Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Benjamin Fele

Modeliranje trendov kriptovalutnih trgov z uporabo tekstovnih podatkov

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Tomaž Curk

Ljubljana, 2017

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Borze kriptovalut so podvržene podobnim mehanizmom, kot veljajo za ostale borze. Vlagatelji in trg se odzivajo na dogajanje na trgu in tako določajo spreminjanje cen. Dogajanje na trgu se odraža in je tudi določeno na podlagi spletnih novic o trgovanju. Preučite obstoječe pristope napovedovanja trendov trgovanja. Z uporabo metod tekstovnega rudarjenja razvite napovedni model za napovedovanje trendov spreminjanja cen kriptovalut. Model naj uporablja tekstovne in druge podatke o trgu kriptovalut. Model ovrednotite na realnih podatkih o dogajanju na borzah kriptovalut.

Zahvaljujem se svojemu mentorju doc. dr. Tomažu Curku za izkazano podporo pri pisanju diplomske naloge.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
API	application programming interface	vmesnik uporabniškega programa
AUC	area under curve	površina pod krivuljo
CA	classification accuracy	klasifikacijska točnost
HTML	hypettext markup language	jezik za označevanje nadbesedila
JSON	JavaScript object notation	zapis JavaScript objekta
LDA	latent Dirichlet allocation	latentna Dirichletova alokacija
ROC	receiver operating characteristic	krivulja ROC
TF-IDF	term frequency - inverse document frequency	frekvenca besed - inverzna frekvenca dokumenta
URL	uniform resource locator	enolični krajevnik vira

Povzetek

Naslov: Modeliranje trendov kriptovalutnih trgov z uporabo tekstovnih podatkov

Avtor: Benjamin Fele

Cilj diplomskega dela je izgradnja modela, ki napoveduje trende vrednosti kriptovalut na podlagi podatkov spletne borze Poloniex. Problem je zanimiv zaradi potencialnih možnosti avtomatskega trgovanja s kriptovalutami, ki bi jih uspešen model omogočal. Za napovedovanje smo uporabili novice iz obravnavanega področja, ki so bile pridobljene iz spletne strani Reddit. Poleg tekstovnih podatkov so za napovedovanje uporabljeni tudi numerični podatki vrednosti kriptovalut pred objavo novice. Problem smo zastavili kot trirazredno klasifikacijo. Z uporabo metode *TF-IDF*, informacije o sentimentu, polariteti ter podatkih o člankih in stanju trga pred objavo besedila je bila dosežena $50,8\%$ klasifikacijska točnost na validacijski množici ter $49,3\%$ klasifikacijska točnost na testni množici. Za učenje in napovedovanje smo uporabili metodo podpornih vektorjev. Kljub relativno dobri klasifikacijski točnosti, model v praksi verjetno ne bi dajal dobičkonosnih napovedi.

Ključne besede: tekstovno rudarjenje, podatkovno modeliranje, metoda podpornih vektorjev, kriptovalute, borza.

Abstract

Title: Modeling cryptocurrency market trends using textual data

Author: Benjamin Fele

The aim of this work is to build a model that predicts cryptocurrency trends based on data from the Poloniex online market. The problem is interesting because of the potential for automated cryptocurrency trading that a successful model would allow. For the forecast, we used the news from the subject area, which were obtained through the Reddit website. In addition to textual data, numerical market data before publishing the news is also used for forecasting. We approached the problem as a three-class classification problem. Using the *TF-IDF* method, sentiment information, polarity, and article and market information before the publication of the text, we achieved $50,8\%$ classifying accuracy on the validation set and $49,3\%$ classification accuracy on the test set. We used support vector machines for learning and prediction. We have found that in practice, despite the significant classification accuracy, the model is unlikely to yield profitable returns.

Keywords: text mining, data modeling, support vector machines, cryptocurrencies, stock market.

Poglavje 1

Uvod

Napovedovanje trendov vrednostnih papirjev je privlačna tema za raziskovanje zaradi potencialnih možnosti zaslužkov. Posamezniki ali skupine, ki so se v začetkih borze morale zanašati na lastne občutke o prihodnjih trendih vrednostnih papirjev, so skozi čas razvili finančne inštrumente in napredne trgovalne strategije, ki povečujejo potencialne zaslužke. Panoga napovedovanja je občutno rast dosegla s povečanjem računske moči računalnikov in popularizacije interneta, ki omogočata trgovanje brez potrebe fizične prisotnosti na borzi ter avtomatsko trgovanje. Slednje naj bi dandanes obsegalo kar 60% vsega trgovanja borz.

S kriptovalutami, ki se sicer še uveljavljajo kot plačilno sredstvo, je mogoče trgovati po enakih principih kot veljajo za tradicionalne vrednostne papirje, na spletu pa je mogoče najti mnogo borz, ki to omogočajo. Slabša uveljavljenost omogoča višjo potencialno rast, kar je še posebej privlačno ob dejstvu, da v primerjavi s tradicionalnimi borzami na borzah kriptovalut še ne dominirajo agenti avtomatskega trgovanja oziroma so ti manj kompleksni.

Za cilj diplomske naloge smo si zadali implementacijo modela, ki bi bil uspešen pri napovedovanju trendov kriptovalut. Delo smo začeli s pridobivanjem člankov in numeričnih podatkov kriptovalut, kar je opisano v poglavju 2. Sledila je gradnja atributov in implementacija metod (poglavje 3), ki so ovrednotene v poglavju 4.

1.1 Kriptovalute in borza Poloniex

Kriptovalute so plačilno sredstvo, ki temelji na tehnologiji *blockchain*. Tehnologija omogoča decentralizirano shranjevanje in izmenjavo informacij. Za vsak nov zapis se ustvari nov blok, ki nosi informacijo o tem zapisu in povezavo na prejšnjega. Kot prva kriptovaluta in primer splošne uporabe se šteje *Bitcoin*, katerega prva implementacija sega v leto 2009.

Po uveljavitvi kriptovalute *Bitcoin* je bilo razvitih še mnogo drugih valut, ki temeljijo na podobnih tehnologijah in odpravljajo potencialne težave *Bitcoin*. Zaradi možnosti pohitritve, pocenitve in decentralizacije transakcij celotna panoga v zadnjih letih doživlja rast. Priljubljene kriptovalute po [7], katerih vrednost je v letu 2016 narasla so Bitcoin (BTC), Monero (XMR), Ripple (XRP), Ethereum (ETH), Ethereum Classic (ETC), Dash (DASH) in Litecoin (LTC).

Z vsemi od naštetih valut je možno trgovati na borzi Poloniex, ki je ena najbolj priljubljenih spletnih borz. V času pisanja diplomske naloge ponuja trgovanje s skupaj 66 kriptovalutami, v obdobju za katerega je bila izvedena klasifikacija besedil pa je bilo teh vseh skupaj 77, ki so naštete v prilogi A.

1.2 Napovedovanje cen kriptovalut

Komercialne aplikacije podatkovnega rudarjenja v namen napovedovanja trendov kriptovalut so največkrat implementirane v obliki avtomatskega trgovanja, ki v večini primerov temelji na tehnični analizi (t.j., napovedovanje na podlagi kvantitativnih podatkov o stanju trga). Med bolj priljubljene spletne ponudnike avtomatskega trgovanja s kriptovalutami spadata Haasbot (`www.haasonline.com`) in Cryptotrader (`cryptotrader.org`).

Ponudnikov storitev na področju fundamentalne analize (t.j., trgovanje na podlagi kvalitativnih podatkov o stanju trgovanih entitet) na področju kriptovalut nismo zasledili. Našli smo dva trenutno aktualna ponudnika na področju tradicionalnih vrednostnih papirjev: Sentimentrader (`sentimentrader.com`) in Trump2Cash (`https://github.com/maxbbraun/trump2cash`).

Malo v praksi dobičkonosnih produktov oziroma storitev je sicer javnih, kar je v nasprotju z relativno veliko akademskimi raziskavami, ki kažejo, da je napovedovanje trendov s pomočjo različnih načinov analize trga mogoče. Večina del, opisanih v [3], problem rešuje s klasifikacijo in pri tem dosega signifikantno klasifikacijsko točnost. V nasprotju s klasifikacijo, regresija daje slabše rezultate, ki so v teoriji redkeje dobičkonosni. Klasifikacija je največkrat realizirana z uporabo dveh do treh razredov in tako tehnične kot fundamentalne analize. Modeli so najpogosteje naučeni z metodo podpornih vektorjev, metodo naključnih gozdov ali naivnim Bayesom. V primeru dvorazrednega problema klasifikacijska točnost variira od 55% do 84%, pri primeru trirazrednega pa so doseženi rezultati med 40% in 67%. Vse raziskovalno delo, ki smo ga zasledili, je opravljeno na podatkih običajnih borz in ne borz kriptovalut.

Poglavje 2

Podatki

Kot izhodišče za napovedovanje trendov kriptovalut so uporabljeni tekstovni podatki. Izbrani tekstovni podatki so članki iz področja kriptovalut.

Kvalitativne podatke dopolnjujejo tudi kvantitativni podatki stanja trga, kot sta obseg trgovanja in splošen trend pred objavo besedila. Za preverjanje točnosti napovedi je izračunana sprememba cene na podlagi trgovanja v določenem časovnem oknu po objavi besedila. V obeh primerih so uporabljeni podatki borze Poloniex.

Podatki so bili shranjeni v sistemu za upravljanje podatkovnih baz *MongoDB* različice 3.4.0. Shranjevaje je realizirano v ne-relacijskih podatkovnih bazah, v kateri so podatki nadalje strukturirani v zbirke. *MongoDB* za hranjenje uporablja format JSON in omogoča shranjevanje večine pogostih podatkovnih struktur vključno z datumskimi objekti, slovarji in seznami.

