

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Rok Fortuna

**Sistem za avtomatizirano trgovanje z
uporabo strojnega učenja, rudarjenja
podatkovnih tokov in tehnične analize
trgovanja**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJ RAČUNALNIŠTVA IN
MATEMATIKE

MENTOR: prof. dr. Matjaž B. Jurič

SOMENTOR: doc. dr. Lovro Šubelj

Ljubljana 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Preučite in analizirajte avtomatske trgovalne sisteme in opredelite njihovo zgradbo. Opredelite osnovne pojme tehnične analize trgovanja ter analizirajte možnosti uporabe strojnega učenja pri avtomatskih trgovalnih sistemih. Identificirajte in opišite najpomembnejše algoritme. Zasnуйте in implementirajte lasten sistem za avtomatsko trgovanje, ki uporablja izbrane tehnike strojnega učenja. Zasnovan avtomatski trgovalni sistem naj se čim bolj približa realnim razmeram na trgu. Ovrednotite uspešnost napovedovanja na realnih tržnih podatkih kriptovalut in na podlagi pridobljenega znanja predstavite možne izboljšave sistema.

IZJAVA O AVTORSTVU ZAKLJUČNEGA DELA

Spodaj podpisani Rok Fortuna, z vpisno številko 63120229, sem avtor zaključnega dela z naslovom:

Sistem za avtomatizirano trgovanje z uporabo strojnega učenja, rudarjenja podatkovnih tokov in tehnične analize trgovanja

IZJAVLJAM

1. da sem zaključno delo študija izdelal samostojno pod mentorstvom prof. dr. Matjaža B. Juriča in somentorstvom doc. dr. Lovra Šublja;
2. da je tiskana oblika pisnega zaključnega dela študija istovetna elektronski obliki pisnega zaključnega dela študija;
3. da sem pridobil/-a vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v pisnem zaključnem delu študija jasno označil/-a;
4. da sem pri pripravi pisnega zaključnega dela študija ravnal/-a v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil/-a soglasje etične komisije;
5. soglašam, da se elektronska oblika pisnega zaključnega dela študija uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
6. da na UL neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja pisnega zaključnega dela študija na voljo javnosti na svetovnem spletu preko Repozitorija UL;
7. dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija.

V Ljubljani, dne 23. julija 2016

Podpis študenta/-ke:

Zahvaljujem se za vodenje in vzpodbudo mentorjema, prof. dr. Matjažu B. Juriču in doc. dr. Lovru Šublju. Obenem bi se zahvalil še svoji družini za podporo, ki mi jo je nudila v vseh letih študija in v času pisanja diplome.

Svoji dragi Maji.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Opis problema	1
1.2	Sodobno trgovanje	3
1.3	Motivacija	4
1.4	Cilji naloge	5
2	Avtomatski trgovalni sistemi	7
2.1	Uvod v avtomatske trgovalne sisteme	7
2.2	Digitalne borze	7
2.3	Zgradba ATS	10
3	Tehnična analiza trgovanja	17
3.1	Osnovni pojmi tehnične analize	18
3.2	Trgovalni indikatorji	20
4	Strojno učenje in trgovanje	25
4.1	Osnovno o strojnem učenju	25
4.2	Paradigme strojnega učenja	26
4.3	Trgovanje z vidika strojnega učenja	31
4.4	Algoritem k -najbližjih sosedov	32
4.5	Naivni Bayesov klasifikator	34

KAZALO

4.6	Umetne nevronske mreže	35
5	Zasnova in implementacija lastnega ATS	43
5.1	Zgradba našega ATS	43
5.2	Implementacija simulacije ATS	47
6	Rezultati in diskusija	51
6.1	Načini ovrednotenja uspešnosti ATS	51
6.2	Rezultati	53
6.3	Diskusija	57
7	Zaključek	61
8	Dodatek	63
8.1	Razširjeni rezultati	63

Seznam uporabljenih kratic

kratica	angleško	slovensko
GUI	graphical user interface	grafični uporabniški vmesnik
ATS	automated trading system	sistem za avtomatsko trgovanje
HTML	HyperText Markup Language	označevalni jezik
API	application programming interface	programski vmesnik
ECN	Electronic Communication Network	tržna mreža
REST	representational state transfer	reprezentativni prenos stanja
JSON	JavaScript Object Notation	Javascript objektni zapis
HTTP	Hypertext Transfer Protocol	spletni protokol
kNN	<i>k</i> -nearest neighbours	algoritem <i>k</i> -najbližnjih sosedov
ANN	artificial neural network	umetna nevrnska mreža

Povzetek

Digitalno trgovanje z dobrinami (delnicami, valutami itd.) danes močno izpodriva klasično trgovanje, saj se je veliko borz preselilo v oblak. Računalnik poskrbi za izmenjavo dobrin, hkrati pa lahko tudi samostojno trguje. Avtomatski trgovalni sistem je računalniški sistem, ki trguje z dobrinami brez posredovanja človeka. Tak pristop poleg objektivnosti in empiričnosti odločitev omogoča hitre izvršitve kupčij, ki so ključne za uspeh. V diplomski nalogi raziskujemo avtomatske trgovalne sisteme in njihov način delovanja. Naredimo pregled področja tehnične analize trgovanja, ki se ukvarja s kvantificiranjem gibanja cen na trgu. Definiramo problem trgovanja z vidika nadzorovanega strojnega učenja in rudarjenja podatkovnih tokov. S področja strojnega učenja posebej izpostavimo algoritem k -najbližjih sosedov, umetne nevronske mreže in naivni Bayesov klasifikator. S pomočjo omenjenih algoritmov in podatkov, pridobljenih z metodami tehnične analize trgovanja, zasnujemo avtomatski trgovalni sistem. S simulacijo ga ovrednotimo na realnem gibanju cen kriptovalute Bitcoin in kriptovalute Litecoin. Avtomatsko trgovanje se z vidika strojnega učenja izkaže za zelo zahteven problem. Kljub temu nam uspe, z uporabo algoritma k -najbližjih sosedov in umetnih nevronskih mrež, doseči zadovoljivo napovedno uspešnost in posledično profitabilnost.

Ključne besede: avtomatski trgovalni sistemi, strojno učenje, umetne nevronske mreže, k -najbližjih sosedov, naivni Bayesov klasifikator, tehnična analiza trgovanja.

Abstract

Digital trading of securities is beginning to dominate over classical trading and the trading exchanges are rapidly migrating to the cloud. Computer is not only present in the exchange process but is also capable of making trading decisions on human's behalf. Automated trading system is a computer system, capable of trading without human interaction. The benefits of such an approach to trading are objectivity and fast execution of orders, which are often crucial for success. In this thesis we examine automated trading systems and their structure. We study the field of technical analysis which quantifies market price movements. We define trading as a supervised machine learning and stream mining problem and examine the k -nearest neighbours algorithm, naive Bayes classifier and artificial neural networks. Based on our research we design an automated trading system. We evaluate its performance on actual market data of cryptocurrencies Bitcoin and Litecoin using a simulated environment. Automated trading turns out to be a difficult machine learning problem, but with the use of the k -nearest neighbours algorithm and artificial neural networks we manage to achieve decent success in our predictions and profitability.

Keywords: automated trading systems, machine learning, artificial neural networks, k -nearest neighbours, naive Bayes classifier, technical analysis.

Poglavje 1

Uvod

1.1 Opis problema

Vlaganje kapitala je dejavnost, pri kateri posameznik ali organizacija svoj denar vloži v neko dobrino v upanju, da v prihodnosti njena cena zraste. Navadno so te dobrine delnice, nepremičnine, valute, lahko pa tudi surovine (drage kovine, nafta itd.). Glede na čas trajanja naložb (čas od nakupa do prodaje dobrine), ločimo dva poglavitna načina vlaganja kapitala: investiranje in trgovanje. Investiranje je nakup dobrine za daljši čas, tudi več let ali celo desetletij. Investitor dobrino kupi, ker verjame v njeno dolgoročno vrednost in se ne zanima za kratkotrajna nihanja njene cene. Na drugi strani je trgovanje precej aktivnejša oblika vlaganja kapitala. Trgovalec kupi dobrino za krajše časovno obdobje (lahko od par minut do par dni) in izkorišča nihanja njene cene, ki ga povzročajo ostali udeleženci na trgu. Ciljni dobiček kupčije pri trgovanju je bistveno manjši v primerjavi z investiranjem, vendar se obrestuje na krajši rok.

Strategije trgovanja se razlikujejo glede na način sprejemanja trgovalnih odločitev.

- **ustvarjanje trga:** trgovalec izkorišča majhne razlike prodajne in nakupne cene v kombinaciji z majhnimi nihanji trga,
- **arbitraža:** trgovalec izkorišča razlike cen istih dobrin na različnih digitalnih borzah,
- **špekulacijsko trgovanje:** trgovalec napoveduje ceno dobrine v prihodnosti

Za trgovanje na prva dva načina je ključna hitrost trgovalnih odločitev. Oba načina namreč izkoriščata anomalije na trgu, ki so prisotne le kratek čas. Za profitabilen izkoristek teh anomalij je potrebna draga strojna oprema, saj smo le na ta način konkurenčni velikim trgovalnim podjetjem. Slednja namreč ustvarjajo konkurenčnost na trgu in zmanjšujejo čas prisotnosti takih anomalij. Za samostojne trgovalce in tudi marsikatera manjša podjetja pa je taka strojna oprema težko dosegljiva.

Na drugi strani špekulacijsko trgovanje ne zahteva tolikšne hitrosti pri trgovalnih odločitvah. Gre namreč za vrsto trgovanja, ki želi znana dejstva o dobrini izkoristiti za napoved njene vrednosti v prihodnosti. Špekulacijsko trgovanje, če se ga lotimo na sistematičen način, sprejema trgovalne odločitve na podlagi prej definiranih pravil brez človekovih subjektivnih odločitev. Deli se na dve širši med seboj zelo različni paradigmi: tehnično analizo trgovanja in temeljno analizo trgovanja. Tehnična analiza trgovanja se ukvarja predvsem z merjenjem gibanja cen dobrin. S pomočjo glavnih dveh podatkov: cene in velikosti transakcij, si želi ustvariti sliko o situaciji na trgu. Želi napovedati čas trajanja trendov, obrate, skoke cen itd. Temeljna analiza trgovanja pa se osredotoča predvsem na podatke o dobrini, s katero trguje. V primeru delnic podjetja preučuje dobičke podjetja, njegovo perspektivnost, premoženje, naložbene odločitve, splošno ekonomsko stanje na trgu itd. Njen glavni cilj je podati mnenje o vrednosti podjetja oz. dobrine in na podlagi tega sprejemati trgovalne odločitve.

1.2 Sodobno trgovanje

Sodobno trgovanje se precej razlikuje od načina trgovanja v preteklosti. V preteklosti je bilo trgovanje precej manj dinamično. Pri trgovanju z delnicami je moral biti posameznik fizično prisoten na borzi, kjer je svoj denar zamenjeval za delnice in obratno. Trgovanje je bilo centralizirano v borznih hišah, ki so za usluge posrednika pri kupčijah zaračunavale pristojbine. Pri trgovanju z valutami je bila situacija podobna. Posameznik je moral nakupe tujih valut urejati v menjalnicah ali bankah, kjer je moral biti fizično prisoten. Danes pa so se borzne in bančne storitve v večini preselile v oblak.

Digitalne borze delnic in valut so spletne platforme, ki omogočajo posamezniku, da svoja sredstva naloži na svoj spletni račun, s katerim prek spletnega uporabniškega vmesnika trguje. Slednji mu omogoča oddajanje svojih ponudb in sprejemanje ponudb drugih uporabnikov digitalnih borz. Ko uporabnik ponudbo odda, se ta zabeleži v zaledni sistem digitalne tržnice in je nato prek uporabniškega vmesnika na voljo ostalim udeležencem trga. Ko trgovelec sprejme ponudbo ali pa nekdo sprejme njegovo ponudbo, digitalna tržnica poskrbi za izmenjavo dobrin. Digitalna izmenjava dobrin je spremenila veliko vidikov trgovanja. V nadaljevanju so navedene spremembe, ki so jih prinesle digitalne borze.

- hitrejša izvedba transakcij - digitalna izmenjava dobrin,
- manjše pristojbine - ni več osebja na borzi, ki bi skrbelo za izmenjavo dobrin,
- večja likvidnost (večje število transakcij, večji pretok kapitala),
- lažja dostopnost trga in s tem večja konkurenca,
- večja transparentnost transakcij,
- manjši razponi cen (razlika med prodajno in nakupno ceno).

Med ključnimi spremembami so tudi razširjene možnosti trgovanja. Poleg grafičnih uporabniških vmesnikov (GUI) mnoge spletne tržnice izpostavljajo

spletne storitve. Prek njih lahko iz programske kode preko REST protokola pridobivamo podatke o trgu in trgujemo. To nam omogoča avtomatizirati postopek trgovanja. S pridobivanjem podatkov iz digitalnih tržnic in njihovo obdelavo lahko zgradimo avtomatski trgovalni sistem (ATS), ki ob izpolnitvi določenih pogojev sklepa kupčije.

1.3 Motivacija

Trgovanja se lahko lotimo na več načinov s pomočjo različnih tehnik. Pri tem si želimo čim večje sistematičnosti in čim bolj minimizirati naključnost trgovanja ter posledično tudi finančno tveganje. Ključnega pomena je pred uporabo našo strategijo testirati, kako se je obnašala v preteklosti in jo prilagoditi do njene optimalnosti. V ta namen se lahko poslužimo številnih znanih ekonomskih pristopov (bodisi tehnične analize ali pa temeljne analize trgovanja), med katerimi moramo izluščiti tiste, ki imajo največjo informacijsko vrednost, da smo na trgu konkurenčni. Pri tem so nam najbolj v oporo področja znanosti, kot sta strojno učenje in statistika.

