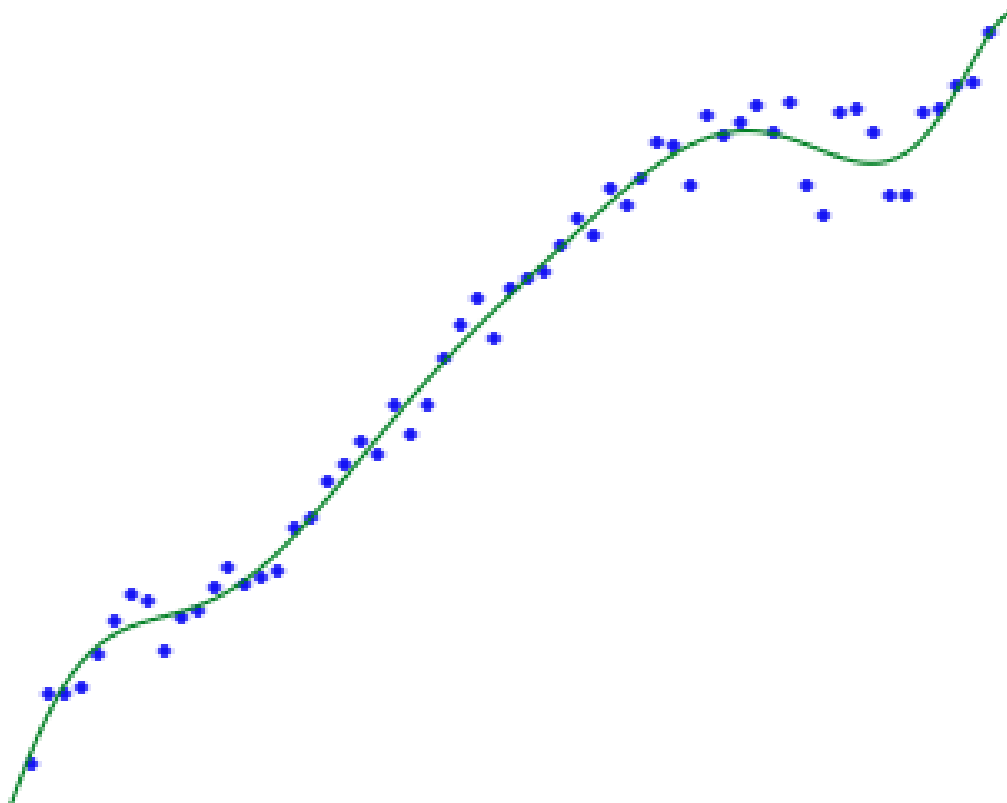
prav tako počasi vzpenja. MACD se dviguje, kar pomeni, da se prav tako splača vlagati v sklad. Indeks relativne moči se giblje zelo blizu vrednosti 70, vendar je nikoli ne prečka, zato menim, da je ta indikator nevtralen. Kot zadnji indikator pa sem uporabil stohastični oscilator, ki se prav tako kot RSI giblje blizu zgornje meje, vendar je nikoli ne prečka, zato ga puščam kot nevtralnega.

Glede na rezultate indikatorjev menim, da se splača vlagati v sklad na začetku avgusta, saj trije pozitivni in dva nevtralna rezultata prepričajo, da je sklad dobra investicija za naslednji mesec.

6.3 Primerjava

V obeh primerih sem prišel do ugotovitve, da se splača vlagati v sklad. Realnost je kljub vsemu malo drugačna, saj se je vrednost sklada v obdobju od 1. 8. 2016 do 1. 9. 2016 dvignila za samo \$0.17. Naše napovedi niso bile napačne, saj se je vrednost sklada dvignila, kljub vsemu pa smo pričakovali večje dvige skladov, saj sta tako polinomska regresija kot tehnična analiza bolj prepričljivo kazali na dvig vrednosti sklada. To lahko razberemo s slike 6.2, kjer sem na enem grafu predstavil vse indikatorje tehnične analize, ki sem jih uporabljal za izvedbo tehnične analize. Iz tega lahko ugotovimo, da je gibanje vrednosti delnic zelo nepredvidljivo, čeprav so indikatorji pri dvigu vrednosti zelo suvereni.