Pridobljeni podatki obsegajo enoletno obdobje, od 1. 11. 2015 do 31. 10. 2016. Shranjeni so v dveh podatkovnih bazah:

- `crypto_prices`: za vsako kriptovaluto je ustvarjena zbirka, ki hrani podatke o cenah za izbrano časovno obdobje. Vseh zbirk je 77, imena pa ustrezajo kraticam kriptovalut.
- `news`: v treh zbirkah so hranjeni podatki o novicah:
  - `articles`: podatki o člankih,

- `authors`: podatki o avtorjih člankov,
- `sources`: podatki o spletnih straneh, od koder so bili preneseni članki.

V nadaljevanju je podrobneje opisano pridobivanje in shranjevanje besedil ter podatkov o vrednostih kriptovalut.

2.1 Cene kriptovalut

Borza Poloniex hrani podatke o vseh cenah kriptovalut, s katerimi so na borzi trgovali v preteklosti. Omogoča pridobitev cen za okna 5 minut, 15 minut, 30 minut, 2 uri, 4 ure in 1 dan. Ker je cilj diplomske naloge napovedovanje cen v realnem času, je bilo izbrano časovno okno petih minut, ki predstavlja največjo mogočo ločljivost napovedi v tem delu.

Prenesene so bile vrednosti 77 kriptovalut, ki so predstavljene v seznamu slovarjev v formatu JSON. Vsak slovar vsebuje informacije o času v formatu *Unix* in podatke o ceni v izbranem obdobju. Časi objav so v greenwiškem časovnem pasu kar zagotavlja konsistentnost napovedi. Pri shranjevanju v podatkovno bazo čas služi kot atribut `_id`, od vseh preostalih vrednosti pa so nadalje uporabljeni le atributi `weightedAverage, close` in `open`. Prvi atribut je utežen s številom trgovanih enot pri določeni ceni, preostala atributa pa hranita vrednost o prvi in zadnji ceni v določenem 5-minutnem intervalu.

2.2 Novice

Ker so kriptovalute in novice o kriptovalutah precej nišno področje, je ponudnikov, ki bi indeksirali novice o dogodkih na tem področju malo. Težava je tudi v tem, da je malo ponudnikov, ki bi dostop do indeksiranih novic ponujali brezplačno. Kot najboljši kompromis so bile uporabljene objave na spletni strani `reddit.com`. Spletna stran ponuja obsežen API, ki je uporabljen za pridobivanje podatkov. Za vsako izmed trgovanih kriptovalut je bila

sestavljena poizvedba, ki vrača odgovore v formatu JSON. Poizvedba vrača rezultate objav, ki imajo v naslovu ime valute ali njeno kratico.

Odgovor na poizvedbo ima nekaj splošnih ključev kot so `before` in `after`, ki identificirata prejšnjo in naslednjo stran rezultatov. Pod ključem `children` so v seznamu shranjeni dejanski rezultati poizvedbe, ki nosijo informacije o avtorju, glasovih, datumu, povezavi na objavo ter ključ `is_self`, ki določa, ali je objava vezana na spletno stran `reddit.com` ali na zunanjo povezavo.

S pomočjo slednjega ključa je določeno, če je objava potencialna novica ali ne. V primeru, da je `is_self == False`, je prenesena koda HTML spletne strani s povezavo shranjena pod ključem `URL`. Ključne vrednosti, ki so potrebne za nadaljnjo uporabo novic, so naslov, besedilo in čas objave. Članki, pri katerih so te vrednosti najdene, so shranjeni v podatkovno bazo, preostali pa so zavrženi.

Iskanje in izločanje zgoraj omenjenih atributov je realizirano s Pythonovo knjižnico *newspaper3k*. Knjižnica ponuja ekstrakcijo naslovov, datumov objave, besedil in avtorjev na podlagi meta-podatkov, ki so prisotni v kodi HTML članka. Ponuja tudi analizo besedil in izločanje ključnih besed, ki pa niso bili uporabljeni v okviru te knjižnice.

Relevantni atributi za vsak članek shranjen v podatkovni bazi so:

- `_id`: generiran je unikaten ključ za vsak članek. Ključ ima format `valuta:datum:naslov`,
- `title`: naslov članka,
- `text`: besedilo članka,
- `date`: datum objave članka,
- `currency`: omenjena valuta v članku.

Kot je razloženo v poglavju 2.3, so pri učenju upoštevani le podatki z natančno uro objave. Ostali datumi imajo zaradi bolj ali manj poenotenega formata meta-podatkov na voljo tudi časovni pas objave (ki je skoraj v vseh

primerih v greenwiškem časovnem pasu), kar onemogoči učenje in napovedovanje z irelevantnimi podatki.

Večina člankov je v angleškem jeziku, vsa neangleška besedila, ki so bila uspešno prenesena, pa so bila izločena z ročnim urejanjem in klasifikacijo relevantnosti, opisanim v poglavju 2.3.1. Uporaba besedil v le enem jeziku odstrani potencialen šum, ki bi nastal zaradi majhnega števila neangleških besedil.

Vseh prenesenih in shranjenih člankov je 24096. Nekateri od teh so podvojeni, kar je odvisno od omemb več različnih valut v članku.

2.3 Napake v podatkih

Zaradi načina implementacije branja datumov iz kode HTML, je bil pri 39% vseh prenesenih člankov čas objave shranjen le s podatkom o dnevu, mesecu in letu objave. Do napake pride, ker knjižnica *newspaper3k* podatek o datumu najprej poskusi pridobiti iz povezave na novico; le-ta pa ponavadi vsebuje čas objave le do dneva natančno. Če povezava ne vsebuje datuma, je podatek pridobljen iz meta-podatkov za katere velja bolj konsistenten format in v praksi dosega visoko natančnost pravilnosti ekstrahiranja časa [4]. V primeru, da članek nima natančnega podatka o uri objave, le-ta ni uporabljen za učenje modela.

Nekateri preneseni podatki sicer imajo pravilno ekstrahiran datum, a napačno ekstrahirano besedilo članka. Pogosta napaka je obvestilo o onemogočenem prenašanju kode HTML izven spletnega brskalnika. Problem velikega števila člankov je tudi, da ne govorijo o kriptovalutah. To je posebej izrazito pri valutah Gamercoin (GAME), Pinkcoin (PINK) in Horizon (HZ), pri katerih je večina rezultatov povezanih s primarnimi pomeni imen oziroma kratic.

Poleg težav z besedili je nekaj napak tudi v pridobljenih informacijah o stanju na trgu. Osamljeni primeri shranjenih podatkov vsebujejo vrednosti enake 0, kar v praksi vsaj za ceno kriptovalute nikoli ni res. Težava izvira

iz podatkov shranjenih na borzi Poloniex. Pri gradnji matrik smo ignorirali primere s ceno valute enako 0, kar dodatno zmanjša končni nabor primerov.

2.3.1 Ročno urejanje in klasificiranje s pomočjo strojnega učenja

Da je mogoče doseči relevantne in konsistentne rezultate morajo vsa besedila ustrezati naslednjim kriterijem:

- besedilo članka mora biti pravilno preneseno,
- besedilo mora biti v angleškem jeziku,
- besedilo članka mora biti povezano s tematiko kriptovalut.

Večina besedil tem kriterijem ne ustreza, zato je bilo ročno urejenih 1000 člankov, ki ustrezajo zgoraj naštetim kriterijem. Ti so nato uporabljeni za učenje modela, ki determinira relevantnost vseh ostalih besedil. Napovedni model zgradimo na podlagi matrike pojavitev vseh besed. Za klasifikacijo relevantnosti je bila uporabljena metoda podpornih vektorjev s stohastičnim gradientnim sestopom, ki je implementirana v razredu `SGDClassifier` knjižnice *sklearn*. Besedilom je bil dodeljen razred *relevantno* in *irrelavantno* (oziroma 0 in 1).

Nad učno množico je bila dosežena 92% klasifikacijska točnost ob 69% večinskem razredu, kjer večinski razred predstavlja nepomembna besedila. Ob klasifikaciji je bil za vsak članek dodan atribut `relevant`, ki nosi logično vrednost glede na relevantnost besedila. S tem pristopom je bilo število relevantnih člankov zmanjšano na 6811. Po izločanju člankov, ki nimajo natančne informacije o času objave besedila, je za nadaljnjo obdelavo uporabnih 4221 besedil.

2.4 Dodatno čiščenje in obdelava besedil

Besedila so bila ob napovedovanju razdeljena na posamezne besede, nad katerimi je potekala nadaljnja obdelava:

- velike črke so bile spremenjene v male,
- odstranjeni so bili vsi znaki, ki ne spadajo v besede (ustrezajo regularnemu izrazu `[\W]+`).

Pred gradnjo matrik, iz katerih je naučen model za napovedovanje, so bile besede še lematizirane in krnjene. Za oba postopka je bila uporabljena Pythonova knjižnica *nltk* in sicer razreda `WordNetLemmatizer` in `PorterStemmer`. Vsaki krnjeni besedi je bila dodana tudi besedna vrsta, kar pri nadaljnjem učenju ločuje besede z enakim korenom, a različno vlogo v besedilih. Odstranjevanje nepomembnih oziroma blokiranih besed zaradi nadaljnje uporabe vrednosti *TF-IDF* filtriranja atributov ni bilo potrebno.