1.4 Cilji naloge

V diplomski nalogi raziščemo avtomatske trgovalne sisteme in njihovo zgradbo. Preučimo področje tehnične analize trgovanja in metod, ki jih uporablja. Naredimo pregled osnov strojnega učenja in izpostavimo algoritme, ki so bili v preteklosti uspešno uporabljeni v namene trgovanja. S pomočjo metod tehnične analize trgovanja definiramo problem avtomatskega trgovanja v kontekstu strojnega učenja. Pridobljeno znanje uporabimo in zasnujemo lasten ATS. Slednjega implementiramo v obliki simulacije, pri čemer se skušamo približati realnim razmeram na trgu. Na koncu ovrednotimo njegovo delovanje na realnih tržnih podatkih kriptovalut in predlagamo možne izboljšave ter razširitve sistema.

Poglavje 2

Avtomatski trgovalni sistemi

2.1 Uvod v avtomatske trgovalne sisteme

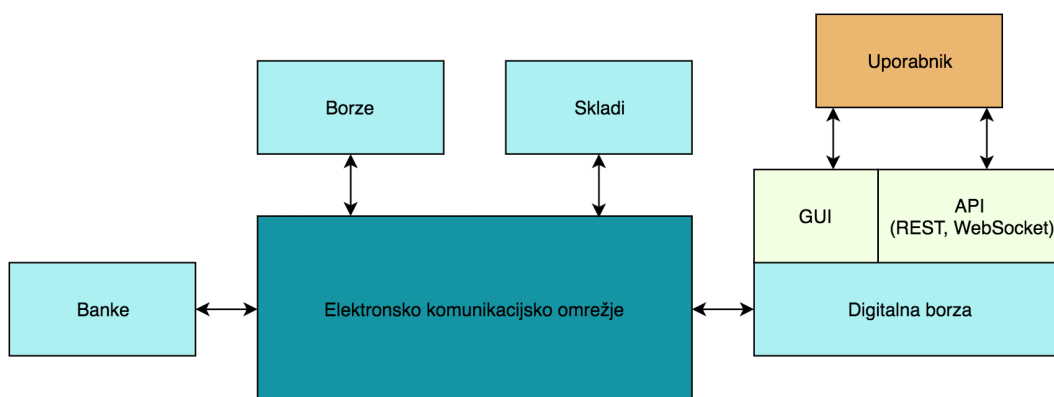
Avtomatski trgovalni sistem je računalniški sistem, ki trguje brez posredovanja človeka. Njegova prednost je izvajanje velikega števila kupčij pri velikih hitrostih, širok pregled trgov in objektivnost trgovalnih odločitev. Slabost avtomatskih trgovalnih sistemov je, da s kapitalom upravlja računalnik. Če je trgovalna programska oprema neustrezno zasnovana, lahko uporaba takega sistema prinese finančno izgubo. Avtomatski trgovalni sistemi ne bi bili mogoči brez podpore digitalnih borz, ki so opisane v nadaljevanju.

2.2 Digitalne borze

Digitalne borze za razliko od tradicionalnih borz, ki zahtevajo fizično prisotnost stranke za nakup neke delnice, zagotavljajo spletne vmesnike za trgovanje. Ti vmesniki so lahko grafični (GUI), večina digitalnih borz pa izpostavlja tudi REST oz. WebSocket spletne storitve. Prek njih lahko, če se avtentificiramo, trgujemo enako kot z GUI. Digitalne borze ne bi bile mogoče brez podpore omrežja, ki jih povezuje in s tem ustvarja virtualen trg.

2.2.1 Povezovanje digitalnih borz

V primeru delnic se v ozadju digitalne borze povežejo preko digitalnih trgovalnih omrežij. Ta omrežja so lahko privatna ali pa javna t.i. elektronska komunikacijska tržna omrežja (Electronic Communication Network - ECN). Privatna omrežja so vzpostavljena za namen pretoka velikih transakcij brez udeležbe velikih javnih borz (več o njih v [4]). Na drugi strani so elektronska komunikacijska omrežja velike borzne mreže, ki povezujejo različne finančne ustanove. Primarne dobrine ki se trgujejo preko njih so valute in vrednostni papirji. Najbolj znani med njimi so NASDAQ, NYSE (New York Stock Exchange) in Globex. Slika 2.1 prikazuje delovanje digitalnega trgovalnega omrežja.



Slika 2.1: Shema delovanja digitalnega trgovalnega omrežja.

2.2.2 Digitalne borze kriptovalut

Digitalne borze kriptovalut delujejo na podoben način kot digitalne delniške borze, le da se jim ni treba povezovati preko ECN, saj transakcije pri kriptovalutah potekajo po peer-to-peer principu preko spleta. Kriptovaluta se pri kupčiji z uporabo kriptografije prenakaže iz prodajalčevega računa na kupčev račun. Vse večje digitalne borze kriptovalut izpostavljajo REST in nekatere celo WebSocket storitve, zaradi česar so primerne za avtomatizirano trgovanje.

Bitstamp je v Sloveniji ustanovljena digitalna borza za kriptovaluto Bitcoin. Njen API ¹ izpostavlja množico prosto dostopnih REST in WebSocket storitev ter tudi nekatere zaščitene storitve, ki zahtevajo avtentikacijo z uporabniškim računom.

¹<https://si.bitstamp.net/api/>

2.3 Zgradba ATS

V grobem lahko ATS delimo na naslednje tri enote [14]: podatkovno enoto, odločitveno enoto in izvršilno enoto. Podatkovna enota skrbi za dotok podatkov v sistem. Odločitvena enota te podatke analizira, jih vrednoti in podaja trgovalne odločitve. Izvršilna enota skrbi za izvršitev teh odločitev.

2.3.1 Podatkovna enota

Iz spletnega tržišča lahko podatkovna enota preko spletnih storitev pridobi dva temeljna sklopa podatkov: knjigo naročil in pretekle transakcije za neko časovno obdobje. Če za primer vzamemo Bitstamp API, omenjen v prejšnjem razdelku, nam ta ponuja vrsto poizvedovalnih spletnih storitev, ki pa vse temeljijo na izračunih in povzetkih teh dveh temeljnih sklopov. Vse, kar se iz digitalnega tržišča da izvedeti, lahko izvemo iz knjige naročil in preteklih transakcij. Sledita primera HTTP zahtevkov in odgovorov z uporabo Bitstampovega programskega vmesnika za pridobitev knjige naročil (odsek 2.1 in preteklih transakcij (odsek 2.2).

Zahtevek: GET https://www.bitstamp.net/api/order_book/

Odgovor s knjigo naročil v formatu JSON:

```
{
  "timestamp": "1437897259",
  "bids": [
    ["288.99", "12.98844800"],
    ["288.67", "4.00000000"],
    ["288.65", "4.10030000"],
    ["288.64", "0.48069728"],
    ...
  ]
}
```

Odsek 2.1: GET zahtevek za pridobitev knjige naročil z uporabo Bitstamp API.

Knjiga naročil vsebuje aktivne ponudbe za prodajo in nakup valute. Vrednost polja "timestamp" je časovna značka pridobitve knjige naročil. Polje "bids" pa vsebuje seznam parov cena - količina. S tem nam povzame vse na Bitstamp oddane ponudbe za nakup in prodajo Bitcoina, ki so na voljo v tem trenutku. Knjiga naročil nam lahko dodatno pomaga razumeti dogajanje na trgu. Daje nam informacijo o razponu nakupne in prodajne cene, poleg tega pa nam količina ob posameznih cenah daje informacijo o razmerju nakupnih in prodajnih naročil. Sledi primer pridobitve zgodovine transakcij za določeno časovno obdobje.

Zahtevek: GET <https://www.bitstamp.net/api/transactions?time=minute>

Odgovor v obliki JSON:

```
[
  {
    date: "1437897414",
    tid: 9005545,
    price: "288.99",
    type: 1,
    amount: "3.28344797"
  },
  {
    date: "1437897253",
    tid: 9005544,
    price: "288.99",
    type: 1,
    amount: "20.22500000",
  }
]
```

Odsek 2.2: GET zahtevek preteklih transakcij z Bitstampa.

S poizvedbenim parametrom "time" v spletnem naslovu specificiramo časovno okno vrnjenih transakcij. Če ta parameter nastavimo na "minute" nam zahtevek vrne vse transakcije v zadnji minuti. V odgovoru se nahaja seznam transakcij s časovno značko, tipom (nakup - 0, prodaja - 1), ceno (v ameriških dolarjih) in količino izmenjanih Bitcoinov. Ti podatki so osnova za izračune pri tehnični analizi trgovanja (več v poglavju 3).

Podatki iz zunanjih virov

Nekateri avtomatski trgovalni sistemi poleg podatkov, ki jih ponujajo digitalne borze uporabljajo tudi podatke iz zunanjih virov. To so predvsem podatki iz drugih digitalnih borz, finančne novice in objave na socialnih omrežjih.

Cene istih dobrin so si na različnih digitalnih borzah zelo podobne, saj jih uravnavajo arbitražni trgovalci. Kljub temu pa se navadno velike nenadne spremembe v ceni dobrine ne odražajo na vseh digitalnih borzah enako hitro. Take spremembe so, če jih zaznamo dovolj hitro, lahko ključen faktor pri profitabilnosti avtomatskega trgovalnega sistema.

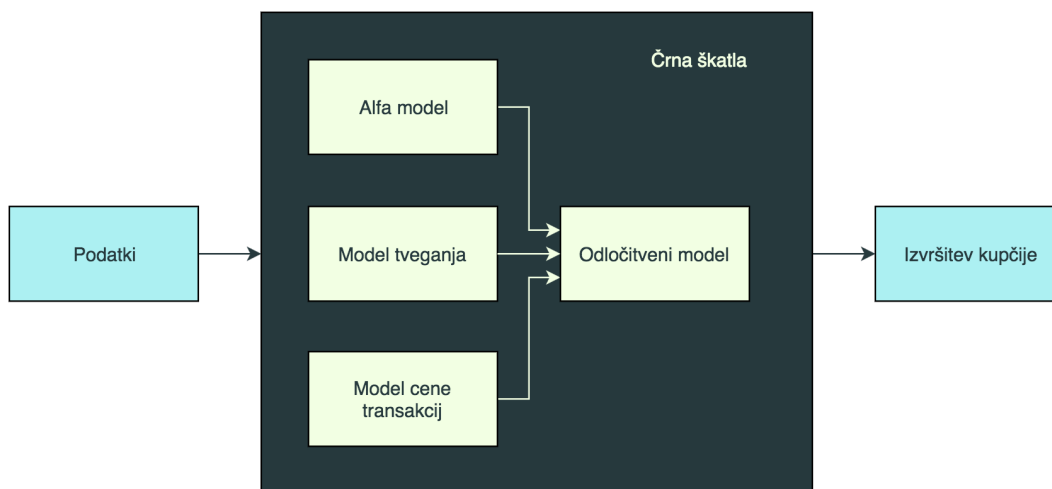
Poleg podatkov iz drugih digitalnih borz, pa pomembno vlogo pri trgovalnih odločitvah igrajo finančne novice [3]. Cene dobrin so zelo odvisne od globalne finančne situacije, političnih razmer, objav podjetij o letnem zaslužku itd. S hitrim dostopom do svežih novic si lahko pridobimo ključno prednost pred drugimi trgovalci. Obstajajo tudi podjetja, ki novice v realnem času filtrirajo glede na pomembnost za finančne trge in njihove povzetke izpostavljajo skozi spletne storitve. To nam omogoča njihovo preprosto vključitev v podatkovno enoto, brez potrebe po razčlenjevalnikih HTML dokumentov. Najbolj znano podjetje na tem področju je Bloomberg, ki izpostavlja svoj REST API ². Objave iz socialnih omrežij vključimo v podatkovno enoto na podoben način. Zajemamo jih iz uporabniških vmesnikov, ki jih socialna omrežja ponujajo. Na tem področju je najbolj popularen Twitter API ³.

²<http://www.bloomberglabs.com/api/>

³<https://dev.twitter.com/rest/public>

2.3.2 Odločitvena enota

Odločitvena enota je osrednja enota avtomatskega trgovalnega sistema. Vsebuje logiko, ki ovrednoti podatke, ki jih je pridobila podatkovna enota in proizvede signal za nakup oz. prodajo dobrine. Zaradi same kompleksnosti odločitvene enote se je zanjo uveljavil izraz "črna škatla" [14]. Slika 2.2 prikazuje njeno zgradbo.



Slika 2.2: Shema delovanja odločitvene enote.

V nadaljevanju sledi opis treh podenot odločitvenega modela.

Alfa model

Alfa model določa strategijo trgovanja. Pri tem se lahko opira na metode tehnične analize trgovanja, temeljne analize trgovanja, strojnega učenja, statističnih metod itd. Gre za algoritem, ki za vhodne podatke proizvede trgovalni signal, ki ga pred izvršitvijo prestrežeta model tveganja in model cene transakcij.

Model tveganja

Model tveganja omejuje finančno tveganje posamezne kupčije. Pri tem, glede na moč trgovalnega signala, dinamično spreminja količino premoženja, s katero se izvede posamezna kupčija. Pogosto vključuje tudi mehanizem za ustavitve izgube. To pomeni, da zaključi kupčije, ki se ne obnesejo po napovedih alfa modela in s tem omejuje izgubljena sredstva.