V primerjavi sem želel bolj podrobno primerjati tehnično analizo s polinomske regresije na vseh delnicah, vendar je zaradi velike časovne zahtevnosti



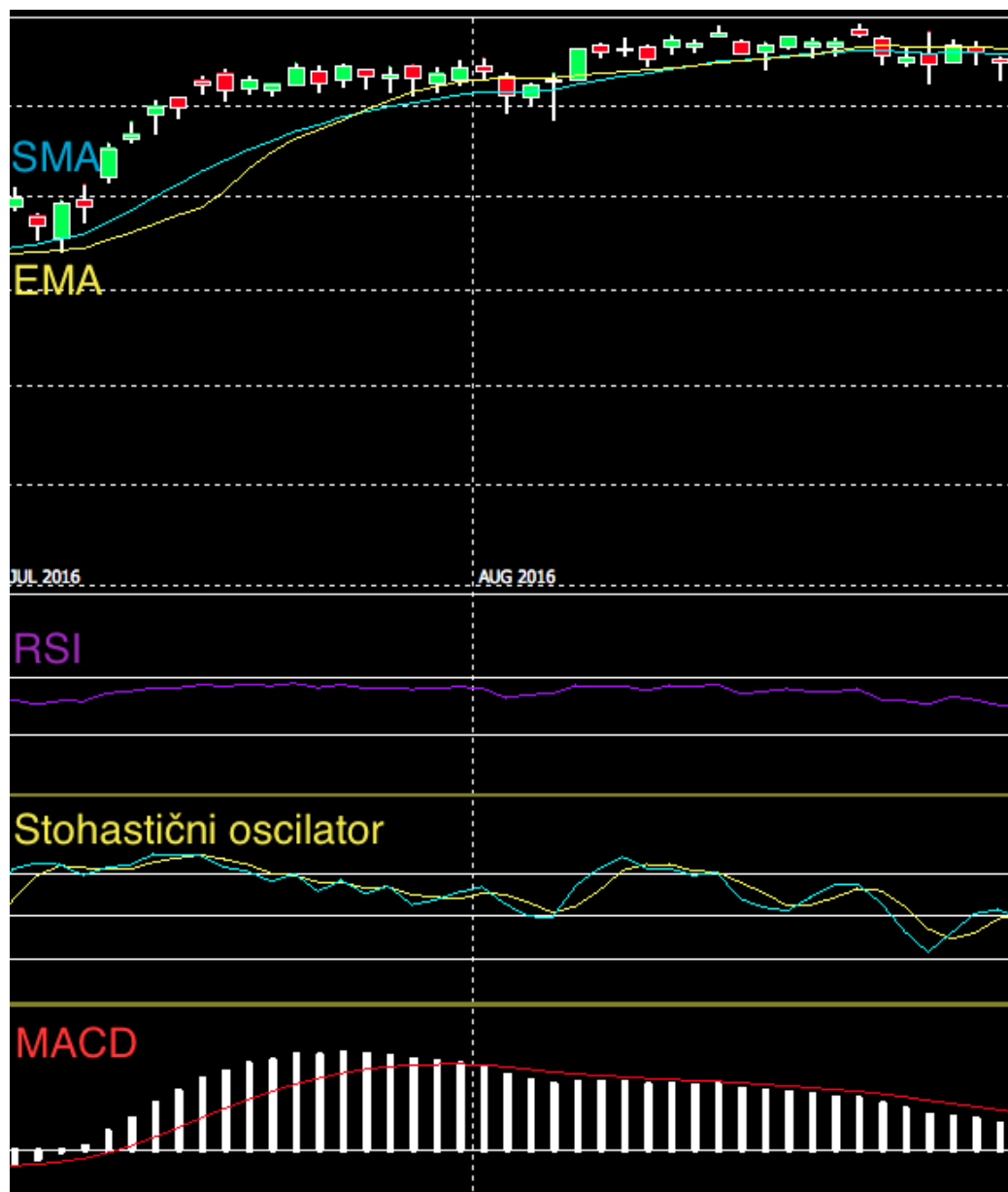
Slika 6.1: Polinomska regresija sklada

to neizvedljivo, saj tehnična analiza temelji na analizi grafov, ki je časovno zelo potratna. Zaradi te omejitve predstavlja lastnost polinomske regresije, da avtomatično analizira in predlaga delnice, ki jim bo vrednost padla ali se dvignila, precejšnjo primerjalno prednost. S polinomske regresije lahko tako v določenem času izvedemo več 1000 analiz delnic, medtem ko s tehnično analizo lahko v istem času analiziramo največ eno ali dve različni delnici. Prav tako različnost metod tehnične analize otežuje razumevanje in interpretacijo rezultatov, saj so nekatere metode zelo sofisticirane ter zahtevne za uporabo.

Vendar se prednost tehnične analize pokaže pri različnih uporabah njenih metod, saj lahko te analize naredimo bolj podrobne in bolj natančne,

prav tako se zanaša na več različnih vrednosti, ki jih polinomska regresija ne upošteva. V diplomskem delu sem opisal samo nekaj metod tehnične analize, ki so največkrat uporabljene, medtem ko jih obstaja še veliko več, ki so težje za uporabo.

Med primerjavo sem ugotovil, da ne obstaja optimalna izbira metod za analizo v vseh pogojih. Če uporabnik želi natančno in zanesljivo analizo za največ nekaj deset različnih vrednostnih papirjev, bi na podlagi analiz, ki sem jih izvedel, uporabil metode tehnične analize. Kot dodaten faktor pri odločitvi o prodaji ali nakupu vrednostnega papirja bi uporabil še polinomske regresije, s katero bi dobil hitro potrditev. Če pa uporabnik želi izvesti analizo za večje količine različnih vrednostnih papirjev in mu je obdelava večje količine različnih vrednostnih papirjev bolj pomembna od same natančnosti, pa bi izbral polinomske regresije.