Ker članki pogosto govorijo o več kot eni kriptovaluti oziroma kriptovalute sploh niso osrednja tema besedila, je bilo izbranih n besed okoli omembe kriptovalute o kateri naj bi govorilo besedilo. Spremenljivka n je v našem primeru enaka 50, kar je glede na raziskavo [6] povprečna dolžina odstavka v spletnih besedilih. Prečiščen in formatiran tekst je v podatkovni bazi shranjen pod atributi `reduced_clean_text` in `reduced_clean_title`.

Poglavje 3

Metode

Ker je napovedovanje trendov osnovano na tekstovnih podatkih, so napovedi izvedene ob objavi vsakega novega besedila. Za to je potrebna izgradnja matrike, ki čim bolje povzame trenuten članek, članke objavljene v določenem časovnem oknu pred njim ter stanje na trgu pred objavo. V ta namen predlagamo naslednji nabor atributov, ki opisujejo vsebino besedil:

- **sentiment** je izračunan s pomočjo slovarja, ki ga ponuja SentiWordNet. Ta za vsako besedo v slovarju ponuja tri parametre: pozitivnost, negativnost in objektivnost. Za izračun sentimenta vsakega besedila je uporabljena enačba po [8]:

$$sentiment_{word} = score_{positive} - score_{negative} \tag{3.1}$$

$$sentiment_{text} = (\sum sentiment_{word})/n \tag{3.2}$$

  Sentiment besedila je povprečje sentimentov posameznih besed. Ker je sentiment na nivoju besede omejen na interval $[-1, 1]$, enako velja tudi za sentiment celotnega besedila. Vrednost -1 priča o zelo negativnem besedilu, vrednost 1 pa o zelo pozitivnem.

- **polariteta** je značilka, ki daje podatek o čustveni porazdelitvi besedila. Izračunana je kot standardni odklon sentimentov vseh besed; članki, ki vsebujejo veliko pozitivnih in negativnih besed imajo torej visoko vrednost polaritete in obratno. Vrednost je v intervalu $[0, 1]$.

- **verjetnostna porazdelitev tem besedila**: vsak članek ima izračunanih 100 atributov, kjer vsak določa verjetnost, da besedilo pripada posamezni temi oziroma skupini. Namen uporabe tem je diskriminacija člankov glede na kriptovalute in pomembne dogodke v podatkih. Vrednosti so omejene na interval $[0, 1]$. Uporabili smo implementacijo latentne Dirichletove alokacije knjižnice *lda* [10]. Ta sprejme pričakovano število tem ter vrne verjetnostne porazdelitve pripadnosti temam za vsako besedilo. Število tem je bilo izbrano z ročnim poskušanjem in iskanjem; manjše vrednosti so članke o različnih valutah grupirale skupaj, medtem ko večje vrednosti niso smiselno razdelile besedil. Prednost latentne Dirichletova alokacije je vračanje verjetnostnih porazdelitev za vse skupine (teme), kar da bolj podrobno informacijo in boljše možnosti primerjanja besedil med sabo v primerjavi z, npr., metodo voditeljev.

- **TF-IDF**: uporabljen je razred `TfidfVectorizer` knjižnice *sklearn*. V poglavju 3.2 nadalje predlagamo različne načine uteževanja.

- **ime valute v naslovu članka**: za bolj jasno določanje ali članek govori o specifični kriptovaluti ali je le-ta le omenjena, v naslovu članka preverimo pojavitev imena valute. Če je ime prisotno, atribut dobi vrednost 1, sicer pa dobi vrednost 0.

3.1 Modeliranje po časovni osi

Ker le eno besedilo redko odseva razmere na trgu in s tem posledično daje manjšo zmožnost napovedovanja trendov, so kot atributi uporabljene tudi povprečne vrednosti atributov besedil in razmere na trgu za izbrana časovna obdobja pred objavo besedila. Pri računanju teh atributov so upoštevana le besedila, ki govorijo o isti kriptovaluti kot obravnavano besedilo. Značilke, ki opisujejo vsebino besedil pred objavo trenutno obravnavanega članka, so:

- **distribucija preteklih besedil**: atribut določa časovno porazdelitev besedil. Izračunan je kot povprečje uteži, ki so uporabljene v nadalje-

vanju. Posamezne uteži imajo vrednost med 0 in 1, pri čemer največjo utež dobi besedilo, od objave katerega je preteklo najmanj časa. Višje povprečje uteži torej pove, da je do povečanja frekvence objav prišlo pred kratkim, nižje povprečje nakazuje na zmanjšanje frekvence objav. Vrednost distribucije $d = 0.5$ nakazuje na uniformno porazdelitev objav. Vrednosti distribucije so v intervalu $[0, 1]$.

- **utežen povprečen sentiment preteklih besedil**: povprečen sentiment vseh preteklih člankov, utežen s pretečenim časom od trenutno obravnavanega besedila. Besedila objavljena pred kratkim dobijo višjo težo, tista, objavljena na začetku izbranega časovnega okna, pa dobijo nižjo težo. Vrednosti povprečnih sentimentov so v intervalu $[-1, 1]$.

- **utežena povprečna polariteta preteklih besedil**: povprečna polariteta vseh preteklih člankov utežena s pretečenim časom od trenutno obravnavanega besedila. Besedila objavljena pred kratkim dobijo višjo težo, tista, objavljena na začetku izbranega časovnega okna, pa dobijo nižjo težo. Vrednosti polaritet so v intervalu $[0, 1]$.

Kot že omenjeno je vsaka utež izračunana na podlagi posameznih besedil oziroma njihovih časov objave, kar bolj podrobno pojasnjuje naslednja enačba:

$$weight_i = \frac{time_i - time_0}{time\_window} \tag{3.3}$$

kjer števec predstavlja razliko med časom objave besedila ($time_i$) in začetkom časovnega okna ($time_0$) v sekundah, imenovalec pa velikost časovnega okna.

Atributi tehnične analize so atributi, ki niso pridobljeni iz ali kakorkoli povezani s samimi besedili. To so numerični atributi, ki pričajo o stanju trga pred objavo obravnavanega besedila:

- **spreminjanje cene** kriptovalute o kateri govori besedilo. Ta je bila izračunana kot relativna sprememba med prvo in zadnjo ceno v časovnem oknu.

- **obseg trgovanja** (*angl. volume*) kriptovalute o kateri govori besedilo. Ta je izračunan kot delež trgovanih enot ene kriptovalute v razmerju s trgovanimi enotami celega trga:

$$volume_{time\_window} = \frac{total\_volume_{currency}}{total\_volume_{all\_currencies}} \tag{3.4}$$

  kjer *total_volume* označuje obseg trgovanja obravnavane valute *currency* oziroma obseg trgovanja celotnega trga (*all_currencies*) v času *time_window*. Izračunana vrednost je v intervalu $[0, 1]$.

- **relativne spremembe vrednosti celotnega trga**: podobno kot spremembe vrednosti posameznih valut so bile izračunane tudi spremembe celotnega trga. Ker imajo kriptovalute različne absolutne vrednosti, je sprememba celotnega trga izračunana kot povprečje relativnih sprememb posameznih valut v 5-minutnih (300-sekundnih) časovnih intervalih. Relativna sprememba vrednosti trga je nato za poljubno časovno okno izračunana z enačbo:

$$price\_change_{time\_window} = \prod_{i=0}^{n}(1 + price\_change_{300sec}) \tag{3.5}$$

  kjer je spremenljivka *time_window* izbrano časovno okno, n pa je enak $time\_window/300$.

Ob procesiranju vsakega članka smo izračunali zgoraj omenjene vrednosti atributov za več časovnih oken. To temelji na predpostavki, da lahko več oken nosi več informacij o kratkoročnih, srednjeročnih in dolgoročnih trendih pred objavo besedila. Število in velikost oken je izbrano na podlagi mere medsebojne informacije, kar je nadalje razloženo v poglavju 3.4. Namesto zgoraj naštetih šestih atributov je torej za vsak članek zgrajenih $6 * \check{s}tevilo\ oken$ atributov.

Zgradili smo tudi atribut, ki opisuje težo besedila v odvisnosti od frekvence objav člankov v preteklosti. Izračunan je po formuli $1/n$, kjer spremenljivka n predstavlja število objav v izbranem časovnem oknu. Teža besedila je omejena na interval $(0, 1]$. Atribut je uporabljen ob predpostavki,

da obstaja odvisnost med frekvenco besedil in njihovim vplivom; visoka frekvenca naj bi zmanjševala vpliv posameznih novic, nizka frekvenca pa naj bi ga povečala.

Preizkusili smo tudi utežene verjetnostne porazdelitve tem besedila. Te so bile utežene z enakimi utežmi kot, na primer, povprečen sentiment. Ker gre za povprečje distribucij, so te v enakem intervalu kot distribucije tem posameznih besedil.

V primeru teže besedila in povprečne distribucije tem smo uporabili le eno časovno okno, ki je bilo najdeno z naključnim iskanjem, opisanim v poglavju 3.4.

3.2 Uteževanje *TF-IDF*

Ob visoki frekvenci objavljenih člankov je pričakovano, da je vpliv posameznega besedila manjši. V ta namen predlagamo uteževanje besed v trenutno obravnavanem besedilu s povprečnimi vrednostmi *TF-IDF* preteklih besedil, sentimentom in pogostostjo besede v finančnih besedilih.

Povprečen *TF-IDF* preteklih besedil je utežen z enakimi utežmi kot atributi opisani v poglavju 3.1. Vrednosti so utežene z vrednostmi od 0 do 1, glede na pretečen čas do objave trenutno obravnavanega teksta.