Model cene transakcij

Model cene transakcij se ukvarja s pristojbinami in vplivom naše transakcije na trg. Signal za trgovanje ni vedno enako močan, kar pomeni, da je alfa model v nekaterih situacijah bolj prepričan v svojo napoved kot pri drugih. Pri trgovanju je pomembno, da izvedemo le kupčije, ki so profitabilne z upoštevanjem pristojbin, ki jih za te kupčije plačamo. Model cene transakcij odloča, kateri signali so prešibki za njihovo izvršitev.

Večja kot je količina pretečene dobrine v transakciji, večji vpliv ima slednja na trg. Če upravljamo z veliko kapitala si ne želimo, da bi naš nakup velike količine dobrine destabiliziral trg in s tem preveč spremenil ceno po kateri smo dobrino sklenili kupiti. Model cene transakcij zato razdeljuje večje kupčije na več manjših zaporednih kupčij z namenom, da bi se cena dobrine na trgu čim manj spremenila.

2.3.3 Izvršilna enota

Prispevki treh podenot odločitvenega modela v prejšnjem razdelku prispevajo podatke o kupcijah, ki jih izvršuje izvršilna enota. Naloga izvršilne enote je oddajanje naročil preko programskih vmesnikov digitalne borze. Pri tem je potrebno posebno pozornost nameniti varnosti, saj so klici, ki jih izvršilna enota izvaja, občutljivi (lastnike avtomatskih trgovalnih sistemov namreč finančno izpostavlja).

Varno pošiljanje zahtevkov z naročili v primeru Bistampa

Spletne storitve, ki jih uporablja izvršilna enota so zaščitene, saj so vezane na uporabniški račun. V primeru digitalne borze Bitstamp njihov API zahteva, da v HTTP zahtevke z naročili vključimo sledeče podatke.

- **API ključ:** ključ, ki ga lahko vsak uporabnik generira za svoj račun (hkrati se generira tudi pripadajoč tajni ključ),
- **števec zahtevkov:** vsak naslednji zahtevek mora imeti števec zahtevkov večji od prejšnjega,
- **podpis:** z uporabo algoritma SHA-256 zgoščena vrednost sporočila (pri zgoščevanju se uporabi tajni ključ).

Ti parametri v zahtevku zagotavljajo temeljne vidike varne komunikacije: avtentikacijo, neponovljivost in zaščiteno pred spremembami. Avtentikacijo omogočajo API ključ, tajni ključ in zgoščeno sporočilo. Ker skrivni ključ poznata le imetnik računa in Bitstamp in ta ni viden v HTTP zahtevku, lahko le imetnik računa in Bitstamp izračunata veljavno zgoščeno vrednost sporočila (za pripet API ključ). Zgoščena vrednost sporočila zagotavlja, da je nemogoče spremeniti zahtevek tako, da bi bil še vedno veljaven. Bitstamp nespremenjenost preverja tako, da z uporabnikovim tajnim ključem izračuna zgoščeno vrednost sporočila in ga primerja s priloženo vrednostjo. Če se vrednosti ne ujemata, je bil originalni zahtevek spremenjen in se ta zavrne. Neponovljivost zahtevkov zagotavlja mehanizem s števcem.

Tipi naročil

Izvršilna enota je zadolžena za izvajanje naročil, ki ji jih proizvede odločitvena enota. Pri tem ima na voljo sledeče tipe naročil.

- **takojšni nakup:** izvedemo takojšni nakup po najnižji prodajni ceni,
- **takojšna prodaja:** izvedemo takojšno prodajo po najvišji nakupni ceni,
- **omejeni nakup:** v knjigo naročil vstavimo ponudbo za nakup po najvišji ceni, katero smo pripravljeni plačati za dobrino,
- **omejena prodaja:** v knjigo naročil vstavimo ponudbo za prodajo po najnižji ceni, po kateri smo pripravljeni prodati dobrino,
- **zaustavitveni nakup:** če se doseže specificirana cena (višja od trenutne), izvedemo takojšni nakup,
- **zaustavitvena prodaja:** če se doseže specificirana cena (nižja od trenutne) se izvede takojšna prodaja.

Bitstamp API podpira le dva izmed zgoraj navedenih: omejeni nakup in omejeno prodajo. Na njihovem grafičnem vmesniku pa so na voljo tudi ostali štirje.

Poglavje 3

Tehnična analiza trgovanja

Tehnična analiza trgovanja je raziskovanje trga samega. Gre za napovedovanje cen v prihodnosti izključno na podlagi preteklih podatkov [6]. V nasprotju s temeljno analizo trga, tehnična analiza ne podaja točne ocene o vrednosti dobrine, pač pa glede na vzorce, ki se pojavljajo v njenem grafu gibanja cen napoveduje, kaj se bo najverjetneje zgodilo. Tehnični analitiki verjamejo v sledeče izkustvene predpostavke.

- cena delnice oz. valute odraža vse, kar se z njo dogaja, saj nanjo vplivajo vsi udeleženci na trgu,
- cene se gibljejo trendovsko in ne naključno,
- pretekli vzorci se ponavljajo (trgovalci se na spreminjanje trga obnašajo podobno, kot so se v preteklosti).

V nadaljevanju so opisani nekateri osnovni pojmi in metode tehnične analize trgovanja.

3.1 Osnovni pojmi tehnične analize

3.1.1 Časovne vrste

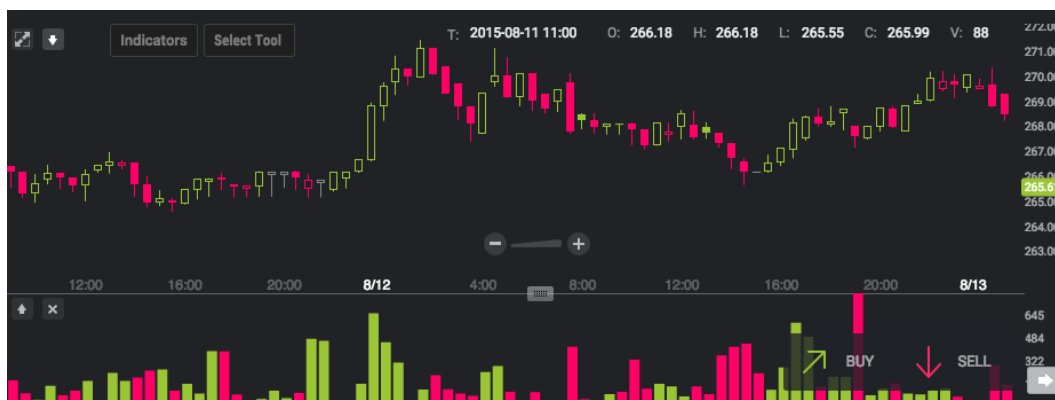
Tehnična analiza trgovanja temelji na teoriji časovnih vrst. Časovne vrste so zaporedja točkastih podatkov, ki predstavljajo zaporedne meritve nekega pojava. Ponavadi želimo z analiziranjem časovnih vrst iz sekvenčnih podatkov izvleči informacije, ki bi nam lahko pomagale razumeti neke merjene pojave, ali pa napovedati njihovo obnašanje v prihodnosti. Časovne vrste ponavadi prikazujemo s črtnim grafom.

3.1.2 Časovne rezine

Tehnična analiza trgovanja povzema pretekle transakcije med trgovalci v t.i. časovne rezine (angl. timeframes), kar pomeni, da ne obravnava vsake transakcije posebej, pač pa jih združuje glede na pripadnost časovnim intervalom. Transakcije, ki so se zgodile znotraj posamezne časovne rezine, prispevajo svoje podatke v temeljne lastnosti časovne rezine. To so začetna in končna cena rezine, najvišja in najnižja cena rezine ter količina pretečene valute (t.i. volumen). Zaporedje teh podatkov tvori časovne vrste, ki jih tehnična analiza trgovanja analizira.

Dolžina intervala, ki določa časovno rezino, je poljubna. Med najpogostejšimi izbirami pa so naslednji intervali.

- minutni: 1, 5, 10, 15, 30,
- urni: 1, 4, 12,
- dnevni: 1, 7.



Slika 3.1: Primer 30-minutnih japonskih svečnikov iz grafičnega uporabniškega vmesnika spletne tržnice Bitstamp.

Posamezno časovno vrsto nekega podatka časovnih rezin lahko upodobimo s črtnim grafom, omenjenem v prejšnjem razdelku. Iz uporabniškega stališča pa je bolj priročno, da bi se na grafovski predstavitvi časovnih rezin za posamezno časovno rezino prikazalo čim več časovnih vrst njenih temeljnih podatkov. V ta namen se uporablja veliko različnih grafičnih predstavitev zaporedja časovnih rezin. Med njimi najbolj uporabljena je upodobitev s t.i. japonskimi svečniki [5]. Element grafa japonskih svečnikov, ki predstavlja eno časovno rezino, je stolpec, ki ima na vrhu in na dnu paličico. Končiča paličic predstavljata najvišjo in najnižjo ceno rezine. Vrh in dno samega stolpca pa začetno oz. končno ceno rezine. Če je stolpec rdeče barve, je začetna cena višja od končne, kar pomeni, da cena dobrine pada. Nasprotno velja za zelen stolpec. Japonski svečnik prikaže štiri od petih temeljnih podatkov. Manjkajoči podatek volumen, pa lahko prikažemo z dopolnilnim stolpičnim grafom, ki leži pod japonskimi svečniki, kot prikazuje slika 3.1.

3.2 Trgovalni indikatorji

Trgovalni indikatorji so meritve podatkov na trgu z namenom pridobitve močnejše informacije o dogajanju na njem. Meritev enega indikatorja pripada eni časovni rezini, vendar pri tem upošteva podatke prejšnjih časovnih rezin. Zaporedne meritve indikatorjev tvorijo časovne vrste indikatorjev. Uporabljajo se za meritve pretoka kapitala, obstoj trendov, volatilitnost (verjetnost nenadnih večjih sprememb cene) itd. Tehnični trgovalci jih zato velikokrat uporabljajo kot trgovalne signale pri sprejemanju odločitev.

Konstrukcija indikatorjev se glede na njihovo zalogo vrednosti deli na omejene indikatorje (oscilatorje) in neomejene indikatorje. Ponavadi je oscilator omejen na vrednosti od 0 do 100. Sledi pregled nekaterih osnovnih trgovalnih indikatorjev in njihove namenskosti.

3.2.1 Preprosto drseče povprečje

Preprosto drseče povprečje (angl. simple moving average) časovne vrste se izračuna kot povprečje k njenih predhodnih podatkov. V primeru drsečega povprečja cene, ga izračunamo kot povprečje cen k prejšnjih časovnih rezin (vključno z njo samo). Spada med neomejene indikatorje.

Za časovno rezino i , pri danem k , se preprosto drseče povprečje izračuna kot

$$SMA(i, k) = \frac{\sum_{j=i-k}^i c_j}{k}, \quad (3.1)$$

kjer je c_j končna cena j -te časovne rezine.

Na analogen način bi lahko izračunali tudi preprosto drseče povprečje volumna, ki se v tehnični analizi trgovanja pogosto uporablja. S pomočjo preprostega drsečega povprečja cene lahko proizvajamo trgovalne signale tako, da za vsako časovno rezino računamo dve preprosti drseči povprečji

za različen k . Preprosto drseče povprečje z manjšim k je bolj občutljivo na spremembe na trgu. Ko se časovni vrsti, ki jih preprosti drseči povprečji tvorita, sekata, lahko to interpretiramo kot trgovalni signal.

3.2.2 Eksponentno drseče povprečje

Preprosto drseče povprečje je enakomerno uteženo. To pomeni, da v končnem izračunu vse upoštevajoče časovne rezine dajejo enak prispevek. Če želimo, da imajo novejša časovna rezina večji vpliv, ga utežimo. Eksponentno drseče povprečje (angl. exponential moving average) je eksponentno uteženo preprosto drseče povprečje.

Za časovno rezino i , pri danem, k se eksponentno drseče povprečje izračuna kot

$$EMA(i, k) = m_i c_i + (1 - m_i) EMA(i - 1, k), \quad (3.2)$$

kjer je c_i končna cena rezine in m_i utež, ki se pri danem k izračuna kot

$$m_i = \frac{2}{k + 1}. \quad (3.3)$$

Njegovo delovanje z namenom produciranja trgovalnih signalov je enako delovanju preprostega drsečega povprečja cen.

3.2.3 MACD

Moving average convergence divergence (MACD) oz. konvergenca in divergenca drsečih povprečij je indikator sestavljen iz sledečih časovnih vrst.

- **MACD indikator:** razlika dveh eksponentnih drsečih povprečij končne cene časovne rezine (prvo z manjšim k , drugo z večjim k),
- **signalni podatek:** eksponentno drseče povprečje MACD indikatorja,
- **divergenca:** razlika MACD indikatorja in signalnega podatka.

Trgovalci signale z MACD trgovalnim indikatorjem razbirajo na mnogo načinov. Najbolj preprosti načini so: sekanje časovnih vrst signalnega podatka in MACD indikatorja, sekanje signalnega podatka z abscisno osjo (sprememba predznaka signalnega podatka) in gibanje divergence za predvidevanje obratov trendov. Za divergenco namreč velja, da ima zelo zglajene vrednosti, kar pomeni, da ni občutljiva na nenadna nihanja.

