Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tilen Miklavič

Analiza vpliva omemb kriptovalut na njihovo ceno

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Slavko Žitnik

Ljubljana, 2019

Copyright. Rezultati diplomske naloge so intelektualna lastnina avtorja in matične fakultete Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, fakultete ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Kandidat: Tilen Miklavič
Naslov: Analiza vpliva omemb kriptovalut na njihovo ceno
Vrsta naloge: npr. Diplomska naloga na univerzitetnem programu prve stopnje Računalništvo in informatika
Mentor: doc. dr. Slavko Žitnik

Opis:
Besedilo teme diplomskega dela študent prepiše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode naj uporabi, morda bo zapisal tudi ključno literaturo.

Title: Public sentiment to cryptocurrency price influence analysis

Description:
opis diplome v angleščini

Za pomoč pri izdelavi diplomske naloge bi se rad zahvalil mnogim prijateljem in znancem, ki so mi s svojim znanjem znali svetovati pri premagovanju ovir.

Kazalo

Povzetek

Abstract

1 Uvod **1**

2 Motiv **3**

3 Pregled literature iz tega področja **5**

3.1 Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis 5

3.2 The predictive power of public Twitter sentiment for forecasting cryptocurrency prices 6

3.3 Coin Market Cap 7

3.4 SatoshiStreetBets 7

4 Prelged uporabljenih tehnologij **9**

4.1 HTML 9

4.2 CSS 9

4.3 JavaScript 10

4.4 BootStrap 10

4.5 NodeJS 11

4.6 RaspberryPi 11

4.7 NoIP.com 11

4.8 Python 12

4.9 Webscraping 12
4.10 Google Firebase 12

5 Podrobnost implementacije 15
5.1 Zbiranje podatkov 15
5.2 Normalizacija podatkov 16
5.3 Prikaz podatkov 17
5.4 Iskanje podobnosti 17
5.5 Pridobivanje aktualnih podatkov 25
5.6 Strežnik 26
5.7 Spletna stran 26

6 Delovanje aplikacije 29

7 Rezultati 35
7.1 XRP 35
7.2 BTC 37

8 Zaključek 39

Literatura 41

Seznam uporabljenih kratic

kratica	angleško	slovensko
CA	classification accuracy	klasifikacijska točnost
DBMS	database management system	sistem za upravljanje podatkovnih baz
SVM	support vector machine	metoda podpornih vektorjev

Povzetek

Naslov: Analiza vpliva omemb kriptovalut na njihovo ceno

Avtor: Tilen Miklavič

V preteklem letu se je izkazalo, da ima javni sentiment na družabnih včasih velik vpliv na določene kriptovalute. Ta podatek bi lahko entuziasti s pridom izkoristili pri svojih analizah. V svoji nalogi bom zbral in analiziral podatke za več kriptovalut in jih razvrstil v seznam po odvisnosti od javnega sentimenta. To mi bo pomagalo pri izdelavi napovednega modela, ki bo podajal obvestila o abnormalnih dogodkih na socialnih omrežjih. Uspešen rezultata naloge bo napovedni model, ki bo pravilno in časovno natančno klasificiral dogajanje na socialnem omrežju. V sodelovanju s spletno aplikacijo bi lahko tako uporabnika v realnem času o teh dogodkih obveščal.

Ključne besede: kriptovaluta, kovanec, socialna omrežja.

Abstract

Title: Public sentiment to cryptocurrency price influence analysis

Author: Tilen Miklavič

In the last year alone, we have learned that public sentiment on social networks poses different influence on cryptocurrencies. Enthusiasts could be using this new information in their analysis. In my work, I will collect and analyze data for different cryptocurrencies and order then on the list based on their ability to be influenced. This will help me with developing a prediction model algorithm, which will notify every time it senses abnormal behaviour on social network. Successful result would be prediction model, which correctly and accurately predicts events on social network. Working with web application, it would notify user in real time.

Keywords: cryptocurrency, coin, social network.