Slika 6.2: Indikatorji tehnične analize sklada

Poglavje 7

Zaključek

V delu sem odgovoril na vsa v uvodu zastavljena vprašanja. Pri prvem cilju sem se spraševal, katere metode poslovne inteligence so primerne za analizo vrednosti gibanja cen vrednostnih papirjev, za kar sem poiskal primeren članek, ki je med seboj primerjal različne regresijske metode poslovne inteligence za napovedovanje cen vrednostnih papirjev. V drugem cilju sem želel ugotoviti, katere metode tehnične analize so najbolj uveljavljene, zato sem v tretjem poglavju opisal in preizkusil različne metode tehnične analize. Odgovor na tretji in četrti cilj sem rešil s primerjavo metod tehnične analize ter polinomske regresije za napovedovanje vrednosti sklada SPDR 500 (SPY). Metode sem primerjal pri napovedovanju v različnih časovnih obdobjih, saj sem tako lahko razkril pozitivne in negativne lastnosti posameznih metod.

Zaradi obsega področja sem svojo raziskavo v diplomskem delu omejil z nekaj osnovnimi omejitvami. Raziskava je bila omejena na podjetja, ki so prisotna na ameriški borzi vsaj 52 mesecev. To omejitev sem postavil, ker sem potreboval dovolj velik vzorec podatkov za učenje polinomske regresije. Med raziskovanjem podatkov sem ugotovil, da je raziskava omejena na časovno obdobje, ki se dogaja po končani recesiji. S to omejitvijo sem zanemaril vpliv večjih gospodarskih sprememb na trgu, zato predpostavljam, da napovedni model, ki napoveduje v daljšem časovnem obdobju, ne bi napovedoval relevantnih napovedi.