3.2.4 Stohastični oscilator

Stohastični oscilator spada med omejene indikatorje. Namenjen je merjenju sprememb v trendu. Sestavljen je iz dveh komponent, komponente K in njenega preprostega drsečega povprečja D . Komponento K za časovno rezino i in periodo k izračunamo tako, da najprej izračunamo maksimum in minimum cen za preteklih k rezin.

$$\begin{aligned} h_{max}(i, k) &= \max_{i-k \leq j \leq i} (h_j) \\ l_{min}(i, k) &= \min_{i-k \leq j \leq i} (l_j) \end{aligned} \quad (3.4)$$

Pri tem sta h_j in l_j najvišja in najnižja cena v rezini j . Dobljeni vrednosti nato vstavimo v sledečo enačbo.

$$K(i, k) = 100 \cdot \frac{(c_i - l_{min}(i, k))}{h_{max}(i, k) - l_{min}(i, k)}, \quad (3.5)$$

kjer je c_i končna cena rezine i .

Zaloga vrednosti K in D komponente je na intervalu od 0 do 100. S stohastičnim oscilatorjem trgujemo glede na sekanja časovnih vrst njegovih komponent, ampak le, če se sekanje zgodi nad ali pod določenim pragom (navadno sta ta praga 80 in 20). Šele tedaj privzamemo, da je trgovalni signal dovolj močan.

3.2.5 Indeks relativne moči

Indeks relativne moči (angl. relative strength index - RSI) ima podobno vlogo kot stohastični oscilator. Prav tako ima zalogo vrednosti na intervalu med 0 in 100. Sestavljen je iz ene same komponente, ki se za rezino i pri danem k izračuna tako, da najprej izračunamo povprečje pozitivnih in povprečje negativnih razlik končnih cen zadnjih k rezin (vključno z i -to rezino).

$$\begin{aligned} avg_+(i, k) &= \frac{(k-1) \cdot avg_+(i-1, k) + d_+(i)}{k} \\ avg_-(i, k) &= \frac{(k-1) \cdot avg_-(i-1, k) + d_-(i)}{k} \end{aligned} \quad in \quad (3.6)$$

Pri tem se $d_+(i)$ in $d_-(i)$ izračunata na sledeč način.

$$\begin{aligned} d_+(i) &= \begin{cases} c_i - c_{i-1} & ; c_i - c_{i-1} > 0 \\ 0 & ; \text{sicer} \end{cases} \\ d_-(i) &= \begin{cases} c_i - c_{i-1} & ; c_i - c_{i-1} < 0 \\ 0 & ; \text{sicer} \end{cases} \end{aligned} \quad in \quad (3.7)$$

Indeks relativne moči se izračuna kot

$$RSI(i, k) = 100 - \frac{100}{1 + \frac{avg_+(i, k)}{avg_-(i, k)}}. \quad (3.8)$$

Indikator ocenjuje, kdaj se bo trend najverjetneje obrnil navzgor in kdaj navzdol (praga sta navadno 30 in 70). Ko je vrednost indikatorja nad zgornjim pragom, to interpretiramo kot povečano verjetnost padca cene. Obratno velja za spodnji prag.

Poglavje 4

Strojno učenje in trgovanje

Odločanje ATS bi lahko implementirali zgolj s trgovalnimi signali, ki jih proizvaja en sam indikator tehnične analize trgovanja (npr. eden izmed opisanih v razdelku 3.2). Kljub temu pa je pri zanašanju na njegove signale potrebna previdnost. Trg je namreč dinamičen in s tem bi se spreminjala tudi uporabnost izbranega indikatorja [7]. Če pa bi pri odločitvenem modelu ATS uporabili več indikatorjev, bi nam ti zagotovili več informacije o trgu, kot če bi uporabili le enega (merijo namreč različne lastnosti gibanja cene). S pametnim razbiranjem trgovalnih signalov iz množice trgovalnih indikatorjev se lahko prilagajamo na različne tržne pogoje. V ta namen lahko uporabimo tehnike strojnega učenja, katerih pregled sledi v nadaljevanju poglavja.

4.1 Osnovno o strojnem učenju

Strojno učenje je področje umetne inteligence, ki se ukvarja z računalniškim odkrivanjem znanja iz podatkov. Pojem strojnega učenja se pogosto uporablja v kombinaciji s pojmom podatkovnega rudarjenja. Pri slednjem je poudarek bolj na odkrivanju znanj iz velikih zbirk podatkov. Deli se na tri glavne paradigme: nadzorovano učenje, nenadzorovano učenje in spodbujevalno učenje [10]. V nadaljevanju sledi opis teh treh paradigem, poudarek pa bo predvsem na nadzorovanem učenju, ki je z vidka ATS najbolj zanimivo.

4.2 Paradigme strojnega učenja

4.2.1 Nadzorovano učenje

Pri nadzorovanem učenju so nam dani pari (x, y) , $x \in X$ (vhodna spremenljivka, ponavadi vektor) in $y \in Y$ (izhodna spremenljivka, ponavadi število, lahko tudi vektor). Tem parom ponavadi rečemo primerki (angl. instance). Naloga nadzorovanega učenja je poiskati funkcijo $f : X \rightarrow Y$, ki se čim bolj prilega podatkom (več o prileganju v nadaljevanju). Fazi iskanja take funkcije pravimo faza učenja in se je ponavadi lotimo z enim izmed znanih algoritmov strojnega učenja. V glavnem so algoritmi strojnega učenja matematični modeli, ki so sposobni pod določenimi pogoji poiskati optimalno funkcijo f .

Klasifikacija in regresija

Najbolj pogosti nalogi nadzorovanega učenja sta klasifikacija in regresija. Pri klasifikaciji ima izhodna spremenljivka diskretno zalogo vrednosti. To je lahko bodisi podmnožica celih števil ali pa vektor s komponentami iz celih števil. Intuitivno problem klasifikacije rešuje vprašanje pripadnosti primerkov nekim razredom. Pri regresiji ima izhodna spremenljivka zvezno zalogo vrednosti. To pomeni, da je zaloga vrednosti izhodne spremenljivke bodisi množica realnih števil, ali pa množica vektorjev z realnimi komponentami. Intuitivno regresija napoveduje številsko lastnost primerka.

Validacija nadzorovanega strojnega učenja

V praksi množico primerkov, ki nam je dana v primeru problema nadzorovanega strojnega učenja, razdelimo na tri disjunktne podmnožice: učno množico, testno množico in validacijsko množico. To storimo z namenom,

da bodo naši dobljeni rezultati čim bolj verodostojni. Učno množico uporabljamo zgolj za učenje algoritma strojnega učenja. Testna množica nam v procesu učenja služi za preverjanje uspešnosti našega algoritma. S testiranjem algoritma na testni množici pridobivamo koristne informacije, kako ga v nadaljevanju izboljšati. Z modifikacijami med testiranjem lahko model preveč prilagodimo na testne podatke, kar bi posledično prineslo slabe rezultate v realnosti. Tako situacijo preprečuje testiranje na validacijski množici, ki ga izvedemo na koncu, ko smo zaključili s prilagajanjem modela. Pogosta izbira za velikost učne množice podatkov je 60%, testne množice 20% in validacijske množice 20% vseh primerkov.

Merjenje uspešnosti algoritma pri klasifikaciji

Pri klasifikaciji z binarno izhodno spremenljivko (ali primerek pripada nekemu razredu ali ne) si najpogosteje pomagamo z matriko napake (angl. error matrix). Pri tem delimo napovedi v štiri kategorije: pravilno pozitivne (PP), napačno pozitivne (NP), pravilno negativne (PN) in napačno negativne (NN). Če primerek pravilno klasificiramo kot pripadnika nekega razreda, je napoved pravilno pozitivna. Če za primerek pravilno napovemo, da ne pripada nekemu razredu, je napoved pravilno negativna. Razmislek je analogen v primeru napačnih napovedi. Uspešnost nadzorovanega učenja lahko tako ponazorimo z matriko napake

$$A = \begin{bmatrix} PP & NN \\ NP & PN \end{bmatrix},$$

kjer so PP , NP , NN in PN števila primerkov, ki pripadajo prej omenjenim kategorijam. Uspešnost algoritma pri klasifikacijskem problemu lahko definiramo s pomočjo natančnosti in priklica.

$$\text{natancnost} = \frac{PP}{PP + NP}$$

$$\text{priklic} = \frac{PP}{PP + NN}$$

Natančnost nam pove delež primerkov, ki smo jih pravilno klasificirali kot predstavnike razreda med vsemi, ki smo jih klasificirali kot predstavnike razreda. Na drugi strani nam priklic pove, koliko izmed dejanskih predstavnikov razreda smo pravilno klasificirali. Najbolj uspešen je algoritem, ki maksimizira obe vrednosti. Želeli bi si, da nam uspešnost meri le ena količina namesto dveh, pri čemer nam pomaga mera F , ki je definirana kot

$$F = \frac{2 * \textit{natancnost} * \textit{priklic}}{\textit{natancnost} + \textit{priklic}}.$$

Uspešen algoritem ima visoko natančnost in visok priklic, zato ima posledično tudi visoko mero F . Tako kot natančnost in priklic pa tudi mera F zavzame vrednosti iz intervala $[0, 1]$.

V primeru klasifikacije primerkov v več razredov, pa se zgoraj opisane mere za uspešnost posplošijo. Namesto kvadratne matrike napake velikosti 2 imamo kvadratno matriko napake velikosti n , kjer je n število možnih razredov.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix},$$

kjer je a_{ij} število primerkov, ki so razreda i , vendar so bili klasificirani kot razred j . Natančnost in priklic sta tako definirana za vsak razred posebej

$$\textit{natancnost}_i = \frac{a_{ii}}{\sum_j a_{ji}}$$

$$\textit{priklic}_i = \frac{a_{ii}}{\sum_j a_{ij}}.$$

Spremeni se tudi mera F , ki je v tem primeru definirana za vsak razred posebej

$$F_i = \frac{2 * \textit{natancnost}_i * \textit{priklic}_i}{\textit{natancnost}_i + \textit{priklic}_i}.$$

Tudi pri klasifikaciji v več razredov pa si želimo imeti enoštevilsko mero (ne le vektor), ki bi odražala uspešnost algoritma. Mero F lahko preoblikujemo v uteženo mero F , tako da njene komponente množimo z deleži razredov, ki tem komponentam pripadajo in dobljena števila seštejemo. Spet dobimo število na intervalu $[0, 1]$, ki nam pove klasifikacijsko uspešnost našega algoritma.

Merjenje uspešnosti algoritma pri regresiji

Pri regresiji zaradi zveznosti problema prileganje merimo z napakami. Gre za funkcije, ki za podane pare $(f(x_i), y_i)$, kjer je $f(x_i)$ napovedana vrednost in y_i dejanska vrednost izhodne spremenljivke i -tega primerka, merijo njihovo različnost. Manjše kot so te napake, boljša je naša iskana funkcija. Popularni funkciji napak pri regresijskih problemih sta srednja kvadratna napaka in srednja absolutna napaka. Za n parov (x_i, y_i) se srednja kvadratna napaka izračuna kot

$$MSE = \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2, \quad (4.1)$$

srednja absolutna napaka pa kot

$$MAE = \frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|. \quad (4.2)$$

4.2.2 Nenadzorovano učenje

Pri nenadzorovanem učenju nimamo na voljo izhodne spremenljivke, ampak le vhodno spremenljivko. V podatkih želimo najti neko strukturo oz. v njih odkriti zakonitosti. Naloga učnega algoritma je za učne primerke določiti njihove razrede tako, da so si primerki znotraj učnih razredov čim bolj podobni. Podobnost med primerki merimo z neko metriko, ki je ponavadi dana vnaprej. Pri tem je število razredov lahko izbrano vnaprej, ali pa ga mora določiti sam učni algoritem. Algoritmi nenadzorovanega učenja navadno uporabljajo naslednje pristope.

- **od spodaj navzgor:** na začetku je vsak primerek svoj razred, tekom izvajanja algoritma pa se ti združujejo glede na podobnost,
- **od zgoraj navzdol:** na začetku so vsi primerki znotraj enega razreda, ki se tekom algoritma deli na podrazrede,
- **n -razredov:** na začetku se izbere n primerkov (predstavnikov razredov), katerim se glede na podobnost priključijo ostali (ta postopek se ponavlja).

Pogoste naloge nenadzorovanega učenja so gručenje, ugotavljanje verjetnostne porazdelitve, filtriranje itd.

4.2.3 Spodbujevalno učenje

Spodbujevalno učenje temelji na avtonomnosti računalnika. Slednji komunicira direktno z dinamičnim okoljem z namenom, da bi dosegel nek cilj. Podatki mu torej niso znani, ampak jih pridobiva z interakcijo z okoljem (izvajanjem akcij). Za doseg cilja je ponavadi potrebno zaporedje večih akcij. Računalnik pri tem teži k maksimiziranju nagrade (ki jo prejme za uspešno dosežen cilj) in minimiziranju kazni (ki jo prejme za neuspeh).

4.3 Trgovanje z vidika strojnega učenja

Trgovanje je z vidika strojnega učenja problem, ki ga najlažje opišemo kot problem nadzorovanega učenja. Predstavljamo si ga lahko kot klasifikacijo ali regresijo časovnih rezin. Vhodna množica primerkov so torej pretekle časovne rezine in njihove temeljne lastnosti (predstavljene v poglavju 3.1.2). Za vsako preteklo časovno rezino je mogoče izvedeti, kakšna je bila sprememba cene dobrine v njeni prihodnosti. Slednjo lahko merimo glede na neko samostojno časovno rezino v prihodnosti, ali pa glede na interval časovnih rezin v prihodnosti. Sprememba cene v prihodnosti časovne rezine nam lahko služi kot izhodna spremenljivka primerka. Za njeno zalogo vrednosti imamo na voljo tri diskretne razrede (cena zraste, cena pade, cena se bistveno ne spremeni) ali pa zvezno količino (spremembo cene dobrine po časovni rezini). Naloga algoritma strojnega učenja je na podlagi znanih lastnosti časovne rezine napovedati gibanje cene v prihodnosti. Ker učna množica ni fiksna, pač pa se skozi čas osvežuje s podatki o novejših časovnih rezinah, sega problem na področje rudarjenja podatkovnih tokov. Algoritem strojnega učenja ima vlogo alfa modela odločitvene enote v ATS, saj določa strategijo odločanja.