Vsaka beseda v *TF-IDF* matriki trenutnega besedila je obtežena po naslednji enačbi:

$$tfidf_W = \frac{1 * tfidf_w}{n} + \frac{(n-1) * average\_tfidf_w}{n} \tag{3.6}$$

kjer je $tfidf_w$ *TF-IDF* vrednost obravnavane besede, n število vseh besedil zajetih v trenutnem časovnem oknu in $average\_tfidf_w$ uteženo povprečje preteklih besedil. Uteževanje *TF-IDF* je izvedeno le nad besedami, ki so v trenutno obravnavanem besedilu in je implementirano po enakem principu kot, npr., uteževanje porazdelitev tem. Časovno okno, ki smo ga uporabili je enako časovnemu oknu pri računanju teže besedil in porazdelitve tem in je določeno z naključnim iskanjem.

Vrednosti sentimenta posameznih besed so pridobljene iz slovarja SentiWordNet. Sentiment je izračunan po enačbi 3.1, ki vrne vrednosti med -1 in 1. V namen bolj intuitivnega uteževanja so sentimentu prištete vrednosti 1 oziroma -1, glede na:

$$sentiment_w = \begin{cases} sentiment_w - 1 & sentiment_w < 0 \\ sentiment_w + 1 & sentiment_w >= 0 \end{cases} \tag{3.7}$$

Pogosti finančni izrazi in njihovi atributi so bili preneseni iz spletne strani `www.academicvocabulary.info`. Uporabljenih je bilo 200 besed, ki so posebej pogosta v finančnih besedilih. Poleg drugih atributov je za vsako besedo pridobljeno tudi razmerje frekvence vsebovanosti v finančnih besedilih napram običajnim besedilom. Pridobljene vrednosti so normalizirane na interval od 0 do 1. Iz enakega razloga kot pri sentimentu, smo tudi tu prišteli 1.

Prej izračunano vrednost $tfidf_w$ vsake besede smo poskusili še nadalje utežiti z:

$$tfidf_w = tfidf_w * sentiment_w \tag{3.8}$$

$$tfidf_w = tfidf_w * financial\_weight_w \tag{3.9}$$

$$tfidf_w = tfidf_w * sentiment_w * financial\_weight_w \tag{3.10}$$

kjer spremenljivka $sentiment_w$ opisuje sentiment obravnavane besede, težo besede v finančnem slovarju pa opisuje spremenljivka $financial\_weight_w$.

3.3 Ciljni razredi

Za klasifikacijo smo izbrali tri ciljne razrede. Ti so *pozitivna sprememba*, *negativna sprememba* in *ni spremembe* oziroma njihove oznake 1, -1 in 0. O pripadnosti besedila nekemu razredu odloča naslednja funkcija:

$$Y(i) = \begin{cases} 0 & 0 < abs(price\_change_i) < threshold \\ 1 & price\_change_i >= threshold \\ -1 & price\_change_i <= -threshold \end{cases} \tag{3.11}$$

kjer je spremenljivka *threshold* določena na podlagi verjetnostne porazdelitve sprememb cen, $price\_change_i$ pa je sprememba cene po objavi i-tega besedila.

Podano je bilo časovno okno za katerega model napoveduje trend. Okno je bilo izbrano glede na najboljši rezultat naključnega iskanja. Pri izračunu ciljnega razreda sta bila obravnavana dva načina spremembe cene:

- sprememba cene je izražena kot relativna sprememba cene ob objavi besedila in zadnjo ceno v časovnem oknu. Če je sprememba cene pozitivna je besedilu dodeljen razred *pozitivna sprememba*, sicer pa je dodeljen razred *negativna sprememba*. V primeru trgovanja v praksi takšen opis trenda olajša odločitev o prodaji.

- sprememba cene je izražena kot največja relativna absolutna sprememba med prvo ceno in maksimumom oziroma minimumom v časovnem oknu. Če je absolutna negativna sprememba cene večja kot absolutna pozitivna sprememba, je besedilo klasificirano z *negativna sprememba*, sicer pa mu je dodeljen razred *pozitivna sprememba*. Ta tip klasifikacije poveča potencialen dobiček, a oteži izbiro primernega časa za prodajo.

Uporabljen je bil drugi pristop, saj bolj natančno odgovarja na vprašanje kaj se zgodi s ceno določene valute po objavi članka in je bolj neodvisen od velikosti okna. Način izračuna spremembe cene sicer ni bistveno vplival na klasifikacijsko točnost modela.

3.4 Izbiranje optimalnih hiperparametrov

Izbiro optimalnih hiperparametrov lahko razdelimo na tri stopnje. Prva stopnja zadeva parametre potrebne za gradnjo matrik, druga stopnja zadeva iskanje optimalnega števila atributov, v tretji stopnji pa so najdeni optimalni parametri metod za učenje.

V prvih dveh stopnjah je za učenje in napovedovanje uporabljena metoda podpornih vektorjev, v tretji fazi pa smo preizkusili tudi druge učne algoritme. Samo učenje in vrednotenje je podrobneje opisano v poglavju 3.5.

3.4.1 Izbira parametrov vhodnih podatkov

Iskali smo najboljše velikosti okna za napovedovanje, oken atributov preteklih stanj in pragov za določanje ciljnih razredov. Pri izbiranju parametrov smo se omejili na naslednje intervale:

- **velikost okna za napovedovanje**: od 5 minut do 6 ur
- **velikost oken za računanje atributov preteklih stanj** opisanih v poglavjih 3.1 in 3.2: od 5 minut do 48 ur.

Prag za določanje ciljnih razredov ni omejen na določen interval, pač pa je izbran iz statistične porazdelitve cen in je odvisen od velikosti izbranega napovednega okna. Za iskanje treh najboljših časovnih oken, ki so uporabljeni za izračun stanja na trgu in povprečnih sentimentov, polaritet in distribucij preteklih besedil smo uporabili mero medsebojne informacije. Prednost mere je, da omogoča izračun vrednosti tako za pozitivne kot tudi negativne vrednosti atributov. Pri računanju medsebojne informacije je bila uporabljena funkcija `mutual_info_classif` knjižnice *sklearn*. Ta kot vhodni parameter sprejme atribute za vsak učni primer v obliki matrike in ciljne razrede, ki tem primerom pripadajo. Mera izračuna odvisnost med vrednostmi atributa in ciljnimi razredi. Ker so ciljni razredi odvisni od velikosti napovednega časovnega okna in izbranega pragu se tudi medsebojna informacija spreminja glede na vrednost teh dveh parametrov.

V našem primeru so atributi vhodnih matrik za pridobitev vrednosti medsebojnih informacij izračunani za 500 različnih časovnih oken na intervalu od 5 minut do 48 ur. Vsako od časovnih oken je uporabljeno za računanje šestih atributov, ki so opredeljeni v poglavju 3.1. Vhodna matrika torej obsega 3000 atributov, ki so izračunani za vsak članek. Pri računanju mere medsebojne informacije je rezultat, ki ga dobimo, seznam vrednosti medsebojne informacije za vsak atribut. Ker nas zanima doprinos oken, so vrednosti atributov enako velikih oken povprečene. Izmed vrnjenih vrednosti so bila izbrana tista okna, ki imajo medsebojno informacijo v zgornjih $2,5\%$ vseh vrednosti.

Izmed vseh iskanih hiperparametrov tako ostaneta dve neodvisni spremenljivki. To sta napovedno časovno okno in časovno okno za izračun povprečnih vrednosti *TF-IDF*, povprečnih distribucij tem ter teže besedila. Optimalne vrednosti teh so bile poiskane z naključnim iskanjem v 120 iteracijah. Pri tem smo v vsaki iteraciji poiskali optimalne vrednosti ostalih parametrov kot je opisano v prejšnjih odstavkih. Naključno iskanje je bilo izbrano, ker je gradnja matrik z vsemi mogočimi kombinacijami zamudna, naključno iskanje pa v relativno malo poskusih pridobi reprezentativen vzorec končnih rezultatov. Število iteracij je bilo izbrano na podlagi zapisa v članku [11], ki z enačbo 3.12 utemeljuje, da že relativno malo poskusov naključnega iskanja z veliko gotovostjo da rezultate, ki so v deležu p vseh najboljših mogočih rezultatov:

$$1 - (1 - p)^n \geq 0.95 \tag{3.12}$$

V našem primeru lahko torej z vsaj 95% gotovostjo trdimo, da je za pridobitev vsaj enega rezultata med najboljšimi $p = 0.025$ rezultati pričakovano zadostno število poskusov $n = 120$. Optimalni parametri so bili izbrani na podlagi največje razlike med doseženo klasifikacijsko točnostjo in velikostjo večinskega razreda.

3.4.2 Izbira optimalnega števila atributov

Poleg določanja spremenljivk, ki so potrebne za gradnjo matrik, smo morali oceniti tudi optimalno število atributov, ki daje najboljše rezultate in kar se da zmanjša prekomerno prileganje (*angl. overfitting*). Število je bilo določeno na podlagi najboljšega rezultata 60 poskusov učenja z različnim številom leteh, kar dobro pokrije vsa mogoča števila atributov. Z enačbo 3.12 utemeljujemo, da smo s tem dobili klasifikacijsko točnost v zgornjih 5% vseh mogočih rezultatov.

Na vsakem koraku izbire atributov so bili preizkušeni trije tipi algoritmov za izbiro pomembnih značilk. Uporabljena je bila knjižnica *sklearn* in sicer razreda `SelectFromModel` in `SelectPercentile`. Razred `SelectFromModel` koristi pristope, ki so sicer del samostojnih algoritmov strojnega učenja in

imajo vgrajeno vrednotenje značilk. Med takšne algoritme spadata metoda naključnih gozdov in metoda podpornih vektorjev. `SelectPercentile` pa sprejme razrede, ki implementirajo različne mere vrednotenja prispevka značilk; to je, npr., mera medsebojne informacije in mera chi2. Atributi so evalvirani z naslednjimi metodami in njihovimi implementacijami v knjižnici *sklearn*:

- **metoda podpornih vektorjev**: implementacija `LinearSVC`,
- **metoda naključnih gozdov**: implementacija `RandomForestClassifier`,
- **mera medsebojne informacije**: implementacija `mutual_info_classif`.