Poglavje 1

Uvod

V svoji diplomski nalogi bom natančneje pogledal, kako na različne kripto valute vpliva javni sentiment investitorjev na družabnih omrežjih. Različni kroptokovanci so drugače podvrženi flukcuaciji cene zaradi količine in sentimenta zapisov na socialnih omrežjih (npr. Twitter). V okviru naloge bom s pomočjo različnih tehnik analiziral grafa količine zapisov na družabnem omrežju Twitter o določeni kriptovaluti in cene te kriptovalute skozi čas. Nato bom poskusil poiskati kakšne povezave med dvema omenjenima količinama in razvrstiti kovance na seznam kovancev bolj podvrženim vplivom sentimenta z družbenih omrežij. Ugotovitve iz tega dela mi bodo pomagale, da ustvarim napovedni model, ki bo nato v realnem časo analiziral dnevne količine zapisov na Twitterju in jih primerjal z že zbranimi zvorci obnašanja v preteklosti. V primeru, da bo opazil podobnosti bo dogodek klasificiral kot kazalec za možno abnormalno dogajanje na trgu. Podatke iz tako zbranih kazalnikov bom potem naprej uporabil pri izdelavi spletne aplikacije za obveščanje uporabnika o teh dogodkih. S pomočjo te aplikacije bo lahko končni uporabnik vedno obveščen, če bo na Twitterju prišlo do dogodkov, ki bi lahko nakazovali, da bo cena kriptovalute poskočila ali pa močno padla. Ta kazalec bi bil lahko potencialnemu investitorju še dodatna informacija, na podlagi katere bi se lahko odločil za nakup ali prodajo neke določene kriptovalute. Po uspešni izdelavi programa za napoved bi tega seveda še stestiral na

način, kjer bi vsako opozorilo naprej klasificiral glede na uspešnost opozorila na dogodek.

Poglavje 2

Motiv

Tema pregleda socialnih omrežij in sentimenta ljudi pri nakupovanju delnic in kriptovalut verjetno ni mlada, ampak je svojo popularnost seveda pridobila predvsem v preteklem letu z odvijanjem dogajanj okol socialnega omrežja Reddit in delnic AMC in GME. Dogajanje okoli množičnega nakupovanja delnic in kriptovalut se tam tudi ni končala, saj smo kmalu za tem bili priča množičnemu nakupovanju kriptovalute Dogecoin (in podobnih), katere je ponovno vodila močna skupnost uporabnikov Reddita. Na kriptovalute bolj vpliva samo povpraševanje na trgu kot na delnice, zato me je prešinila misel, da bi lahko na tem področju zagotovo pridobil nekakšne numerične podatke, iz katerih bi lahko špekuliral na rast ali padec kriptovalute. Moja začetna teza je bila, da so nekatere kriptovalute bolj dovzetne za flukcuacijo cene kot druge, velik faktor mora biti seveda tržna kapitalizacijo določene kriptovalute. Za vse nadobudne programerje bi bil nek seznam kriptovalut, najbolj dovzetnih za manipulacijo na družabnih omrežjih še kako dobrodošel, saj bi lahko s takšno kategorizacijo pravilno ovrednostili določena obnašanja na socialnih omrežjih. Vendar je sam seznam teh kropto kovancev dosti neuporaben, če moramo ročno spremljati dogajanje na Twitterju. Zato bi bilo v veliko ponoč napisati odločevalni algoritem, ki bi lahko klasificiaral dogajanje na Twitterju za določene kriptovalute (najvišje s prej omenjenega seznama) in nato zabeleženi dogodek primerjal s predhodnimi dogodki, ki so že po-

kazali, da so potencialni kazalci za spremembe na trgu. Takšen algoritem, skupaj s spletno aplikacijo za pregled in obveščanje končnih uporabnikov glede stanja bi se lahko v tej niši izkazal za zelo uporabnega.

Poglavje 3

Pregled literature iz tega področja

V tem poglavju bom raziskal in opisal literaturo, spletne strani in aplikacije, ki trenutno opravljajo isto ali podobno nalogo, kot sem si jo zastavil sam. Cilj raziskovanja področja je ugotoviti količino zanimanja za tovrstne rešitve in zbiranje informacij in pogledov na različne načine pristopa k reševanju problema napovedovanja gibanja trga kriptovalut.