Na področju trgovanja s podporo strojnega učenja so bile v preteklosti uporabljene mnoge tehnike. Manj uspešne so bile tiste, ki so predpostavljale obstoj stroge strukture v podatkih (npr. linearnosti). Linearnost podatkov pomeni, da je izhodna spremenljivka v linearnem razmerju z vhodno spremenljivko. V nadaljevanju sledi pregled algoritma k -najbližjih sosedov, naivnega Bayesovega klasifikatorja in umetnih nevronske mreže. Te tri metode ne predpostavljajo linearne strukture v podatkih in so bile v preteklosti uspešno aplicirane v trgovalne namene [1, 7].

4.4 Algoritem k -najbližjih sosedov

Algoritem k -najbližjih sosedov (angl. k -nearest neighbours - kNN) je ena najpreprostejših metod strojnega učenja, saj se faza učenja algoritma dogaja kar v času predikcije. Za dan primerek iz množice primerkov za predikcijo (tisti, katerih izhodno spremenljivko moramo določiti) iz množice primerkov za učenje (tisti, katerih izhodne spremenljivke so znane) izberemo k najbližjih primerkov. Parameter k je pri tem vnaprej izbran parameter in je poljuben, ponavadi pridobljen s testiranjem algoritma. Dostikrat v primeru klasifikacije z enim izhodnim razredom vzamemo za parameter k liho število in se s tem izognemo izenačenim primerom. Performančno je algoritem potraten, saj moramo vsak primerek iz testne množice primerjati z vsakim primerkom v učni množici.

4.4.1 Funkcije sosednosti

Da izberemo k -najbližjih sosedov primerku, ki ga določamo, moramo definirati funkcijo sosednosti, ki kvantificira podobnost med dvema primerkoma. To pomeni, da za dva primerka na podlagi njunih vhodnih spremenljivk izračuna njuno razdaljo. Če so posamezne komponente vhodnih spremenljivk zvezne, je zelo pogosta izbira za funkcijo sosednosti evklidska razdalja. Za par vhodnih spremenljivk x_1 in x_2 dimenzije n se izračuna kot

$$E(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_1(i) - x_2(i))^2}, \quad (4.3)$$

kjer sta $x_1(i)$ in $x_2(i)$ i -ti komponenti pripadajočih vektorjev.

Če so posamezne komponente vhodnih spremenljivk diskretne (to pomeni, da je njihova zaloga vrednosti končna množica), pa se pogosto uporabljata Hammingova in Manhattanska razdalja. Hammingova razdalja se za par vektorjev vhodnih spremenljivk izračuna kot število mest, na katerih sta si ta različna. Manhattanska razdalja pa se izračuna podobno kot evklidska, le na diskreten način.

Pri vhodnih spremenljivkah x_1 in x_2 dimenzije n se izračuna kot

$$M(x_1, x_2) = \sum_{i=1}^n |x_1(i) - x_2(i)|. \quad (4.4)$$

4.4.2 Izračun izhodne spremenljivke

Ko izberemo množico k -najbližjih sosedov, nam ostane izračun izhodne spremenljivke primerka. Glede na to, ali gre za klasifikacijski ali regresijski problem ločimo naslednji dve možnosti.

- **klasifikacija:** izhodna spremenljivka je razred, zato je najbolj smiselno, da je izhodna spremenljivka vzroca, ki ga določamo tisti razred, ki prevladuje v množici k -najbližjih sosedov,
- **regresija:** izhodna spremenljivka je zvezna, zato je najbolj smiselno vzeti povprečje izhodnih spremenljivk primerkov iz množice k -najbližjih sosedov.

Včasih je pri izračunu izhodne spremenljivke smiselno bližjim primerkom dati večjo vlogo pri izračunu izhodne spremenljivke. Pri klasifikaciji to dosežemo tako, da posamezen primerk glede na razdaljo repliciramo. To pomeni, da ga v izračunu izhodne spremenljivke upoštevamo $\lfloor \frac{L}{d} \rfloor$ -krat, kjer je L poljuben parameter in d razdalja. Pri regresiji lahko razdaljo upoštevamo kot faktor pri posameznih členih v izračunu povprečja izhodne spremenljivke (člene v vsoti množimo z $\frac{1}{d}$).

4.5 Naivni Bayesov klasifikator

Naivni Bayesov klasifikator je preprost verjetnostni klasifikator, ki za attribute primerka predpostavlja, da so pogojno neodvisni od izhodne spremenljivke. Matematično to lastnost izrazimo kot

$$p(a_1, a_2, \dots, a_n | c_j) = \prod_i p(a_i | c_j),$$

kjer je $p(a_1, a_2, \dots, a_n | c_j)$ verjetnost, da ima primerek z razredom c_j vrednosti atributov a_1, a_2, \dots, a_n (analogno za desno stran enačbe). Pri zgornji predpostavki velja

$$p(c_j | a_1, a_2, \dots, a_n) = p(c_j) \cdot \prod_i \frac{p(c_j | a_i)}{p(a_i)}.$$

To pomeni, da če imamo primerek z danimi vrednostmi atributov a_1, a_2, \dots, a_n , lahko izračunamo verjetnost, da ta pripada posameznim razredom. Med razredi izberemo tistega, pri katerem ima primerek največjo verjetnost.

Problem nastane, ko je nek atribut primerka zvezno porazdeljen. Iz teorije verjetnosti vemo, da je verjetnost primerka, ki bo imel vrednost i -tega atributa ravno a_i enaka 0. Kar pomeni, da bi v zgornjem produktu bil člen $p(c_j | a_i)$ enak 0 za vsak i . V takih primerih si lahko pomagamo bodisi z diskretizacijo zvezne porazdelitve, ali pa s funkcijo gostote zvezne porazdelitve [16].

Kljub temu, da je predpostavka o neodvisnosti atributov zelo močna zahteva in največkrat ne drži, se v praksi naivni Bayesov klasifikator dobro obnaša, poleg tega pa je sam postopek klasifikacije hiter.

4.6 Umetne nevronske mreže

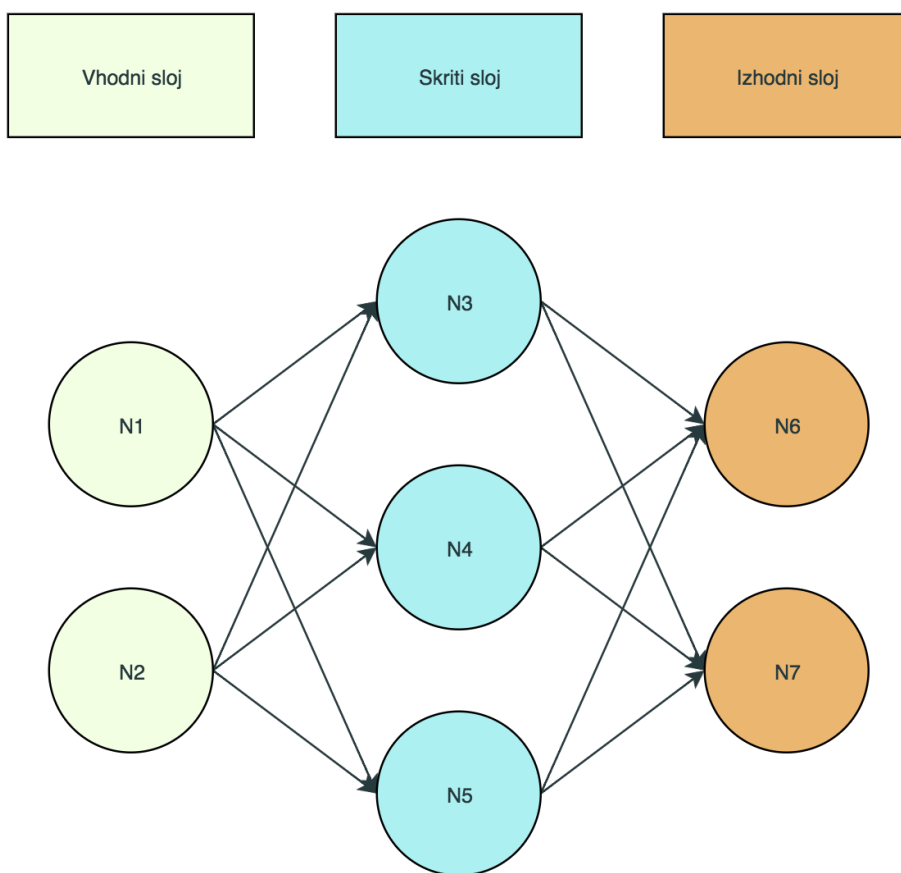
4.6.1 Osnovno o umetnih nevronskih mrežah

Umetna nevronska mreža (angl. Artificial Neural Network - ANN) je model za nadzorovano učenje (za klasifikacijo ali regresijo), ki spominja na nevrone v živčevju biološkega organizma. Gre za sistem povezanih nevronov, ki med seboj komunicirajo. Vsak nevron ima vhodno in izhodno množico uteženih povezav do ostalih nevronov. Nevron prejme različne signale od vhodnih nevronov (do njih vodi vhodna množica povezav), jih obdela in odda signal izhodni množici nevronov (do njih vodi izhodna množica povezav).

Umetna nevronske mreža uči tako, da s pomočjo različnih algoritmov prilagaja uteži na povezavah med nevroni in pri tem zvišuje klasifikacijsko točnost oz. zmanjšuje regresijsko napako. Vsak testni vzorec ponavadi večkrat uporabimo za učenje nevronske mreže (naredimo več iteracij učenja). Če v vzorcih obstaja struktura in smo za naš problem uporabili ustrezno nevronske mreže, potem uteži skozi proces učenja konvergirajo proti optimalnim vrednostim.

4.6.2 Upodabljanje umetnih nevronskih mrež

Umetne nevronske mreže grafično upodabljamo kot grafe povezav in vozlišč. Najbolj osnovne modeliramo z usmerjenimi acikličnimi grafi. Na mnogih področjih (npr. pri prepoznavanju pisave in procesiranju slik) pa se uporabljajo tudi usmerjene ciklične umetne nevronske mreže. Usmerjeni cikli v njihovi strukturi omogočajo pomnjenje prejšnjega stanja [13]. Njihova uporabnost pride do izraza pri sekvenčnih podatkih, ki so med seboj odvisni. V nadaljevanju se bomo osredotočil na usmerjene aciklične umetne nevronske mreže, ki so preprostejše in so bile v preteklosti uspešno aplicirane na področju trgovanja [7].



Slika 4.1: Primer tronivojske umetne nevronske mreže.

4.6.3 Strukture acikličnih umetnih nevronskih mrež

Strukture umetnih nevronskih mrež so zelo različne in služijo različnim problemom. Nevroni se delijo na nivoje. Nevroni iz posameznega nivoja med seboj niso povezani, povezani pa so z nevroni iz največ dveh nivojev. Umetne nevronske mreže se razlikujejo po številu nivojev, številu nevronov v nivojih in načinom medsebojne povezanosti. Slika 4.1 prikazuje tronivojsko polno povezano umetno nevronske mrežo s številom nevronom po nivojih 2, 3, 2. Število nevronov v vhodnem nivoju je enako dolžini vektorja vhodne spremenljivke. Ponavadi imamo v primeru klasifikacijskega problema izhodnih nevronov toliko kot izhodnih razredov (velikost množice zaloge vrednosti), pri čemer se naenkrat aktivira le eden. V primeru regresijskega problema pa je število izhodnih nevronov enako dimenziji vektorja izhodne spremenljivke.

Nevroni poleg povezav vsebujejo tudi aktivacijsko funkcijo. Ta na podlagi vhodnih signalov nevrona le-te transformira v izhodni signal. Naj bo N_i množica vhodnih nevronov i -tega nevrona. Moč signala, ki ga nevron prejme, se izračuna po naslednji enačbi.

$$p_i = \sum_{j \in N_i} w_{ij} \cdot p_j, \quad (4.5)$$

kjer je w_{ij} utež na povezavi med i -tim (trenutnom nevronom) in j -tim nevronom (nevron iz množice nevronov povezanih s trenutnim). Nevron i torej prejme signal z močjo p_i od svojih vhodnih nevronov, kakšen signal pa odda, je odvisno od njegove aktivacijske funkcije.

Če je aktivacijska funkcija linearna, torej oblike $f(x) = k \cdot x + n$, je nevron linearen. Poleg linearne funkcije pa se uporabljajo tudi marsikatero druge aktivacijske funkcije, navedene v nadaljevanju.

- Heavisidova funkcija koraka

$$f(x) = \begin{cases} 0 & ; x \leq 0 \\ 1 & ; x > 0 \end{cases}, \quad (4.6)$$

- hiperbolični tangens

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (4.7)$$

- sigmoidna funkcija:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (4.8)$$

Glavna lastnost teh funkcij je, da so skrčitvene. To pomeni, da katerokoli vhodno vrednost transformirajo v vrednost na intervalu med 0 in 1. Ponavadi imajo nevroni v istem nivoju enako aktivacijsko funkcijo, zato nivoje dostikrat imenujemo po aktivacijski funkciji.