V primeru uporabe metode `SelectFromModel` konstuktor razreda sprejme parameter *prag*, izbrani pa so atributi, katerih utež je večja od *prag* $*$ *povprečje uteži*. V primeru metode `SelectPercentile` pa je podana spremenljivka *percentil*, ki odloča o percentilu najboljših vrednosti, ki jih vrne izbrana mera. Z iterativnim spreminjanjem spremenljivk *prag* in *percentil* je torej izbrano optimalno število atributov.

3.4.3 Izbira hiperparametrov modelov

Kot zadnja optimizacija je bilo implementirano še naključno iskanje hiperparametrov modelov. Za vsakega izmed učnih algoritmov definiranih v poglavju 3.5 je bil definiran nabor parametrov, ki so bili preizkušeni v 60 iteracijah. V vsakem poskusu smo zaradi naključnih inicializacij nekaterih modelov v namen večje robustnosti rezultatov 5-krat pognali učenje z enakimi parametri, najboljši parametri vsakega algoritma pa so bili izbrani na podlagi povprečne klasifikacijske točnosti. Podobno kot v prejšnjih poglavjih utemeljujemo, da 60 poskusov zadošča, da dobimo klasifikacijsko točnost izmed najboljših 5% vseh mogočih rezultatov.

3.5 Učenje in vrednotenje

Tako učenje kot tudi vrednotenje je potekalo z razredi in funkcijami Pythonove knjižnice *sklearn*. Za učenje je bila izbrana metoda podpornih vektorjev (implementacija `LinearSVC`), ki je po [3] najpogosteje uporabljena metoda za učenje in klasifikacijo tekstovnih podatkov. V knjigi [5] je pokazano, da metoda na podatkih zbirke Reuters na različnih primerih klasifikacije besedil v primerjavi z naivnim Bayesom in metodo k-najbližjih sosedov daje najboljše rezultate. Poleg metode podpornih vektorjev so bili sicer preizkušeni še trije učni algoritmi:

- **metoda k-najbližjih sosedov**: implementacija `KNeighboursClassifier`,
- **nevronska mreža**: implementacija `MLPClassifier`,
- **metoda naključnih gozdov**: implementacija `RandomForestClassifier`.

Končni rezultati pa so bili ovrednoteni z merami:

- **klasifikacijska točnost**: uporabljena je funkcija `accuracy_score`,
- **priklic**: uporabljena je funkcija `recall_score`,
- **natančnost**: uporabljena je funkcija `precision_score`,
- **površina pod krivuljo ROC**: uporabljena je funkcija `roc_auc_score`.

Ker gre za trirazredni problem, so priklic, natančnost in površina pod krivuljo ROC izračunani za vsak razred posebej. Izbran pristop k učenju in napovedovanju kar se da posnema učenje in napovedovanje, ki bi bilo uporabljeno v primeru avtomatskega trgovanja v praksi. Tako v primeru preizkušanja parametrov kot tudi v primeru končne evalvacije modela učenje poteka do dneva objave besedila za katerega napovedujemo trend. Število učenj modela je torej enako številu različnih dni objav člankov v validacijski oziroma testni množici. Ta v primeru končne evalvacije modelov predstavlja 20% vseh

podatkov. V primeru preizkušanja parametrov je 80% preostalih podatkov nadalje razdeljenih v razmerju 80 : 20, katere večji del je namenjen za učenje, preostanek pa za evalvacijo.

Poglavje 4

Rezultati

Poglavje opisuje rezultate dosežene ob uporabi metod opisanih v poglavju 3. Po pregledu značilnosti vhodnih podatkov sledijo rezultati iskanja hiperparametrov, vrednotenje prispevka skupin značilk in končni rezultati na validacijski in testni množici.

4.1 Značilnosti vhodnih podatkov

Kot že omenjeno so podatki razdeljeni na učno, validacijsko in testno množico. Testna množica predstavlja 20% vseh podatkov, preostalih 80% podatkov pa je nadalje razdeljenih na učno in validacijsko množico v razmerju $8 : 2$. Vseh zgrajenih atributov za vsak članek je 77143, katerih razdelitev podrobneje opisuje tabela 4.1. Končno število primerov, ki so na voljo za učenje in klasifikacijo je 3977. Testna množica obsega 795 primerov, učna in validacijska pa po 2546 in 636 primerov. Testna množica zajema podatke 63 dni, validacijska pa 44 dni. Glede na to, da model učimo s podatki do dneva objave besedila, to pomeni, da je v primeru preizkušanja parametrov učenje izvedeno 44-krat, v primeru evalvacije pa 63-krat. Validacijsko množico uporabljamo za preizkušanje hiperparametrov in atributov, testna množica pa je uporabljena le za končno evalvacijo modela.

Vidimo, da večino atributov predstavljajo vrednosti *TF-IDF*. V prilogi B

tip značilk	število
značilke trenutnega besedila	6
značilke preteklih besedil	12
značilke preteklih stanj na trgu	12
vrednosti TF-IDF	76913
značilke distribucije tem	100
značilke distribucije tem člankov pred objavo	100

Tabela 4.1: Število zgrajenih atributov

je podanih 50 najpomembnejših atributov pridobljenih z metriko medsebojne informacije. Kljub temu, da je atributov vrednosti *TF-IDF* večina, je teh med najpomembnejšimi atributi precej manj od ostalih značilk, kot so podatki o stanju na trgu pred objavo besedila, teme in pretekli sentimenti in polaritete.

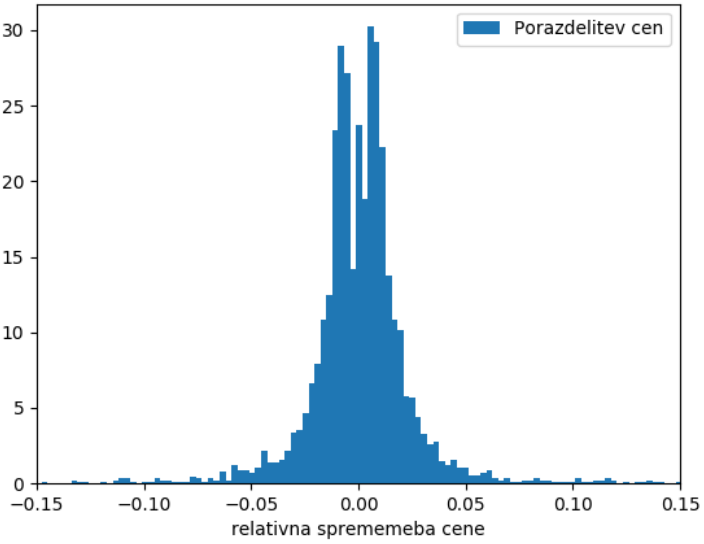
4.1.1 Ciljni razredi

Osnova za določanje ciljnih razredov je porazdelitev relativnih sprememb cen po objavi besedila. Te so porazdeljene kot prikazuje slika 4.1. Vidno je, da že ob enournem časovnem oknu povprečje pozitivnih in negativnih sprememb konvergira stran od skupnega povprečja, kar je pričakovano, saj so spremembe sorazmerno s pretečenim časom ponavadi vse večje.

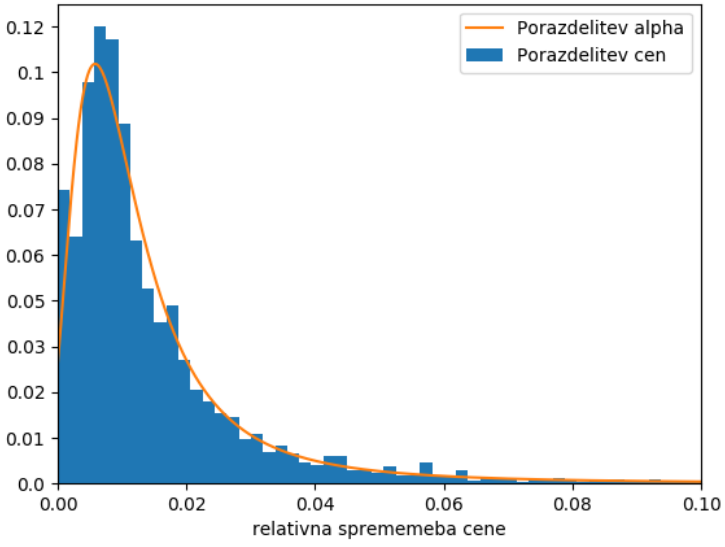
Ugotovili smo, da se porazdelitev sprememb cen najbolje prilagaja porazdelitvi alpha kot prikazuje slika 4.2. Za iskanje prilagoditev so bile uporabljene absolutne vrednosti sprememb, saj je le-te lažje aproksimirati z znanimi statističnimi porazdelitvami.

4.2 Aproksimacija optimalnih parametrov

Pri aproksimaciji so rezultati doseženi z načinom uteževanja in uporabe atributov glede na rezultate opisane v poglavju 4.3.