3.1 Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis

V tej raziskavi [7] so avtorji analizirali količino omemb kriptovalut Bitcoin in Ethereum na Twitterju in popularnost glede na spletno orodje Google Trends. Zanimalo jih je, ali na nihanje cene teh dveh kriptovalut bolj vpliva količina, ali sentiment zapisov na Twitterju. Ugotovili so, da ima Twitter vedno večjo moč pri odločitvah velikih skupin ljudi za nakup ene izmed teh dveh kriptovalut. Analiza pa je ravno tako pokazala, da zanašanje na sentiment zapisov na Twitterju, ni najbolj zanesljiva metoda, saj je le ta pogosteje pozitiven (tudi v primeru, ko se cena kriptovalute spušča). Ker sta Bitcoin in Ethereum kriptovaluti z zelo visoko tržno kapitalizacijo, se natančna na-

poved lahko izkaže za zelo pomembno informacijo tudi pri nakupu drugih kriptovalut. Pristopa do napovedovanja premika cene so se lotili z uporabo linearnegam modela, na katerega so vnesli podatke o količini Twitter zapisov in ceni posamezne kriptovalute. Rezultati so se izkazali za uspešne, saj so v večini primerov premik cene pravilno napovedali.

3.2 The predictive power of public Twitter sentiment for forecasting cryptocurrency prices

V naslednji raziskavi [8] avtorja ubereta malce drugačen pristop do analize vpliva omemb na Twitterju na ceno določenih kriptovalut. V svoj pregled sta vključila devet največjih kriptovalut, pri katerih sta se zanašala predvsem na analizo javnega sentimenta z uporabo leksikografske analize. S tem pristopom sta ugotovila, da se s pomočjo analize javnega sentimenta lahko z veliko mero gotovosti napove gibanje kriptovalut Bitcoin, Bitcoin Cash in Litecoin. Malce drugačen pristop, z uporabo bikovega / medvedjega razmerja (bull / bear ratio), pa se lahko napoveduje kriptovaluti EOS in TRON. Bull / bear ratio je razmerje, ki je določeno s pomočjo "anketiranja"večjih finančnih investitorjev. Kot kriptovaluta, katere cena reflektira samo količino Twitter zapisov se je izkazala kriptovaluta Ripple, ravno tako pa količina Twitter zapisov (in javni sentiment) vpliva tudi na kriptovaluto Litecoin. Tekom študije, sta avtorja prišla še do ene zanimive ugotovitve. Okoli 14% vseh Twitter zapisov je generirano z računov, ki jih upravljajo avtomatski programi. To pomeni, da obstaja velika verjetnost, da določeni uporabniki že iskoriščajo znanje o javnem sentimentu in količini omemb kriptovalut na Twitterju kot vpliv na nihanje cene.

3.3 Coin Market Cap

Spletna stran [5] CoinMarketCap je eno glavnih orodij investitorjev in špekulantov. Od svoje ustanovitve leta 2013 je spletna stran že močno zrasla in jo sedaj obiskuje tudi več kot 2.000.000 uporabnikov dnevno. Stran vsebuje hitro posodobljene informacije o cenah, gibanju, tržnem deležu, novicah o kriptovalutah. Odprli pa so tudi opcijo za nakup kriptovalut kar preko njihove strani. Zaradi enostavnosti uporabe in visoke uporabnosti ostaja ena izmed najbolj popularnih in najbolj obiskanih spletnih strani povezanih s kriptovalutami.

3.4 SatoshiStreetBets

Družabno omrežje je če skoraj desetletje znano po gostovanju sobe za špekulante o delniškem trgu WallStreet Bets [4]. Ta soba je močno pridobila na popularnosti letos, po zelo znanem incidentu okoli delnic Gamestop (GME) in AMC Entertainment Holdings, kjer so člani sobe z masovnimi nakupi uspeli ceno delnic močno dvigniti. Pri tem so marsikateri investitorji, ki so stavili proti tem podjetjem izgubili visoke vsote svojih vložkov. Soba Satoshi Street Bets [6] je bila ustvarjena lanskega leta z namenom kopiranja špekulacij iz sobe Wall Street Bets in prenašanjem teh na trg kriptovalut. Pri tem so bili relativno uspešni, saj jim je s množičnim nakupovanjem uspelo dvigniti cene različnih kriptovalut, kot so Doge Coin, Shiba inu, SafeMoon in druge. Soba je tako postala glavno zbirališče špekulantnih investitorjev v majhne in visoko likvidne kripto kovance, ki si po večini želijo hitrih visokih zaslužkov.