Diplomsko delo lahko razširimo z implementacijo novih algoritmov za napovedovanje vrednosti cen vrednostnih papirjev ali pa z izbiro drugačne metode napovedovanja. Diplomsko delo je možno razširiti tako, da bi namesto točne vrednosti vrednostnega papirja napovedoval dvig ali padec delnice. S tem načinom bi lahko uporabil še druge metode, kot je klasifikacija. Diplomsko delo pa bi lahko razširil tako, da bi vrednost delnice napovedoval na podlagi zunanjih dejavnikov, kot so svetovne spremembe v gospodarstvu, spremembe v svetovni politiki in ostali podobni dejavniki. S tem bi vsebino dela močno nadgradil in izboljšal natančnost napovedi, vendar je zahtevnost problema izjemno velika.

Literatura

- [1] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4):1165–1188, 2012.
- [2] Ciro Donalek. Supervised and unsupervised learning. In *Astronomy Colloquia. USA*, 2011.
- [3] Spletna stran z opisom eksponentne drseče sredine. <http://www.investopedia.com/terms/e/ema.asp>.
- [4] Introduction to feature selection methods with an example (or how to select the right variables?). <https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select/>
- [5] Bogdan Filipič. Strojno učenje in odkrivanje znanja v podatkovnih bazah (data mining). 2002.
- [6] Spletna stran z opisom enostavne drseče sredine. https://en.wikipedia.org/wiki/Moving_average.
- [7] Matplotlib python knjižnica. <http://matplotlib.org/>.
- [8] Big data: The next frontier for innovation, competition, and productivity. <http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation>.

- [9] Solomon Negash. Business intelligence. 2004.
- [10] Scipy python knjižnica. <http://www.numpy.org/>.
- [11] Lucas Nunno. Stock market price prediction using linear and polynomial regression models. Technical report, Technical report, University of New Mexico, 2014.
- [12] Gašper Perme. Tehnična analiza vrednostnih papirjev in njihovih trgev, 2004.
- [13] Python programski jezik. <https://www.python.org>.
- [14] Scikit python knjižnica. <http://scikit-learn.org/>.
- [15] Spletna stran sklada spdr sp 500. <https://www.spdrs.com/>.
- [16] Spletna stran z opisom tehnične analize. <http://www.investopedia.com/terms/t/technicalanalysis.asp>.
- [17] John W Tukey. Exploratory data analysis. 1977.
- [18] Xlrd python knjižnica. <https://pypi.python.org/pypi/xlrd>.
- [19] Spletna stran yahoo finance. <http://finance.yahoo.com/>.
- [20] Robert K Yin. *Case study research: Design and methods*. Sage publications, 2013.

Poglavje 8

Priloga

```
def podatki():
    #####BRANJE EXCELOVE DATOTEKE#####
    book = xlrd.open_workbook("data/spy.xls")
    sh = book.sheet_by_index(0)
    imeDelnice=[]
    potDelnice=[]
    stDelnic=[]
    for rx in range(sh.nrows):
        imeDelnice.append(sh.cell_value(rowx=rx, colx=1))
        potDelnice.append
            ("data/"+sh.cell_value(rowx=rx, colx=1)+".csv")
        stDelnic.append(sh.cell_value(rowx=rx, colx=4))
        #print sh.cell_value(rowx=rx, colx=1)
    #print imeDelnice[100]

    #####BRANJE VREDNOSTI DELNIC#####
    vrednosti=[]
    for i in potDelnice:
        with open(i) as csvfile:
            f=csv.reader
```

```

        (csvfile , delimiter=',', quotechar='"')
    tmp=[]
    a=0
    for row in f:
        if(a!=0):
            tmp.append(row[6])
            a=a+1
    vrednosti.append(list(reversed(tmp)))

slovarSt=dict(zip(imeDelnice , stDelnic))
slovar=dict(zip(imeDelnice , vrednosti))
return slovar , slovarSt

```

```

def odstraniDelnice(slovar):
    ###Izpiše podjetja mlajša od 52 mesecev###
    for i in slovar:
        vrednosti=slovar[i]
        vrednosti=np.asarray(vrednosti , dtype=float)
        if(len(vrednosti)<52):
            print i