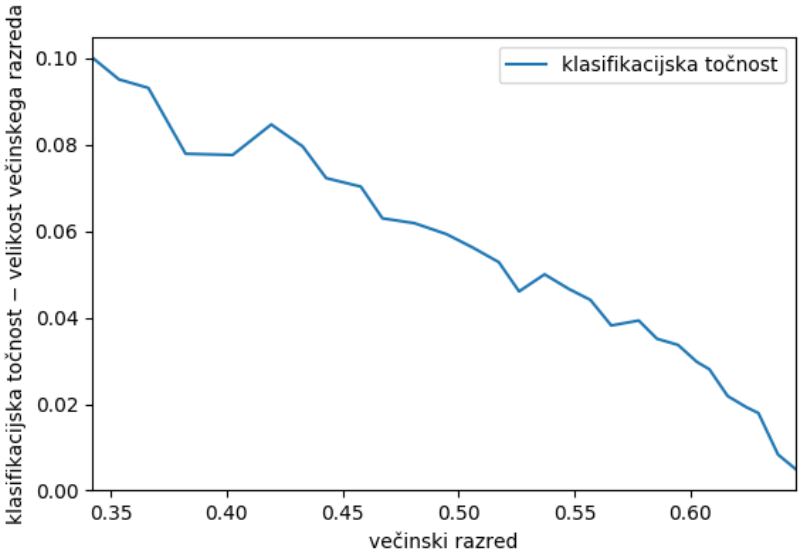
Slika 4.1: Porazdelitev cen ob enournem časovnem oknu.



Slika 4.2: Porazdelitev absolutne spremembe cen ob enournem časovnem oknu ter porazdelitev alpha.

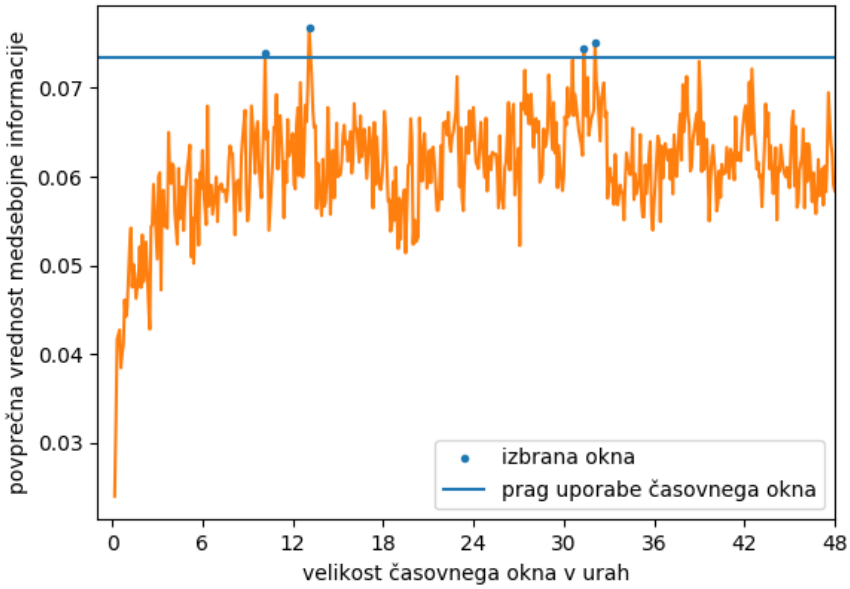
4.2.1 Določanje parametrov podatkov

Prag za določanje ciljnih razredov člankov je izračunan na podlagi porazdelitve prikazane na sliki 4.2. Kot prikazuje slika 4.3 se model nauči največ, ko so razredi kar se da uniformno porazdeljeni. Iz tega sledi, da je optimalni prag tisti, ki podatke razdeli na tretjine, kar je mogoče določiti z zbirno funkcijo verjetnosti porazdelitve alpha. Spremembe manjše od praga padejo v razred *ni spremembe*, medtem ko je ostalim glede na predznak dodeljen razred *pozitivna sprememba* ali *negativna sprememba*. Glede na to, da so spremembe cen porazdeljene simetrično, je v vsakem razredu približno enako število primerov. Trditev, da so porazdelitve podatkov simetrične, izhaja iz hipoteze naključnega sprehoda, ki pravi, da imajo tako pozitivne kot negativne spremembe vrednosti verjetnost 50 : 50, kar je opisano v članku [9].



Slika 4.3: Odvisnost klasifikacijske točnosti od velikosti ciljnega razreda.

Časovna okna uporabljena pri izračunu sentimentov, polaritet ter distribucij in stanja trga pred objavo besedila so bila vrednotena z uporabo mere medsebojne informacije. Rezultate vrednotenja prikazuje slika 4.4. Povprečja medsebojne informacije atributov časovnih oken velikih do 6 ur so re-



Slika 4.4: Primer določanja časovnih oken s povprečno mero medsebojne informacije.

lativno neodvisna od ciljnih razredov. To je pričakovano, saj manjša časovna okna nosijo malo informacije o stanju pred objavo besedila. Atributi časovnih oken, večjih od šestih ur pa imajo precej podobne vrednosti medsebojne informacije. Kljub temu je mogoče najti ekstreme, za katere v našem primeru velja, da so med $2,5\%$ najboljših rezultatov kot je to prikazano na sliki 4.4. Graf vrednosti medsebojne informacije se sicer spreminja glede na izbrano napovedno časovno okno, a ima v vseh primerih konsistentne lastnosti. Edina izjema je končno izbrano število oken, ki variira od 1 do 8.

Tabela 4.2 prikazuje končni nabor parametrov matrike. Vidimo, da je okno preteklih porazdelitev tem in povprečnih vrednosti *TF-IDF* skladno s prej omenjeno hipotezo, da večja okna nosijo več informacij o preteklih stanjih. Prag za določanje ciljnih razredov je enak 0.0146, kar pomeni, da je za dodelitev razreda *pozitivna sprememba* ali *negativna sprememba* potrebna vsaj $1,46\%$ sprememba cene v podanem napovednem časovnem oknu. Večinski razred učne množice ob podanem pragu je *ni spremembe*, ki je velik

parameter	**vrednost**
okno preteklih porazdelitev tem in *TF-IDF*	2665 min
okna preteklih besedil in stanj na trgu	985, 1195, 1245, 1260 min
napovedno časovno okno	315 min
prag določanja ciljnih razredov	0.0146

Tabela 4.2: Optimalni hiperparametri.

$34,8\%$. Razreda *pozitivna sprememba* in *negativna sprememba* sta velika $34,3\%$ in $30,9\%$. V tej fazi je bila dosežena najboljša klasifikacijska točnost $50,8\%$.

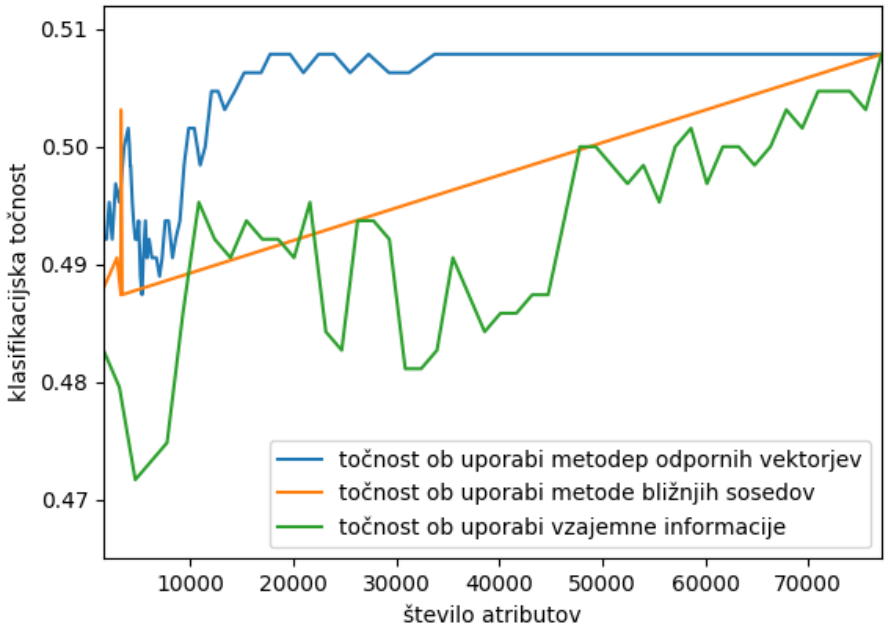
4.2.2 Določanje optimalnega števila atributov

Druga faza iskanja hiperparametrov zadeva število atributov modela, kjer smo za gradnjo učne matrike uporabili parametre iz tabele 4.2. V tabeli 4.3 vidimo, da filtriranje atributov neglede na uporabljeno metodo ne prispeva k izboljšanju rezultata. Hkrati je zanimivo, da uporaba le 4% vseh atributov pri filtriranju z metodo naključnih gozdov da skoraj enake rezultate kot uporaba vseh atributov. Razlog za to je verjetno v velikem številu nepomembnih vrednosti *TF-IDF* kot so, npr., vezniki in predlogi. Optimalno število atributov glede na izbiro metode znatno variira, razlog za kar je morda v majhni odvisnosti števila značilk in klasifikacijske točnosti. Kot prikazuje slika 4.5 je vpliv števila atributov na uspešnost učenja največ 4%.

Kjub temu, da filtriranje atributov ne izboljša že prej doseženih rezultatov, je bilo le-to vseeno uporabljeno v namen prekomirnega prileganja. Uporabili smo metodo podpornih vektorjev, ki je od vseh preizkušenih metod najhitrejša ter nudi dober kompromis med potencialnim pretiranim prileganjem pri testiranju na testni ter klasifikacijsko točnostjo na validacijski množici.

metoda filtriranja	št. atributov	prag/percentil	CA
metoda podpornih vektorjev	17727	1,188	50,8%
metoda naključnih gozdov	3017	1,5	50,6%
mera medsebojne informacije	70972	0,08	50,4%

Tabela 4.3: Optimalno število atributov, pragi in klasifikacijske točnosti glede na tip uporabljenega algoritma.