Perceptron

Najbolj osnovna struktura umetne nevrnske mreže je perceptron, ki služi kot binarni klasifikator. To je klasifikator, ki ima le dve možni vrednosti izhodne spremenljivke. Ima dvonivojsko strukturo s poljubnim številom nevronov v vhodnem nivoju in enim nevronom v izhodnem nivoju. Če se izhodni nevron aktivira, vzorec pripada enemu izmed razredov, sicer pa drugemu. Aktivacijska funkcija izhodnega nevrna v perceptronu je podobna Heavisideovi funkciji in je navedena v sledeči enačbi.

$$f(x) = \begin{cases} 0 & ; x \leq A \\ 1 & ; x > A \end{cases}, \quad (4.9)$$

kjer je A vnaprej izbran parameter.

Perceptron se uporablja za probleme, pri katerih so vzorci linearno ločljivi. V primeru dvodimenzionalnih vhodnih vektorjev linearna ločljivost pomeni, če si vzorce predstavljamo grafično, da lahko vzorec za učenje razmejimo na dva razreda s premico. Pri vhodnih vektorjih z n dimenzijami pa to pomeni, da jih lahko razmejimo s hiperravnino dimenzije $n - 1$. Za linearno ločljive probleme v primeru perceptrona obstaja preprost algoritem, ki za dano množico podatkov vedno najde optimalno rešitev [11]. V primeru, da vzorci nimajo lastnosti linearne ločljivosti, perceptron ne zagotavlja optimalne rešitve.

Večnivojski perceptron

Večnivojski perceptron je večnivojska aciklična umetna nevrnska mreža, v kateri so zaporedni nivoji med seboj polno povezani. To pomeni, da je vsak nevron v i -tem nivoju povezan z vsemi nevrni iz nivojev $i - 1$ in $i - 2$. Aktivacijska funkcija v srednjih (t.i. skritih) nivojih ni linearna, zato so uporabni tudi za probleme, pri katerih vzorci niso linearno ločljivi. Večnivojski perceptron ima lahko več izhodnih nevronov, kar omogoča klasifikacijo v več razredov in večdimenzionalno regresijo.

V primeru perceptrona z enim skritim nivojem, se je uveljavilo pravilo dobre prakse za izbiro števila skritih nevronov [15]

$$H = (I \cdot + O)/2, \quad (4.10)$$

kjer je H število skritih, I število vhodnih in O število izhodnih nevronov.

4.6.4 Učenje umetnih nevronske mreže

Umetne nevronske mreže se učijo tako, da na podlagi učne množice podatkov z različnimi algoritmi prilagajajo uteži na povezavah med nevrni. Če za nek problem izberemo ustrezno nevronske mrežo in ustrezen algoritem za učenje, potem uteži v večini primerov konvergirajo proti optimalni vrednosti za uporabljeno učno množico vzorcev. V nadaljevanju sledita groba opisa algoritmov za učenje perceptronov in večnivojskih perceptronov.

Algoritem za učenje perceptronov

Zaradi same preprostosti perceptronov je tudi algoritem za njihovo učenje relativno preprost. Najprej definiramo ustrezne spremenljivke, ki jih bomo uporabili v algoritmu prilagajanja uteži.

- $S = (x_1, y_1), \dots, (x_n, y_n)$ je množica parov učnih vzorcev (vhodna spremenljivka, izhodna spremenljivka),
- $x_{j,i}$ je i -ta vrednost v j -tem vhodnem vektorju,
- $w(t)$ je vektor uteži povezav med vhodnimi vozlišči in izhodnim vozliščem in $w_i(t)$ je njegova i -ta komponenta,
- α je vrednost med 0 in 1, ki ji rečemo stopnja učenja (večja kot je, večji bo vpliv učenja na uteži perceptrona).

Algoritem za učenje perceptronov:

1. inicializiramo uteži na povezavah na poljubno vrednost
2. za vsak učni vzorec (x_j, y_j) posodobimo uteži v perceptronu z izvedbo sledečih korakov:
 - (a) izračunamo vrednost, ki bi jo vrnila umetna nevronska mreža za vhodno spremenljivko vzorca: $y'_j = f(x_j \cdot w(t))$, kjer je \cdot skalarni produkt vektorjev
 - (b) posodobimo komponente vektorja utež:
$$w_i(t+1) = w_i(t) + \alpha \cdot (y_j - y'_j) \cdot x_{ji}$$

Algoritem 4.2: Algoritem za učenje perceptronov.

Da se dokazati, da za linarno ločljiv učni vzorec, algoritem konvergira proti optimalni vrednosti [2].

Algoritem za učenje večnivojskih perceptronov

Pri večnivojskih perceptronih se uporablja t.i. algoritem razširjanja vzvratnih napak v kombinaciji z neko optimizacijsko metodo (ponavadi z gradientnim spustom). Metoda izračuna gradient funkcije napake in nato z metodo gradientnega spusta posodobimo uteži v umetni nevronske mreži. Metoda zahteva, da so aktivacijske funkcije nevronov odvedljive. Podrobnosti algoritma razširjanja vzvratnih napak so na voljo v [9].

Poglavje 5

Zasnova in implementacija lastnega ATS

V tem poglavju predstavljamo zasnovo lastnega ATS in njegovo implementacijo. Sklicujemo se teoretično ozadje predstavljeno v prejšnjih poglavjih diplomske naloge. ATS je zasnovan tako, da z uporabo trgovalnih indikatorjev iz področja tehnične analize trgovanja (opisana v poglavju 3) in algoritmov strojnega učenja (opisani v poglavju 4) napoveduje gibanje cene neke dobrine.

5.1 Zgradba našega ATS

ATS je zgrajen iz treh podenot, ki so opisane v razdelku 2.3. V nadaljevanju sledi arhitekturni pregled podatkovne in odločitvene enote v našem zasnovanem sistemu. Naloge izvršilne enote so trivialne in niso problemsko specifične, saj ta deluje le kot vmesni člen med odločitveno enoto in spletnim programskim vmesnikom digitalne borze. Iz tega razloga jih v tem razdelku ne bomo omenjali.

5.1.1 Podatkovna enota

Naloga podatkovne enote je odločitveni enoti zagotoviti podatke v obliki, ki jo ta pričakuje za potrebe odločanja. V našem ATS uporabljamo izključno podatke o transakcijah na spletnih borzah. Podatkov o knjigi naročil (več v razdelku 2.3.1) nismo uporabili. Prav tako nismo uporabili podatkov iz drugih spletnih virov, kot so finančne novice in objave na socialnih omrežjih. Sledijo razdelki z opisom zaporedja nalog, ki jih opravlja naša podatkovna enota.

Pridobivanje podatkov o transakcijah

Ena izmed nalog naše podatkovne enote je sprotno pridobivanje podatkov iz spletnega tržišča o izvedenih transakcijah za določeno preteklo časovno obdobje (analogno procesu predstavljenemu v razdelku 2.3.1). Po pridobitvi vseh transakcij za preteklo časovno obdobje ima podatkovna enota za vsako transakcijo na voljo tri podatke: časovno značko sklenitve transakcije, količino pretečene dobrine in ceno, po kateri se je kupčija sklenila.

Grajenje časovnih rezin

Druga naloga podatkovne enote je iz tekočih transakcij zgraditi časovno rezino in izračunati njene temeljne lastnosti. Transakcije, ki se časovno nahajajo na intervalu, ki ga predstavlja časovna rezina (v primeru našega ATS je interval ena ura), prispevajo svoje podatke k izračunu temeljnih lastnosti časovne rezine - začetne in končne cene, najvišje in najnižje cene ter količine pretečene dobrine (glej razdelek 3.1.2).

Izračun trgovalnih indikatorjev

Naslednja naloga podatkovne enote v našem ATS je za vsako časovno rezino na podlagi njenih temeljnih podatkov in temeljnih podatkov njenih predhodnic (ki si jih mora shranjevati) izračunati vrednosti sledečih trgovalnih indikatorjev. V naslednjem seznamu pomenita besedi "krajše" in "daljše" dolžini periode k pri izračunu trgovalnega indikatorja (glej razdelek 3.2).

- krajše ($k = 7$) in daljše ($k = 14$) preprosto drseče povprečje končne cene,
- krajše ($k = 7$) in daljše ($k = 14$) eksponentno drseče povprečje končne cene,
- MACD indikator z eksponentnima drsečima povprečjema ($k = 12$ in $k = 26$) ter signalnim podatkom ($k = 9$),
- stohastični oscilator, s komponento K ($k = 14$) in komponento D ($k = 3$),
- indeks relativne moči s parametrom $k = 14$,
- dve preprosti drseči povprečji volumna ($k = 7$ in $k = 14$).

Izhod podatkovne enote

Podatkovna enota torej za vsako časovno rezino odločitveni enoti zagotovi vektor z informacijo o indikatorjih opisanih v prejšnjem razdelku. Naša odločitvena enota ne uporablja le indikatorjev trenutne časovne rezine, ampak glede na njene predhodnice in njihove podatke sklepa kupčije. V nadaljevanju je opisana odločitvena enota zasnovanega ATS.

5.1.2 Odločitvena enota

Odločitvena enota v našem ATS vsebuje vse tri podkomponente, opisane v poglavju 2.3.2, alfa model, model tveganja in model cene transakcij.

Alfa model

Alfa model v našem ATS pri odločanju uporablja algoritme strojnega učenja. Pri tem imamo na voljo algoritme kNN, naivni Bayesov klasifikator in umetno nevronska mrežo. Vhodna spremenljivka omenjenih algoritmov je vektor indikatorjev časovne rezine, ki ga zagotovi podatkovna enota našega ATS. Izhodna spremenljivka je diskretna in ima tri možne vrednosti, ki označujejo trgovalne akcije: "kupi", "prodaj" in "ne stori ničesar".

Model tveganja

Model tveganja je preprost. Zasnovan je na način, da vedno sklepamo kupčije le z deležem kapitala. To pomeni, da ne izpostavljam celotnega kapitala posameznim kupčijam, saj bi na ta način v primeru napačnih odločitev lahko imeli velike izgube.

Model cene transakcij

Model cene transakcij je zasnovan tako, da preteklim časovnim rezinam dodeljemo vrednost izhodne spremenljivke z ozirom na pristojbine, ki bi jih plačali, če bi trgovali na realnem tržišču. To pomeni, da preteklim časovnim rezinam, za katere vemo spremembe cen v prihodnosti, izberemo izhodni razred med tremi razredi ("kupi", "prodaj", "ne stori ničesar"). Pristojbine na realnem tržišču znašajo okoli 0.15%. Zato časovnim rezinam, pri katerih je sprememba cene premajhna (za mejo smo izbrali 0.5%), dodelimo razred "ne stori ničesar". S tem zahtevamo, da je trgovalni signal dovolj močen, preden ga upoštevamo. Brez modela cene transakcij bi naš ATS veliko trgoval, kar bi v realnem primeru povzročilo veliko zaračunavanj pristojbin in posledično veliko izgubo.

5.2 Implementacija simulacije ATS

V prejšnjih podpoglavjih je bila opisana arhitekturna zasnova našega ATS. V sklopu diplomske naloge pa smo implementirali tudi njegovo simulacijo, katere opis sledi v nadaljevanju razdelka.

Simuliran ATS smo implementirali v jeziku Java 1.8. Pri tem smo se želeli čim bolj približati realnosti, saj bo tako tudi naše vrednotenje njegovega delovanja verodostojno. Ker v simulaciji trgujemo v realnem okolju, je nemogoče oceniti realno ceno nakupa oz. prodaje v primeru trgovalnega signala. Zato smo v času odločitve za nakupno in prodajno ceno vzeli končno ceno dobrine v časovni rezini, za katero smo izvedli napoved. Poleg nakupne in prodajne cene pa je pri vsakem naročilu pomembno, da upoštevamo tudi pristojbine, ki bi jih zanj plačali, če bi trgovali v realnosti. V naši simulaciji ATS smo za višino pristojbin vzeli 0.15%, kar je sredinska vrednost v ceniku digitalne borze Bitstamp.

5.2.1 Implementacija gradnikov simuliranega ATS

Naš simuliran sistem vsebuje vse tri gradnike sistemov ATS. Podatkovno enoto smo simulirali s predprocesiranjem preteklih (realnih) tržnih podatkov, ki smo jih nato uporabili v odločitveni enoti ATS. Odločitvena enota je realistična in vsebuje vse tri gradnike (glej razdelek 2.3). Izvršilna enota je simulirana s preprostim izpisom kupčij v konzolo, znotraj katere se ATS izvaja, zato ne potrebuje dodatne razlage. Sledi opis implementacij podatkovne enote in odločitvene enote našega ATS.

Implementacija podatkovne enote

Podatki, ki smo jih uporabili v implementaciji podatkovne enote so pretekle transakcije Bitcoinov na digitalni borzi Bitstamp. Podatke smo pridobili iz spletne strani bitcoincharts ¹, ki ponuja datoteke s podatki o vseh preteklih transakcijah za različne digitalne borze kriptovalute Bitcoin. Datoteke so na voljo v formatu s končnico .csv. Vsaka vrstica v omenjenih datotekah vsebuje sledeče tri podatke.

- **časovna značka:** specificira čas sklenitve kupčije,
- **cena:** cena Bitcoinov v ameriških dolarjih, po kateri je bila kupčija sklenjena,
- **količina:** količina Bitcoinov v transakciji.

S pomočjo skript transformiramo zgornje podatke na način, ki je opisan v razdelku 5.1.1. Izhodna datoteka je dokument s končnico .csv, ki za vsako časovno rezino vsebuje izračunane trgovalne indikatorje in nastavljeno izhodno spremenljivko. Služi kot vhodna datoteka za odločitveni del našega simuliranega ATS.