Poglavje 4

Prelged uporabljenih tehnologij

V tem poglavju bom opisal uporabljene tehnologije v okviru moje diplomske naloge.

4.1 HTML

HTML (Hyper-Text Markup language) je označevalni jezik s pomočjo katerega izdelujemo spletne strani. Uporablja značke za definicijo delov spletne strani, tem značkam pa lahko predpisujemo tudi atribute. HTML uporablja nabor različnih značk, večina teh vedno nastopa v paru (začetna in končna značka). Prva je označena z znakoma < in > z imenom značke med njima. Končni znački pa se doda še poševnico, katera simbolizira, da je to končna značka in ne še ena začetna značka. Hypertext oznaka se nanaša na povezovanje znotraj in zunaj določene spletne strani, kar omogočajo povezave v HTML.

4.2 CSS

CSS (Cascading Style Sheets) je slogovni jezik, ki nam pomaga definirati izgled HTML strani. Z uporabo selektorjev izberemo en sam element na strani (ali pa nabor teh), ter definiramo, kako naj izgledajo. CSS lahko tudi

skrbi za prilagajanje vsebine spletne strani na različne dimenzije in velikosti zaslonov naprav, na katerih lahko uporabniki dstopajo do spletne strani. V današnji dobi interneta, si skoraj ne moremo predstavljati spletne strani, ki nebi na nek način uporabljala CSS za definiranje izgleda. Izlged in odzivnost spletnih strani je postal že tako velik del izdelave, da sedaj obstajajo različne knjižnice in orodja, ki nam pomagajo laže in z manj kode definirati CSS stile.

4.3 JavaScript

JavaScript je programski jezik, ki ga uporabljamo za programiranje obnašanja spletnih strani. Narast popularnosti ga je prinesel do točke, da ga sedaj imenujemo "programski jezik spleta". Z JavaScript definiramo obnašanje spletne strani pri pritiskih na gumbe, izpolnjevanje vnosnih polj, premikanje po strani, spreminjanje strani in podobno. Vedno bolj pa je uporabljen za komunikacijo spletne strani z ostalim spletom. Z uporaob tega jezika navadno obvladujemo prijavo in avtentikacijo uporanbikov, poiščemo in vključimo podatke z drugih spletnih strani v našo spletno stran, komuniciramo z API-ji in strežnikom. Javascript je vedno bolj prisoten na spletni in mobilnih aplikacijah, saj je glavni gradnik v povezovanju in komuniciranju različnih delov spleta med seboj.

4.4 BootStrap

BootStrap je CSS in JavaScript knjižnica, ki nam pomaga definirati izlged in obnašanje spletne strani z bistveno manj vrsticami kode. Za razvijalce, ki oblikovanja niso vešči, CSS pogosto povzroča velike preglavice, saj samo definiranje stilov in zamikanje elementov po spletni strani ni vedno preprosto. Bootstrap pomaga rešiti ta problem s ponudbo vnaprej definiranih stilov, zamikov in obnašanja spletne strani, ki jih razvijalec lahko vključi v svojo spletno stran samo z dodajanjem novega razreda v HTML značko.

4.5 NodeJS

NodeJS je odprtokodno izjavalno okolje, ki izjava Javascript kodo izven brskalnika. Narejeno je bilo z namenom omogočiti razvijalcem s poznavanjem jezika Javascript sprogramirati obnašanje strežnika, brez učenja dodatnega programskega jezika. S tem je bil narejen velik korak k sprejetju Javascript kot jezik svetovnega spleta. NodeJS sem uporabil kot izvajalno okolje na strežniku, v njem pa sem definiral krajišče, s klicem na katerega lahko moja spletna stran pridobi trenutne podatke o opazovanih kriptokovancih.