Slika 4.5: Klasifikacijska točnost v odvisnosti od števila atributov.

4.2.3 Določanje parametrov modelov

V tretji fazi so bili poiskani še optimalni hiperparametri učnih algoritmov. Kot že omenjeno smo za učenje in napovedovanje uporabili metodo podpornih vektorjev, preizkusiti pa smo želeli tudi nekaj drugih metod. Za učenje smo uporabili že prej ugotovljene hiperparametre podatkov ter optimalno število atributov. V naslednjih odstavkih je za vsak preizkušen tip učnega

algoritma naveden nabor parametrov, katerih vrednosti smo spreminjali, ter najboljše klasifikacijske točnosti. Vsi nenavedeni parametri so enaki privzetim kot je navedeno v dokumentacijah algoritmov knjižnice *sklearn*.

Metoda podpornih vektorjev. S funkcijo napake $loss = squared\_hinge$, kaznijo $penalty = l2$ in parametrom $C = 1.0$ je bila dosežena klasifikacijska točnost $50,8\%$. Vsi parametri so enaki privzetim parametrom razreda `LinearSVC()`.

Metoda naključnih gozdov. S številom dreves $n\_estimators = 18$, najmanjšim številom primerov v listih $min\_samples\_leaf = 16$ in načinom brez reduciranja števila atributov ($max\_features = None$) je bila dosežena klasifikacijska točnost $42,6\%$.

Nevronska mreža. Z uporabo stohastičnega gradientnega sestopa $solver = sgd$, funkcijo aktivacije $activation = relu$ in konstantno hitrostjo učenja $learning\_rate = constant$ je bila dosežena najboljša klasifikacijska točnost $41,2\%$.

Metoda k-najbližjih sosedov. Uporaba algoritma ga grupiranje $algorithm = ball\_tree$, velikosti listov $leaf\_size = 37$, številom sosedov $n\_neighbors = 9$ in manhatnske razdalje ($p = 1$) je bila dosežena $46,4\%$ klasifikacijska točnost.

Metoda naključnih gozdov, nevronske mreže in metoda k-najbližjih sosedov pričakovano niso izboljšali rezultatov. Vse prej ugotovljene parametre smo namreč preizkušali z uporabo metode podpornih vektorjev, s čimer smo jih prilagodili za dosego najboljših rezultatov s slednjim algoritmom. Za nadaljevanje učenja in napovedovanja smo torej še naprej uporabljali privzete hiperparametre te metode.

4.3 Vrednotenje prispevka značilk

V poglavju 3 predlagamo uporabo različnih atributov z različnimi načini uteževanja, ki so ovrednoteni v tem poglavju. Dosežene klasifikacijske točnosti v tabelah so rezultat uporabe parametrov in metod, ki so v prejšnjem po-

glavju dajali najboljše rezultate.

4.3.1 Teme

Kot je bilo ugotovljeno že na začetku poglavja atributi tem spadajo med pomembnejše atribute. Na seznamu 50 najpomembnejših atributov v prilogi B polovica vseh pripada temam. Po izvajanju filtriranja med atributi ostane 79 izmed 200 atributov tem pri čemer so atributi povprečnih porazdelitev pred objavo članka v povprečju bolj pomembni.

uporaba značilk tem	**CA**
brez uporabe tem	47,5%
porazdelitev tem trenutnega besedila	47,0%
povprečna porazdelitev tem preteklih besedil	48,9%
porazdelitev tem trenutnega besedila in povprečna porazdelitev tem preteklih besedil	50,8%

Tabela 4.4: Rezultati v primerih različne uporabe tem

4.3.2 Vrednosti *TF-IDF*

Kot je predlagano v poglavju 3.2 smo vrednosti *TF-IDF* uteževali s sentimentom, pogostimi finančnimi izrazi in vrednostmi *TF-IDF* preteklih besedil. Rezultati, prikazani v tabeli 4.5 kažejo, da predlagano uteževanje s sentimenti in finančnimi izrazi ne izboljša rezultatov. To je v nasprotju z rezultati članka [8], kjer je ugotovljeno, da uporaba enakega uteževanja kot je uporabljeno v tem diplomskem delu, klasifikacijsko točnost izboljša za skoraj 3%. Zanimivo je, da je uporaba uteži pogostih finančnih besed brez dodatnega uteževanja izboljšala klasifikacijsko točnost, ko pa so finančne uteži uporabljene skupaj z uteženim povprečjem *TF-IDF*, pa ne doprinesejo k boljšemu rezultatu. Od predlaganih načinov uteževanja je smiselna nadaljnja uporaba le uteženega povprečja *TF-IDF*.

tip uteževanja	**CA**
brez uporabe *TF-IDF* vrednosti	41,2%
neuteženi *TF-IDF*	44,7%
sentiment	44,3%
pogosti finančni izrazi	45,8%
sentiment in pogosti finančni izrazi	44,5%
uteženo povprečje *TF-IDF*	50,8%
uteženo povprečje *TF-IDF* in sentiment	50,2%
uteženo povprečje *TF-IDF* in pogosti finančni izrazi	50,6%
uteženo povprečje *TF-IDF*, sentiment in pogosti finančni izrazi	49,8%

Tabela 4.5: Klasifikacijske točnosti ob različnih načinih uteževanja *TF-IDF*.

4.3.3 Značilke tehnične analize

Značilke tehnične analize so med atributi z najvišjo vrednostjo medsebojne informacije na seznamu atributov v prilogi B. Največ informacije nosijo atributi, ki uporabljajo večja časovna okna, kar smo pokazali že v poglavju 4.2. V tabeli 4.6 vidimo, da ima največji doprinos h klasifikacijski točnosti sprememba obsega trgovanja, spremembe cen pa imajo relativno majhen vpliv.

uporaba značilk tehnične analize	**CA**
brez značilk tehnične analize	48,2%
spremembe cene kriptovalute	48,8%
spremembe cen vseh valut	48,9%
sprememebe obsega trgovanja	50,7%
vse značilke tehnične analize	50,8%

Tabela 4.6: Rezultati prispevkov značilk tehnične analize.

4.4 Končni rezultati

Končni najboljši rezultat na validacijski množici je $50,8\%$ ob $34,8\%$ večinskem razredu. Natančnosti napovedovanja za razreda *pozitivna sprememba* in *negativna sprememba* je pod 50%, glej tabelo 4.7. Model se pri napovedovanju trendov večinoma moti. Površine pod krivuljami ROC so v primerih napovedovanja pozitivnih in negativnih sprememb blizu $0,5$, kar dodatno kaže na slabe zmožnosti napovedovanja modela.

razred	**natančnost**	**priklic**	**AUC**	**% primerov**
neg. sprememba	49,2%	37,8%	0,576	30,9%
ni spremembe	58,7%	54,2%	0,71	34,8%
poz. sprememba	48,1%	60,9%	0,594	34,3%

Tabela 4.7: Rezultati na testni množici za vsak ciljni razred.

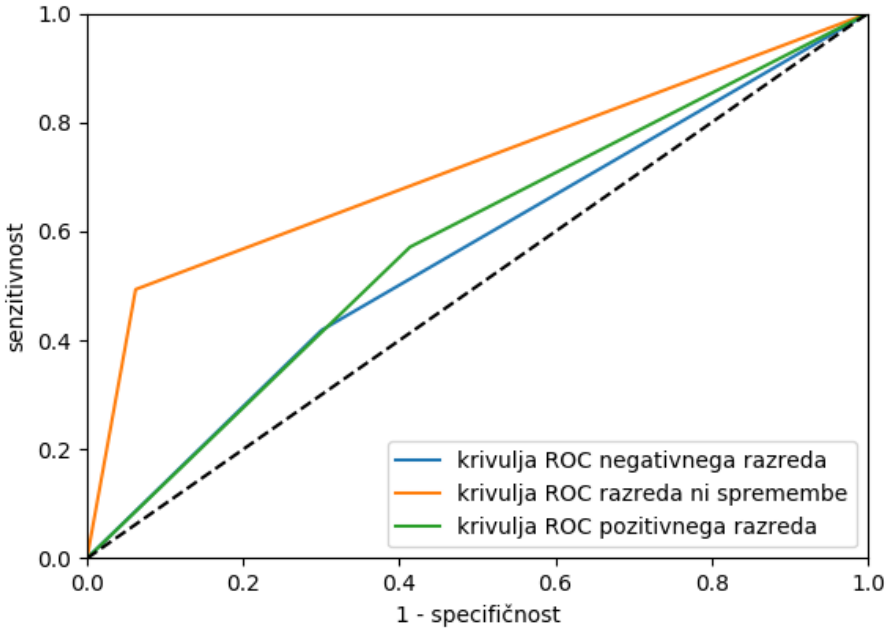
Končna najboljša dosežena klasifikacijska točnost na testni množici je $49,3\%$ ob $36,4\%$ večinskem razredu. Podrobnejši rezultati so prikazani v tabeli 4.8, ki kažejo na podobne rezultate kot pri vrednotenju na validacijski množici s poslabšanjem pri razredih *pozitivna sprememba* in *negativna sprememba*. Ob visoki natančnosti vidimo, da je končni model na novih podatkih dober predvsem pri napovedovanju primerov, po katerih ne pride do spremembe v ceni večje od $1,46\%$ (kar določa prag iz poglavja 4.2). Ciljni razredi so v primerjavi z razredi validacijskih podatkov bolj neenakomerno porazdeljeni.

razred	**natančnost**	**priklic**	**AUC**	**% primerov**
neg. sprememba	44,3%	41,9%	0,559	36,4%
ni spremembe	76,6%	49,3%	0,716	29,3%
poz. sprememba	41,9%	57,1%	0,579	34,3%

Tabela 4.8: Rezultati na validacijski množici za vsak ciljni razred.