```

```

def testnaUcnaMnozica(slovar , ime):
    ###Ustvari ucno in testno mnozico###
    y=[]
    y_testna=[]
    velikost=len(slovar[ime])-24
    a=1
    for i in slovar[ime]:
        if(a==velikost+13):
            break
        if(a<=velikost):
            y.append(i)

```

```

        else:
            y_testna.append(i)
            a+=1
x=[]
a = 0
for i in y:
    a=a+1
    x.append(a)
x=np.asarray(x, dtype=float)
y=np.asarray(y, dtype=float)
return x,y,y_testna

```

```

def regresija(x,y,y_testna , stopnja):
    ###naredi in izrisi polinomsko regresijo###
    y_testna=np.asarray(y_testna , dtype=float)
    z= np.polyfit(x,y, stopnja)
    p = np.poly1d(z) #naredi polinom
    y_napoved=[]
    for i in range(1,13):
        y_napoved.append(p(len(y)+i))
    y_napoved=np.asarray(y_napoved , dtype=float)
    '''
    x=np.linspace(0 , 100 , 1000)
    plt.plot(x, y, ' . ' , x, p(x) , ' - ' )
    plt.ylim(0,200)
    plt.show()
    '''
    return y_testna , y_napoved

```

```

def stopnjaPolinoma():
    #Vrne stopnjo polinoma in mse
    slovar , slovarStopnja=podatki()

```

```
for j in slovar:
    stopnja=1
    minMse=100000
    for i in range(1,10):

        x,y,y_testna=testnaUcnaMnozica(slovar,j)
        y_testna,y_napoved=regresija(x,y,y_testna,i)
        tmp=mse(y_testna,y_napoved)
        if (tmp<minMse):
            stopnja=i
            minMse=tmp

    skupaj=[]
    skupaj.append(stopnja)
    skupaj.append(math.sqrt(minMse))
    slovarStopnja[j]=skupaj
return slovarStopnja
```

```
seznamPolinomov=stopnjaPolinoma()
a=vrednostDelnic(seznamPolinomov)
mseVrednosti={}
for i in seznamPolinomov:
    mseVrednosti[i]=seznamPolinomov[i][1]
import operator

slabi={}
b={}
print len(a)
for i in a:
    if(a[i][11]<0):
        b[i]=a[i]
        slabi[i]=b[i]
```

```
for i in b:
    del a[i]
    del mseVrednosti[i]
b={}
for i in a:
    a[i]=a[i][11]-a[i][0]
    if(a[i]<0):
        b[i]=a[i]
        slabi[i]=b[i]

for i in b:
    del a[i]
    del mseVrednosti[i]

podatki1 ,podatki2=podatki()
kap1 ,kap2=vrednostSPY()
st=0
max=0
vrednost=0
for i in slabi:
    tmp=len(podatki1[i])
    vrednost=vrednost
    +float(podatki1[i][tmp-13])*float(podatki2[i])

for i in a:
    tmp=len(podatki1[i])
    vrednost=vrednost
    +float(podatki1[i][tmp-1])*float(podatki2[i])
sortedMse=sorted(mseVrednosti.items(),
key=operator.itemgetter(1))
```

```
sortedVrednosti=sorted(a.items() ,
key=operator.itemgetter(1),reverse=True)

for i in range(1,300,1):
    velikostSklada=i
    kapital2015=kap1
    kapital2015=kapital2015/velikostSklada
    konec=0
    for i in range(0,velikostSklada,1):
        j=len(podatki1[sortedVrednosti[i][0]])
        vrednostDelnice=np.asarray
        (podatki1[sortedVrednosti[i][0]][j-13],
        dtype=float)
        stDelnic=kapital2015/vrednostDelnice
        vrednostDelnice=np.asarray
        (podatki1[sortedVrednosti[i][0]][j-1],
        dtype=float)
        konec=konec+stDelnic*vrednostDelnice
    if(konec>kap2):
        print(" velikost _sklada : _{0} _|||
        _vrednost _sklada : _{1:.3f}$" .format(i,konec))
```