4.6 RaspberryPi

RaspberryPi je mikroračunalnik, zgrajen na eni plošči. Uporabil sem ga, da igra vlogo strežnika, na katerem se izvaja NodeJS okolje, kamor lahko spletna stran izvede klic za pridobitev aktualnih informacij o kriptokovancih. Na RaspberryPi računalniku je nameščen operacijski sistem Raspberry Pi OS, ki je bil napisan na Linux osnovi.

4.7 NoIP.com

NoIP je spletna storitev, ki poenostavi gostovanje spletnih strani in strežnikov v domačem okolju. Ko gostujemo spletno stran, potrebujemo statičen IP naslov, če hočemo do nje dostopati tudi izven domačega omrežja. NoIP z namestitvijo klienta na strećniku omogoča povezavo spletnega naslova do dimaničnega IP naslova strežnika in s tem nudi spletno izkušnjo kot pri gostovanju spletne strani na strežniku s statičnim IP naslovom. V sklopu svoje diplomske naloge sem storitev NoIP uporabljal za gostovanje svojega strežnika na statičnem IP naslovu. Ker se moja spletna stran gostuje na spletni storitvi Firebase [3], za gostovanje te nisem imel potrebe, da pa se program za pridobivanje podatkov o Twitter zapisih in spletni strežnik izvajata na isti napravi, sem tudi tega pognal na RaspberryPi-ju. Računalnik je povezan v lokalno omrežje v mojem stanovanju, zato se tudi njegov IP naslov

konstantno spreminja. NoIP poskrbi, da lahko preko enega url naslova vedno dostopam s spletne strani do svojega strežnika.

4.8 Python

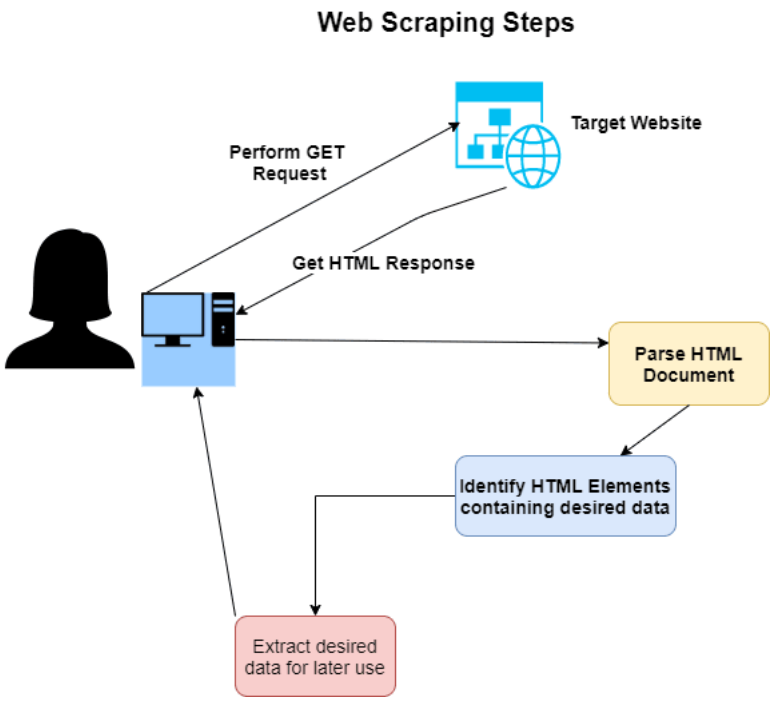
Python je programski jezik, katerega sem v splopu diplomske naloge uporabil za pridobivanje podatkov o količini Twitter zapisov za določen kriptokovanec v vsakem dnevu. RaspberryPi promovira uporabo programskega jezika Python, zato je bila ta odločitev skoraj samoumevna. Z uporabo gonilnika Google Web Driver in Python knjižnice Selenium sem lahko avtomatiziral postopek ekstrakcije potrebnih podatkov z internetne strani CoinDesk [2] in [1] in jih zapisal v datoteko, katero je potem strežnik v NodeJS okolju lahko tudi prebral ob izvedenem klicu.