Slika 4.6 dodatno prikazuje, da model dobro deluje le za napovedovanje razreda *ni spremembe*. V primeru ostalih dveh razredov krivulja ROC kaže

majhno izboljšanje napovedovanja napram naključnemu ugibanju.



Slika 4.6: Krivulje ROC napovedovanja vseh treh ciljnih razredov.

Ob primerjavi klasifikacijske točnosti z ostalimi podobnimi raziskavami lahko pridemo do zaključka, da diplomsko delo daje povprečne rezultate. Pri obravnavanju trirazrednega problema so v [2] z uporabo metode k-najbližjih sosedov in sofisticiranih pristopov tekstovnega rudarjenja dosegli največ 53% klasifikacijsko točnost. Hipoteza, predstavljena v [1] pravi, da sporočila internetnih forumov sicer slabo napovedujejo pozitivne trende, omogočajo pa napovedovanje volatilnosti v prihodnosti. V našem primeru je napovedovanje trenda najboljše za razred *ni spremembe*, iz česar bi lahko sledil podoben zaključek.

Model, zaradi uspešnosti napovedovanja razredov *pozitivna* in *negativna sprememba* pod 50%, v praksi verjetno ne bi dajal dobičkonosnih napovedi. Ne le, da je dosežena nezadostna natančnost, pri trgovanju za vsako transakcijo namreč nastajajo dodatni stroški, ki še zmanjšajo možnost zaslužka.

Poglavje 5

Sklepne ugotovitve

V diplomskem delu smo z uporabo tekstovnega rudarjenja in numeričnih podatkov trgov naučili model, ki tako na validacijski kot testni množici daje vzpodbudno klasifikacijsko točnost pri napovedovanju trendov vrednosti kriptovalut. V primerjavi z večinskim klasifikatorjem smo napovedovanje izboljšali za $12,9\%$. Dosežen rezultat se lahko primerja z rezultati nekaterih drugih raziskav, ki smo jih zasledili. Kljub temu pa je bilo ugotovljeno tudi, da model v praksi ne bi dajal dobičkonosnih napovedi zaradi relativno slabih natančnosti napovedovanja pozitivnih in negativnih trendov.

Opravljeno delo torej ne premika meje najboljšega doseženega rezultata na področju trgovanja v odzivu na novice. Ena izmed omejitev modela je verjetno preprostost zgrajenih učnih podatkov, ki ne vsebujejo sofisticiranih atributov tehnične analize. Za dosego boljših rezultatov pri napovedovanju pozitivnega ali negativnega trenda bi lahko več pozornosti posvetili določanju tem besedil ter iskanju omenjenih entitet ter njihovih relacij. Problem morda predstavlja tudi vir tekstovnih podatkov, ki je morda pristranski zaradi neprofesionalne narave uporabnikov spletne strani Reddit. Smiselna bi bila torej neposredna uporaba specializiranih virov novic. Poleg izboljšav pri generiranju učnih podatkov bi bilo pametno poglobljeno preizkusiti tudi druge učne algoritme, ki smo se jih v diplomskem delu zgolj dotaknili.

Kljub nezmožnosti napovedovanja samih trendov pa model nakazuje po-

tencial za napovedovanje volatilnosti. Napovedovanje razreda, ki opisuje stanje brez sprememb, je namreč relativno boljše od preostalih dveh. Zanimivo bi bilo preizkusiti delovanje modela na dvorazrednem problemu, pri katerem bi poskušali napovedovati značilne absolutne spremembe cen. Podatek o tem je lahko uporabljen kot eden izmed indikatorjev v večjem sistemu za podporo odločanju.

Literatura

[1] Werner Antweiler and Murray Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294, 2004.

[2] S. Leung D. Permunetilleke K. Sankaran J. Zhang B. Wuthrich, V. Cho. Daily stock market forecast from textual web data. Technical report, Hong Kong Univ. of Sci. and Technol., Clear Water Bay, Hong Kong, 1998.

[3] Arman Khadjeh Nassirtoussi et al. Text mining for market prediction: A systematic review. Technical report, Department of Information Science, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia, 2014.

[4] Ran Geva. Article's publication date extractor – an overview. Dosegljivo: `http://blog.webhose.io/2015/12/13/articles-publication-date-extractor-an-overview/`. [Dostopano: 28. 8. 2017].

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. Simple word problems in universal algebras. In *Introduction to Information Retrieval*, pages 279–286. Cambridge University Press, 2008.

[6] Nirmaldasan. Plain paragraph length. Dosegljivo: `https://strainindex.wordpress.com/2010/10/25/plain-paragraph-length/`. [Dostopano: 17. 7. 2017].

[7] Not just bitcoin: The top 7 cryptocurrencies all gained in 2016. Dosegljivo: http://www.coindesk.com/not-just-bitcoin-the-top-7-cryptocurrencies-all-gained-in-2016/. [Dostopano: 3. 7. 2017].

[8] Jiajia Li Phayung Meesad. Stock trend prediction relying on text mining and sentiment analysis with tweets. Technical report, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, 2004.

[9] Random walk theory. Dosegljivo: http://www.investopedia.com/terms/r/randomwalktheory.asp. [Dostopano: 3. 9. 2017].

[10] Topic modeling with latent dirichlet allocation. Dosegljivo: https://pypi.python.org/pypi/lda. [Dostopano: 3. 9. 2017].

[11] Alice Zheng. Evaluating machine learning models. Dosegljivo: https://www.oreilly.com/ideas/evaluating-machine-learning-models/page/5/hyperparameter-tuning. [Dostopano: 20. 8. 2017].

Dodatek A

Obravnavane kriptovalute

Seznam kriptovalut borze Poloniex za obdobje od 1.11.2015 do 31.10.2016:

ime kriptovalute	**oznaka**
Synereo AMP	AMP
Ardor	ARDR
Boolberry	BBR
Bytecoin	BCN
BitCrystals	BCY
Belacoin	BELA
Bitstar	BITS
BlackCoin	BLK
Bitcoin	BTC
BitcoinDark	BTCD
Bitmark	BTM
BitShares	BTS
Burst	BURST
Coin2.0	C2
CLAMS	CLAM
Curecoin	CURE
Dash	DASH
Decred	DCR

DigiByte	DGB
Dogecoin	DOGE
Einsteinium	EMC2
Ethereum Classic	ETC
Ethereum	ETH
Expanse	EXP
Factom	FCT
FoldingCoin	FLDC
Florincoin	FLO
GameCredits	GAME
Gridcoin Research	GRC
Huntercoin	HUC
Horizon	HZ
IO Digital Currency	IOC
LBRY Credits	LBC
Lisk	LSK
Litecoin	LTC
MaidSafeCoin	MAID
Myriadcoin	MYR
Nautiluscoin	NAUT
NAVCoin	NAV
Neoscoin	NEOS
Namecoin	NMC
NobleCoin	NOBL
DNotes	NOTE
NuShares	NSR
NXT	NXT
Omni	OMNI
Pinkcoin	PINK
PotCoin	POT
Peercoin	PPC

Qibuck	QBK
Qora	QORA
Quatloo	QTL
Radium	RADS
Rubycoin	RBY
Augur	REP
Riecoin	RIC
Steem Dollars	SBD
Siacoin	SC
Shadow	SDC
Storjcoin X	SJCX
STEEM	STEEM
Stellar	STR
Syscoin	SYS
SuperNET	UNITY
Viacoin	VIA
Voxels	VOX
VeriCoin	VRC
Vertcoin	VTC
BitcoinPlus	XBC
Counterparty	XCP
NEM	XEM
Magi	XMG
Monero	XMR
Pebblecoin	XPM
Ripple	XRP
Vcash	XVC
Zcash	ZEC

Dodatek B

Pomembni atributi

Seznam 50 najpomembnejših atributov in njihovih vrednosti medsebojne informacije, pridobljenih iz značilk člankov:

ime atributa	**medsebojna informacija**
price_74700	0.1873
volume_74700	0.1812
volume_71700	0.1752
volume_75600	0.1728
price_59100	0.1718
price_75600	0.1670
price_71700	0.1656
volume_59100	0.1475
bitcoin.n	0.0976
average_topic_34	0.0797
average_topic_89	0.0658
average_topic_69	0.0641
w	0.0633
average_topic_1	0.0591
average_topic_32	0.0583
ethereum.n	0.0557
average_topic_36	0.0550

average_topic_21	0.0526
average_topic_43	0.0512
average_topic_50	0.0508
average_topic_38	0.0508
average_topic_82	0.0490
average_topic_8	0.0489
average_topic_84	0.0481
average_topic_64	0.0478
average_topic_22	0.0469
polarity_74700	0.0462
average_topic_12	0.0458
average_topic_66	0.0456
average_topic_51	0.0454
average_topic_18	0.0450
average_topic_83	0.0450
average_topic_40	0.0449
polarity_75600	0.0449
utcthus.n	0.0446
sentiment_75600	0.0443
coinoffering.n	0.0440
average_topic_28	0.0423
hadnt.n	0.0423
readmemd.a	0.0422
average_topic_77	0.0420
matthee.v	0.0420
service.a	0.0420
requiredplayers.n	0.0419
mackert.n	0.0418
average_topic_75	0.0416
endanger.v	0.0415
average_topic_3	0.0411

average_topic_19	0.0409
average_topic_72	0.0402