¹<http://www.bitcoincharts.com>

Implementacija odločitvene enote

Odločitvena enota je implementirana na povsem enak način, ki je opisan v poglavju 5.1.2. Implementirani so vsi trije modeli odločitvene enote. Pri tem je alfa model zgrajen modularno, kar pomeni, da imamo možnost preklapljanja med metodami strojnega učenja, ki jih uporabljamo.

Klasifikatorji so implementirani s pomočjo knjižnice Weka [8] in so v skladu s paradigmo rudarjenja podatkovnih tokov. To pomeni, da se učna množica sproti osvežuje, klasifikatorji pa se znanja učijo sproti in pozabljajo staro znanje. Učna množica je velika 200 primerkov, kar je določeno empirično.

Klasifikator kNN smo implementirali s pomočjo razreda `weka.classifiers.lazy.IBk` iz knjižnice `weka`. Funkcija sosednosti, ki smo jo uporabili je evklidska razdalja, saj je naš vhodni vektor zvezen. Za parameter k smo empirično izbrali vrednost deset. Novemu primerku klasifikator kNN dodeli vrednost večinskega razreda med desetimi najbližjimi sosedi. Pri tem se vpliv desetih najbližjih sosedov uteži, tako da imajo večji vpliv bližji primerki (glej razdelek 4.4).

Naivni Bayesov klasifikator je bil implementiran s pomočjo razreda `weka.classifiers.bayes.NaiveBayes` brez dodatnih nastavitev.

Umetna nevronska mreža je implementirana v obliki večnivojskega perceptrona s pomočjo razreda `weka.classifiers.functions.MultilayerPerceptron`. Sestavljena je iz treh nivojev, pri čemer ima vhodni nivo toliko nevronov, kolikor je atributov primerka. Število nevronov v srednjem nivoju se izračuna po pravilu dobre prakse (glej razdelek 4.6.3). Izhodni nevroni pa so trije, saj imamo tri izhodne razrede ("kupi", "prodaj", "ne stori ničesar").

Poglavje 6

Rezultati in diskusija

6.1 Načini ovrednotenja uspešnosti ATS

Glavna mera za ovrednotenje sistemov ATS je profitabilnost. Slednja je močno odvisna od gibanja cen v obdobju, v katerem trgujemo. Trivialen razmislek nam pove da, če je cena na začetku obdobja nekega trgovanja večja kot na koncu, je možnost profitabilnosti ATS manjša. Analogno velja tudi v obratnem primeru. Iz tega vidika je pomembno, da ATS testiramo na obdobjih, ki vsebujejo različna gibanja cen (rasti, padce, nihanja itd.). Ta obdobja morajo biti dovolj dolga, da izničimo element sreče. Slednji bi nam v krajših časovnih obdobjih lahko prinesel nezanesljive rezultate.

V primeru našega ATS je za razumevanje njegove profitabilnosti pomembno izmeriti uspešnost algoritma strojnega učenja v alfa modelu (glej razdelek 5.1.2). Uspešnost algoritma strojnega učenja bi morala sovpadati s profitabilnostjo ATS. Ker imamo klasifikacijski problem, pri katerem razvrščamo primerke (časovne rezine) v tri razrede: "kupi", "prodaj" in "ne stori ničesar", uspešnost algoritma strojnega učenja izmerimo s pomočjo matrike napake, natančnosti, priklica in mere F (glej razdelek 4.2.1). Ker ima naša izhodna spremenljivka tri možne razrede, je naša matrika napake dimenzije 3. Razredi so oštevilčeni po vrstnem redu: "ne stori ničesar" - 0, "kupi" - 1, "prodaj" - 2. To pomeni, da pri vseh vektorjih v rezultatih (na-

tančnosti, priklicu itd.) prva komponenta ustreza razredu "ne stori ničesar", druga razredu "kupi" in tretja razredu "prodaj". Enaka razporeditev velja tudi za matrike napak.

Za potrebe vrednotenja uspešnosti algoritmov strojnega učenja smo njihove rezultate primerjali s sledečim naivnim algoritmom. Algoritem deluje tako, da za vsako časovno rezino napove enak izhodni razred kot za svojo predhodnico. Tak način napovedovanja je preprost, vendar ni povsem naključen, saj upošteva empirično spoznanje tehnične analize trgovanja, da se cene gibljejo trendovsko [6].

Poleg prej omenjenega naivnega algoritma, smo vključili še algoritem naključnostnega napovedovanja. Slednji deluje tako, da za vsako časovno rezino napove izhodni razred glede na porazdelitev razredov v vhodnih podatkih. Rezultat je torej naključno izbran razred izmed "kupi", "prodaj", "ne stori ničesar" z ustrezno verjetnostjo. Tak način napovedovanja je v praksi spodnja meja uspešnosti, ki bi jo morali izboljšati z vsemi algoritmi strojnega učenja.

6.2 Rezultati

6.2.1 Rezultati ATS pri trgovanju z Bitcoinom

Algoritme strojnega učenja smo testirali na daljšem obdobju gibanja cen Bitcoina, ki je dolgo dve leti in tri mesece. Začetna in končna cena obdobja sta podobni, v njem pa se je zvrstilo mnogo obdobj rasti, padcev in nihanj cene. Na začetku trgovanja ima ATS začetni kapital 1000 ameriških dolarjev.

Osnovni podatki obdobja obdobja trgovanja z Bitcoinom so sledeči.

- **trajanje obdobja:** od 13.11.2013 do 19.1.2016,
- **cena Bitcoina na začetku trgovanja:** 363.00 dolarjev,
- **cena Bitcoina na koncu trgovanja:** 384.47 dolarjev,
- **razlika v ceni:** +5.91%,
- **število enournih časovnih rezin (primerkov) v obdobju:** 19127.



Slika 6.1: Graf cen kriptovalute Bitcoin z dolžino časovne rezine en dan v obdobju od 13.11.2013 do 19.1.2016.

algoritem	utežena mera F	št. kupčij	profitabilnost
kNN	0.549	1929	+10.14%
umetna nevrnska mreža	0.545	2291	+7.67%
naivni Bayesov klasifikator	0.500	3057	-51.33%
naivni algoritem	0.500	5412	-99.57%
naključni algoritem	0.455	5510	-97.84%

Tabela 6.1: Rezultati ATS pri trgovanju z Bitcoinom.

Na sliki 6.1 je gibanje cene Bitcoina v obdobju dveh let in pol. V tabeli 6.1 so rezultati našega ATS z različnimi algoritmi strojnega učenja, vključno z rezultati naivnega algoritma in naključnega napovedovanja (glej razdelek 6.1). V rezultatih so zaradi jasnosti in berljivosti le najbolj ključni podatki - utežena mera F in rezultati trgovanja. Natančnejši rezultati in ostale mere so na voljo v dodatku na koncu (8).

6.2.2 Rezultati ATS pri trgovanju z Litecoinom

Naš ATS smo testirali tudi na drugi množici podatkov in sicer na podatkih gibanja cene kriptovalute Litecoin, ki je ena izmed alternativ Bitcoinu. Na trgu je Litecoin prisoten že dovolj časa, da smo uspeli dobiti podatke o trgovanju z njim za enoletno obdobje. Iz sledečih rezultatov smo izpustili naivno napovedovanje in naključni algoritem, katerih rezultati pri trgovanju z Bitcoinom so očitno slabi. Tudi pri trgovanju z Litecoinom smo imeli na začetku trgovanja začetni kapital 1000 ameriških dolarjev. Osnovni podatki obdobja obdobja trgovanja z Litecoinom so sledeči.

- **trajanje obdobja:** od 1.11.2014 do 7.9.2015,
- **cena Litecoina na začetku trgovanja:** 3.63 dolarja,
- **cena Litecoina na koncu trgovanja:** 3.00 dolarja,
- **razlika v ceni:** -17.36% ,
- **število enournih časovnih rezin (primerkov) v obdobju:** 7450.



Slika 6.2: Graf cen kriptovalute Litecoin v obdobju od 1.11.2014 do 7.9.2015.

algoritem	utežena mera F	št. kupčij	profitabilnost
kNN	0.523	798	-26.05%
umetna nevronska mreža	0.537	1078	+4.43%
naivni Bayesov klasifikator	0.522	887	-56.98%

Tabela 6.2: Rezultati ATS pri trgovanju z Litecoinom.

Na sliki 6.2 je gibanje cene Litecoina v obdobju približno enega leta. V tabeli 6.2 sledijo rezultati našega ATS z uporabo algoritma kNN, umetne nevronske mreže in naivnim Bayesovim klasifikatorjem. Prav tako kot v primeru Bitcoina, so razširjeni rezultati na voljo v dodatku (8).

6.3 Diskusija

Vse algoritme smo testirali na enakem časovnem obdobju in uporabili enake mere uspešnosti za njihovo ovrednotenje, kar pomeni, da so mere primerljive med seboj. Poleg tega smo ATS testirali na različnih podatkih in dovolj dolgih obdobjih, da nam pridobljeni rezultati zadovoljivo prikazujejo realno uspešnost našega pristopa.

6.3.1 Uspešnost napovedovanja

Iz rezultatov napovedovanja našega ATS lahko vidimo, da je sam problem napovedovanja gibanja cene glede na preteklost zelo težek, kar pomeni, da je uporabnega znanja v naših podatkih malo. Kljub temu pa se iz rezultatov vidi, da se vsi algoritmi strojnega učenja obnašajo bolje kot naivni algoritem in naključno napovedovanje, kar dokazuje, da znanje v podatkih je in da smo ga uspešno uporabili. Najbolje uspešnost napovedovanja povzema utežena mera F , vendar pa je za razumevanje njene vrednosti pomembno, da upoštevamo ostale mere, iz katerih je bila le-ta izračunana (natančnost, priklic, mera F).

Najslabše se je glede na uteženo mero F obnašal naključni algoritem. Utežena F mera je pri njem vidno manjša kot pri ostalih pristopih, kar je v skladu s pričakovanji in dokazuje obstoj znanja v naših podatkih. Naključni algoritem namreč tega znanja ne upošteva, pač pa napoveduje v skladu z porazdelitvijo izhodnega razreda. O tem priča tudi dejstvo, da so komponente vektorjev mere F , natančnosti in priklica enake in odražajo ravno porazdelitev izhodnega razreda na naši množici primerkov.

Vrednost utežene F mere naivnega algoritma v primerjavi z naključnim algoritmom priča o obstoju trendov v gibanju cene. Boljši rezultat odraža tudi primerjava vrednosti komponent v vektorjih natančnosti, priklicu in F meri, ki so vidno višje. Iz tega lahko sklepamo da gibanje cen ni naključno in je bolj verjetno, da bo izhodni razred trenutne časovne rezine enak izhodnemu razredu predhodne časovne rezine. Opazimo lahko tudi, da so vektorji

natančnosti, priklica in mere F ponovno enaki. To sledi iz preprostega razmisleka, da sta za naš naivni algoritem velikosti množice primerkov (vrednosti v imenovalcu iz razdelka 4.2.1) enaki.

Algoritmi strojnega učenja bi v praksi morali izboljšati rezultate naivnega algoritma, kar pa za naivni Bayesov klasifikator ne velja. Ima namreč enako uteženo mero F kot naivni algoritem. Iz druge komponente mere F pa lahko sklepamo, da je naivni Bayesov klasifikator dosti bolj uspešen glede napovedovanja razreda "kupi". Kljub temu pa ima prva komponenta mere F (pripada razredu "ne stori ničesar") največjo utež pri izračunu utežene mere F , zato je zaradi njene nižje vrednosti ta enaka. Iz tega rezultata lahko sklepamo, da bi za večjo uspešnost morali izboljšati vrednost prve komponente F mere.

Višjo prvo komponento mere F in tudi posledično najboljši rezultat pri napovedovanju imata algoritem kNN in umetna nevronska mreža. Oba vidno izboljšata uteženo mero F v primerjavi z naivnim Bayesovim klasifikatorjem. Umetna nevronska mreža se od algoritma kNN malce bolje obnese pri napovedovanju razreda "kupi", a slabše pri napovedovanju razreda "ne stori ničesar". Posledično ima malce višjo vrednost utežene mere F algoritem kNN.

V primeru napovedovanja cene kriptovalute Bitcoin smo z metodami strojnega učenja uteženo F mero v primerjavi z naključnim algoritmom izboljšali za 7% vrednosti. To sicer ni izboljšava, ki bi jo želeli v večini problemov, a je v našem primeru zadovoljiva, saj je iz mnogih virov [5, 12, 1] razvidno, da je problem napovedovanja gibanja cen težak in so že zelo majhne razlike v uspešnosti lahko ključen faktor pri dolgoročni profitabilnosti ATS.

6.3.2 Uspešnost trgovanja

Uspešnost trgovanja merimo s profitabilnostjo. Slednja je odvisna od veliko dejavnikov ne le od napovedne uspešnosti naših odločitev. V primeru pravilne klasifikacije v razreda "kupi" in "prodaj" so lahko dejanske spremembe cen zelo različne, kar pomeni, da so nekatere odločitve bolj pomembne od drugih. Kljub temu pa bi morala biti na dolgi rok napovedna uspešnost in profitabilnost v korelaciji. To potrjujejo tudi naši rezultati.

Iz rezultatov pri trgovanju s kriptovaluto Bitcoin vidimo, da se naključno napovedovanje in naivno napovedovanje v realnosti ne obneseta. Problem zanj namreč predstavljajo pristojbine. Če le teh ne bi bilo, bi od naključnega napovedovanja pričakovali, da bo njegova profitabilnost približno enaka razliki cene v trgovalnem obdobju. Naivno napovedovanje bi se moralo v idealnem primeru brez pristojbin odrezati še boljše od naključnega. V realnosti pa iz rezultatov vidimo, da bi s tema pristopoma na dolgi rok izgubili skoraj celoten vložek.