4.9 Webscraping

Webscraping je postopek pridobivanja podatkov s spletnih strani, ki ne ponujajo API-ja, preko katega bi lahko do teh podatkov dostopal. Pristop k izdelavi takšnega programa je relativno preprost. S spletnega strežnika pridobimo HTML pogled spletne strani, nato pa z uporabo postopkov za lociranje elementov na spletni strani izluščimo želene podatke.

4.10 Google Firebase

Google Firebase je spletna platforma, ki razvijalcem ponuja zbirko orodij za razvoj spletnih, mobilnih in nativnih aplikacij. V okviru diplomske naloge sem uporabil njihovo možnost gostovanja spletne strani.



Slika 4.1: Ekstrakcija podatkov s spletne strani

Poglavje 5

Podrobnost implementacije

5.1 Zbiranje podatkov

Na začetku naloge sem zbral zgodovinske podatke za število Twitter zapisov o določenih kriptovalutah in cene teh kriptovalut. Število Twitter zapisov sem črpal iz spletne strani BitInfo Charts [1], podatke o ceni pa iz CoinDesk [2]. Spletna stran CoinDesk ima zelo intuitiven uporabniški umesnik za prenos zgodovinskih podatkov o ceni, medtem ko BitInfo Charts tega nima. Iz slednje strani sem moral podatke, ki so bili uporabljeni za izris grafa na spletni strani, ročno kopirati iz HTML datoteke v novo, JSON datoteko.

Primer JSON datoteke, ki vsebuje podatke o ceni za kriptokovanec ADA.

```
{
  "prices":
  [
    {
      "Currency": "ADA",
      "date": "2018-06-01",
      "price": 0.2189733642,
      "24h Open (USD)": 0.2241552905,
      "24h High (USD)": 0.2249912905,
      "24h Low (USD)": 0.2167457132
    },
    {
```

```
      "Currency": "ADA",
      "date": "2018-06-02",
      "price": 0.2191795566,
      "24h Open (USD)": 0.2232069963,
      "24h High (USD)": 0.2279878336,
      "24h Low (USD)": 0.2138870034
    },
```

Primer JSON datoteke, ki vsebuje podatke o številu Twitter zapisov o kriptokovancu ADA.

```
{
  "tweets":[
    {"date":"2018/01/08","tweet":"780"},
    {"date":"2018/01/09","tweet":"958"},
    {"date":"2018/01/10","tweet":"845"},
    {"date":"2018/01/11","tweet":"1051"},
    {"date":"2018/01/12","tweet":"1497"},
    {"date":"2018/01/13","tweet":"1444"},
    {"date":"2018/01/14","tweet":"2683"},
    {"date":"2018/01/15","tweet":"1680"},
    {"date":"2018/01/16","tweet":"1381"},
    {"date":"2018/01/17","tweet":"1249"},
    {"date":"2018/01/18","tweet":"1403"},
```

Postopek sem tako ponovil za vsak izbran kriptokovanec.

5.2 Normalizacija podatkov

Preden sem lahko določil povezavo med številom zapisov na Twitterju za neko kriptovaluto in njeno ceno, sem moral podatke normalizirati. Na ta način sem se izognil velikemu razkoraku med številom zapisov in ceno pri določenih kriptovalutah. Absolutna razlika med tema dvema količinama me namreč ne zanima, saj sem iskal povezavo v gibanju količin in ne nujno same magnitude gibanja.

Za normalizacijo podatkov sem pripravil program, napisan v programskem jeziku Python; `normalizer.py`. Program se najprej sprehodi čez ce-

loten niz podatkov v seznamu in določi lokalno maksimalno in minimalno vrednost. Tako lahko pri drugem obhodu že ustvari nov seznam podatkov, pri katerem je vsaka nova vrednost izračunana po formuli:

$$X' = \frac{X - Xmin}{Xmax - Xmin}$$

Kot rezultat takšne normalizacije dobimo seznam podatkov med 0 in 1, ki jih lahko tako dosti lažje med seboj primerjamo. Vsi trendi rasti in padanja so še vedno vidni, lažje pa podatke tudi predstavimo na grafu.

Generirane podatke sem potem izvozil v novo JSON datoteko, ter jo ustrezno poimenoval. Tako nastale datoteke bodo uporabljene za analizo povezanosti in prikaz na grafu.

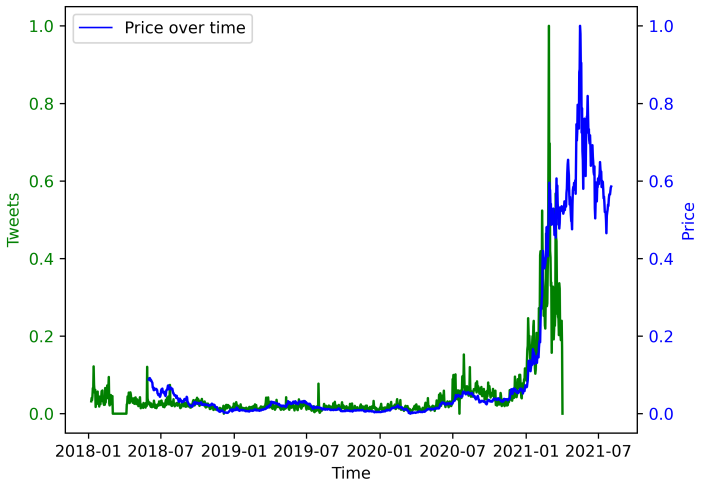
5.3 Prikaz podatkov

Normalizirane podatke si lahko za predstavo tudi prikažemo na grafu. V primeru, da podatkov nebi normalizirali, bi se lahko zgodilo, da bi bila ena količina (cena) dosti nižja kot druga (Twitter omembe) saj sta velikokrat v popolnoma različnih redovih velikosti.

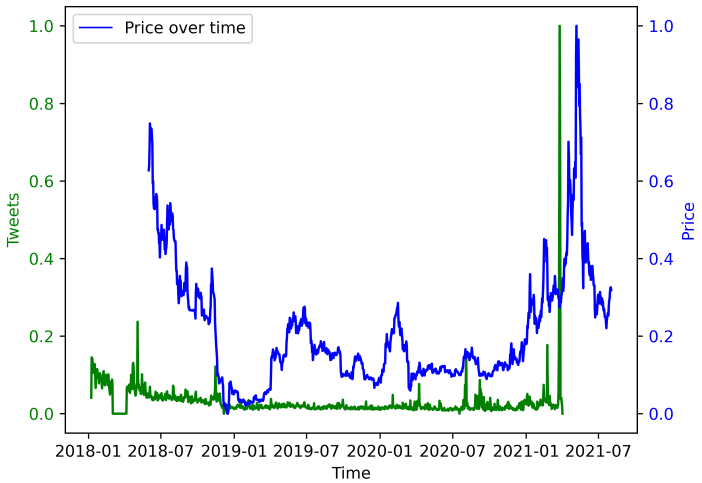
Že samo lahko opazimo, kako sta si krivulji za ceno in količino Twitter zapisov na nekaterih grafih podobni. Kot primer vzemimo Bitcoin. To je kriptovaluta z največjo tržno kaitalizacijo, ravno tako pa dnevno prejme največ Twitter zapisov. Ni presenetljivo, da krivulja, ki predstavlja ceno Bitcoin-a, tako tesno sledi krivulji, ki predstavlja število zapisov na Twitterju. Pri drugi največji kriptovaluti; Ethereum, lahko opazimo podobno obnašanje.

5.4 Iskanje podobnosti

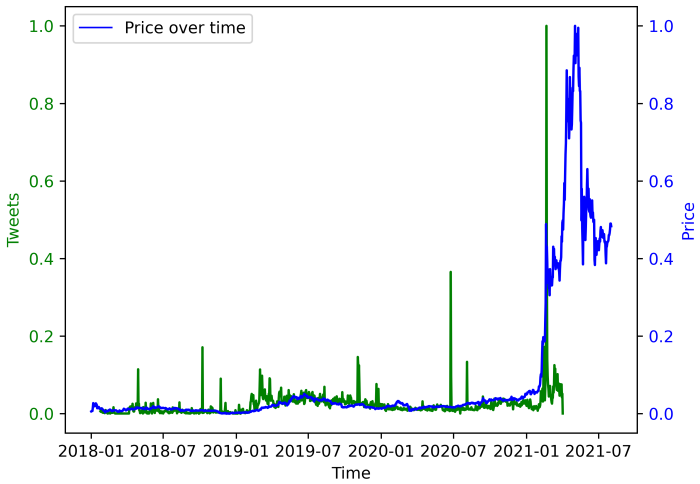
Svoje analize seveda nisem mogel prepustiti samo opazovanju grafov, zato sem napisal program v Pythonu; `similarity.py`, s pomočjo katerega sem poskusil preveriti dejansko podobnost med krivuljama na grafu. Ker sem