Kljub temu pa se iz rezultatov vidi, da lahko z uporabo strojnega učenja dosežemo profitabilnost. Umetna nevronska mreža in algoritem kNN imata najvišjo napovedno točnost, kar se posledično odraža tudi pri profitabilnosti. V obdobju, ko je cena Bitcoina zrasla za 5.91%, smo z omenjenima metodama izboljšali to razliko. Z algoritmom kNN smo zaslužili 10.14%, medtem ko smo pri umetni nevronske mreži zaslužili 7.67%. Naivni Bayesov klasifikator se po pričakovanjih (glede na rezultate napovedne uspešnosti) ni obnesel najbolje in je prinesel izgubo -51.33%. Razlog za to je predvsem v velikosti prvih komponent mer F , ki predstavlja uspešnost napovedovanja razreda "ne stori ničesar" in je pri naivnem Bayesovem klasifikatorju nižja. Sklepamo lahko, da nam v primeru umetne nevronske mreže in algoritma kNN uspe bolje prepoznati časovne rezine, pri katerih se cena bistveno ne spremeni in se poteza na trgu ne izplača. Odraz tega je tudi primerjava števila kupčij pri trgovanju z umetno nevronske mreže in algoritmom kNN z ostalimi metodami. Vidimo namreč, da sta umetna nevronska mreža in algoritem kNN manj trgovala, kar v realnosti pomeni, da so se manjkrat obračunale pristojbine.

Rezultate našega ATS smo potrdili tudi pri trgovanju s kriptovaluto Litecoin. Cena Litecoina je v trgovalnem obdobju padla za 17.36%. To pomeni, da se trgovanje v tem obdobju ni izplačalo. Ker pa je obdobje trgovanja z Litecoinom krajše, so tudi pridobljeni rezultati malce manj zanesljivi. Kljub temu se rezultati ujemajo z rezultati trgovanja z Bitocinom in jih v večji meri potrjujejo. Najbolje se je, tako v napovedovanju cene kot tudi pri trgovanju z Litecoinom, obnesla umetna nevronska mreža. V obdobju, kjer je cena Litecoina padla, je umetna nevronska mreža dosegla profit 4.43%, kar je zelo dober rezultat. Na drugi strani se je algoritem kNN obnesel slabše in je zabeležil 26.05% izgube. Po pričakovanjih se je najslabše obnesel naivni Bayesov klasifikator, ki je zabeležil izgubo 56.98%, kar se ujema tudi z napovedno uspešnostjo.

Poglavje 7

Zaključek

V diplomski nalogi smo raziskali avtomatske trgovalne sisteme s podporo metod tehnične analize trgovanja, algoritmov strojnega učenja in metodologij rudarjenja podatkovnih tokov. Sprva smo se dotaknili gradnikov teh sistemov in predstavili njihove glavne naloge s pomočjo programskega vmesnika spletne borze Bitcoina Bitstamp. Sledil je pregled ključnih pojmov in metod tehnične analize trgovanja. Poseben poudarek smo dali meritvam, ki kvantificirajo trg - trgovalnim indikatorjem. V nadaljevanju smo definirali problem trgovanja z vidika strojnega učenja ter preučili nekatere algoritme nadzorovanega učenja. Nato smo zasnovali lasten ATS. Odločitveno enoto smo zasnovali z uporabo metod strojnega učenja: k -najbližjih sosedov, umetno nevronska mrežo in naivnim Bayesovim klasifikatorjem. Tem pristopom smo za namene testiranja in vrednotenja dodali naivno napovedovanje in naključno napovedovanje. ATS smo implementirali kot simulacijo in njegovo delovanje testirali na realnih tržnih podatkih gibanja cene Bitcoina. Uporabili smo podatke o transakcijah z Bitstampa. Za potrditev pridobljenih rezultatov, smo le-te razširili z dodatno množico podatkov in sicer s podatki o gibanju cene kriptovalute Litecoin.

Problem napovedovanja cene na podlagi preteklih podatkov se je z vidika strojnega učenja izkazal za zelo zahtevnega, saj se trg v večji meri obnaša nepredvidljivo. Profitabilnost samega ATS je močno odvisna od trenda cene v celotnem obdobju, poleg tega pa imajo različne trgovalne odločitve različen vpliv. Uspelo nam je pokazati, da se napovedna uspešnost algoritma strojnega učenja in profitabilnost trgovanja v večji meri ujemata. Naivno napovedovanje in naključno napovedovanje v praksi ne delujeta zaradi pristožbin, zaradi katerih se veliko kupčij ne izplača. Najbolje se je obnesla umetna nevronska mreža, ki je prinesla profit tako pri trgovanju z Bitcoinom, kot tudi pri trgovanju z Litecoinom.

Možnih izboljšav sistema je veliko. Obstaja še veliko drugih trgovalnih indikatorjev, ki jih v sklopu naše raziskave nismo uporabili. Poleg tega smo parametre uporabljenih indikatorjev nastavili glede na priporočila tehnične analize trgovanja [6] in zato niso nujno optimalni. Za izboljšanje performanc bi lahko parametre indikatorjev optimizirali s pomočjo optimizacijskih metod (npr. gradientnega spusta). V sam ATS bi lahko vključili dodatne mehanizme, ki sproti merijo uspešnost napovedi. Na podlagi te uspešnosti bi lahko paralelno uporabljali več metod strojnega učenja, pri čemer bi napovedovali glede na tisto, ki bi trenutno imela najboljše rezultate. Z mehanizmom sprotnega pomnjenja napovedne uspešnosti bi lahko v nepredvidljivih obdobjih preprečili trgovanje in s tem omejili izgubo. Poleg navedenih možnih izboljšav pa bi verjetno največji doprinos prinesla analiza informacij iz pisnih finančnih virov, ki bi jo vključili s pomočjo tekstovnega rudarjenja finančnih novic in socialnih omrežij, saj iz različnih virov vemo, da te pomembno vplivajo na gibanja cene [3].

Poglavje 8

Dodatek

8.1 Razširjeni rezultati

8.1.1 Rezultati ATS pri trgovanju z Bitcoinom

Rezultati kNN

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 10017 & 831 & 837 \\ 2281 & 877 & 538 \\ 2600 & 674 & 472 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.672 \\ 0.368 \\ 0.256 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.857 \\ 0.237 \\ 0.126 \end{bmatrix}$$

- mera F :

$$\begin{bmatrix} 0.754 \\ 0.289 \\ 0.169 \end{bmatrix}$$

- utežena mera F : 0.549

Rezultati trgovanja:

- število kupčij: 1929
- znesek po trgovanju: 1101.36 dolarjev
- razlika v premoženju: +10.14%

Rezultati umetne nevronske mreže

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 9076 & 1453 & 1156 \\ 1932 & 1141 & 623 \\ 2364 & 764 & 618 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.679 \\ 0.340 \\ 0.258 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.777 \\ 0.309 \\ 0.165 \end{bmatrix}$$

- mera F :

$$\begin{bmatrix} 0.724 \\ 0.324 \\ 0.201 \end{bmatrix}$$

- utežena mera F : 0.545

Rezultati trgovanja:

- število kupčij: 2291
- znesek po trgovanju: 1076.66 dolarjev
- razlika v premoženju: +7.67%

Rezultati naivnega Bayesovega klasifikatorja

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 6682 & 2435 & 2568 \\ 1129 & 1591 & 976 \\ 1560 & 1261 & 925 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.713 \\ 0.301 \\ 0.207 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.572 \\ 0.430 \\ 0.247 \end{bmatrix}$$

- mera F :

$$\begin{bmatrix} 0.635 \\ 0.354 \\ 0.225 \end{bmatrix}$$

- utežena mera F : 0.500

Rezultati trgovanja:

- število kupčij: 3057
- znesek po trgovanju: 486.74 dolarjev
- razlika v premoženju: -51.33%

Rezultati naivnega algoritma

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 7949 & 1858 & 1878 \\ 1714 & 869 & 1113 \\ 2022 & 969 & 755 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.680 \\ 0.235 \\ 0.202 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.680 \\ 0.235 \\ 0.202 \end{bmatrix}$$

- mera F :

$$\begin{bmatrix} 0.680 \\ 0.235 \\ 0.202 \end{bmatrix}$$

- utežena mera F : 0.500

Rezultati trgovanja:

- število kupčij: 5412
- znesek po trgovanju: 4.27 dolarjev
- razlika v premoženju: -99.57%

Rezultati naključnega algoritma

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 7244 & 2262 & 2239 \\ 2206 & 716 & 740 \\ 2296 & 683 & 741 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.617 \\ 0.196 \\ 0.199 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.617 \\ 0.196 \\ 0.199 \end{bmatrix}$$

- mera F :

$$\begin{bmatrix} 0.617 \\ 0.196 \\ 0.199 \end{bmatrix}$$

- utežena mera F : 0.455

Rezultati trgovanja:

- število kupčij: 5510
- znesek po trgovanju: 21.57 dolarjev
- razlika v premoženju: -97.84%

8.1.2 Rezultati ATS pri trgovanju z Litecoininom

Rezultati kNN

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 3703 & 474 & 305 \\ 1015 & 264 & 216 \\ 984 & 247 & 242 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.649 \\ 0.268 \\ 0.317 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.826 \\ 0.177 \\ 0.164 \end{bmatrix}$$

- mera F :

$$\begin{bmatrix} 0.727 \\ 0.213 \\ 0.216 \end{bmatrix}$$

- utežena mera F : 0.523

Rezultati trgovanja:

- število kupčij: 798
- znesek po trgovanju: 739.5 dolarjev
- razlika v premoženju: -26.05%

Rezultati umetne nevrnske mreže

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 3544 & 492 & 446 \\ 884 & 273 & 338 \\ 836 & 267 & 370 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.673 \\ 0.265 \\ 0.321 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.791 \\ 0.183 \\ 0.251 \end{bmatrix}$$

- F mera:

$$\begin{bmatrix} 0.727 \\ 0.216 \\ 0.282 \end{bmatrix}$$

- utežena mera F : 0.537

Rezultati trgovanja:

- število kupčij: 1078
- znesek po trgovanju: 1044.34 dolarjev
- razlika v premoženju: +4.43%

Rezultati naivnega Bayesovega klasifikatorja

Rezultati napovedovanja:

- matrika napake:

$$\begin{bmatrix} 3001 & 805 & 676 \\ 662 & 467 & 366 \\ 656 & 431 & 386 \end{bmatrix}$$

- natančnost:

$$\begin{bmatrix} 0.695 \\ 0.274 \\ 0.270 \end{bmatrix}$$

- priklic:

$$\begin{bmatrix} 0.670 \\ 0.312 \\ 0.262 \end{bmatrix}$$

- mera F :

$$\begin{bmatrix} 0.682 \\ 0.292 \\ 0.266 \end{bmatrix}$$

- utežena mera F : 0.522

Rezultati trgovanja:

- število kupčij: 887
- znesek po trgovanju: 430.19 dolarjev
- razlika v premoženju: -56.98%

Literatura

- [1] Khalid Alkhatib, Hassan Najadat, Ismail Hmeidi, and Mohammed K Ali Shatnawi. Stock price prediction using k-nearest neighbor (knn) algorithm. *International Journal of Business, Humanities and Technology*, 3(3):32–44, 2013.
- [2] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [3] Jacob Boudoukh, Ronen Feldman, Shimon Kogan, and Matthew Richardson. Which news moves stock prices? a textual analysis. Technical report, National Bureau of Economic Research, 2013.
- [4] Sabrina Buti, Barbara Rindi, and Ingrid M Werner. Diving into dark pools. *Charles A. Dice Center Working Paper*, (2010-10), 2011.
- [5] Matthieu Duvinage, Paolo Mazza, and Mikael Petitjean. The intra-day performance of market timing strategies and trading systems based on japanese candlesticks. *Quantitative Finance*, 13(7):1059–1070, 2013.
- [6] Robert D Edwards, John Magee, and WHC Bassetti. *Technical analysis of stock trends*. CRC Press, 2007.
- [7] Fernando Fernandez-Rodriguez, Christian Gonzalez-Martel, and Simon Sosvilla-Rivero. On the profitability of technical trading rules based on artificial neural networks:: Evidence from the Madrid stock market. *Economics letters*, 69(1):89–94, 2000.

-
- [8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [9] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605. IEEE, 1989.
- [10] Igor Kononenko and Marko Robnik Šikonja. *Inteligentni sistemi*. Založba FE in FRI, 2010.
- [11] Mario Marchand, Mostefa Golea, and Pál Ruján. A convergence theorem for sequential learning in two-layer perceptrons. *EPL (Europhysics Letters)*, 11(6):487, 1990.
- [12] John M Marynowski, Catalin D Voinescu, Stefan Puscasu, and Thomas M O’donnell. Automated trading system in an electronic trading exchange, February 13 2007. US Patent 7,177,833.
- [13] LR Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 2001.
- [14] Rishi K Narang. *Inside the Black Box: A Simple Guide to Quantitative and High Frequency Trading*, volume 883. John Wiley & Sons, 2013.
- [15] Seyed Reza Shahamiri and Siti Salwah Binti Salim. Real-time frequency-based noise-robust automatic speech recognition using multi-nets artificial neural networks: A multi-views multi-learners approach. *Neuro-computing*, 129:199–207, 2014.
- [16] Ying Yang and Geoffrey I Webb. A comparative study of discretization methods for naive-bayes classifiers. In *Proceedings of PKAW*, volume 2002. Citeseer, 2002.