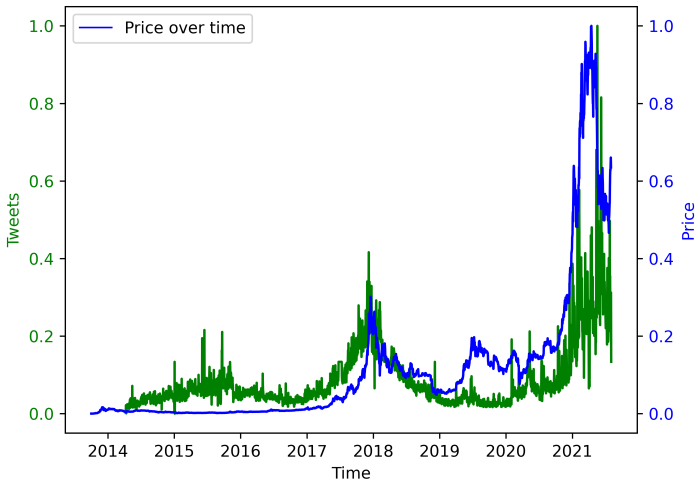
Slika 5.1: ADA - tweets / price



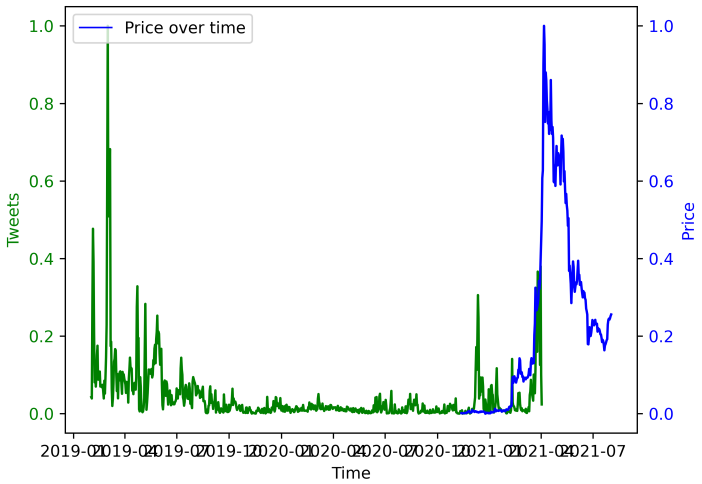
Slika 5.2: BCH - tweets / price



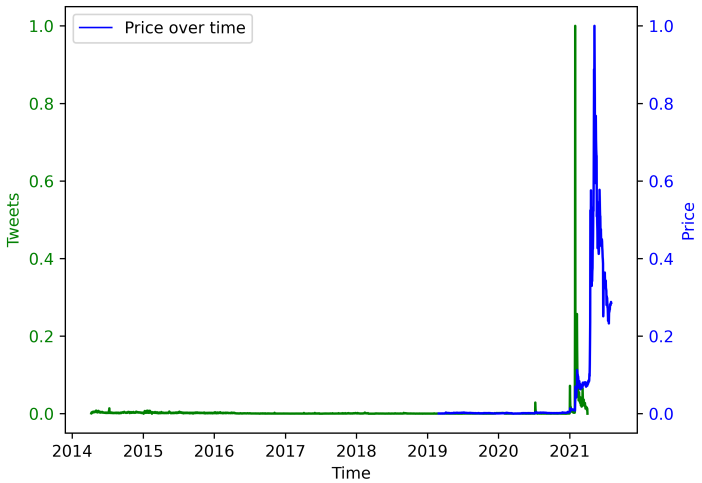
Slika 5.3: BNB - tweets / price



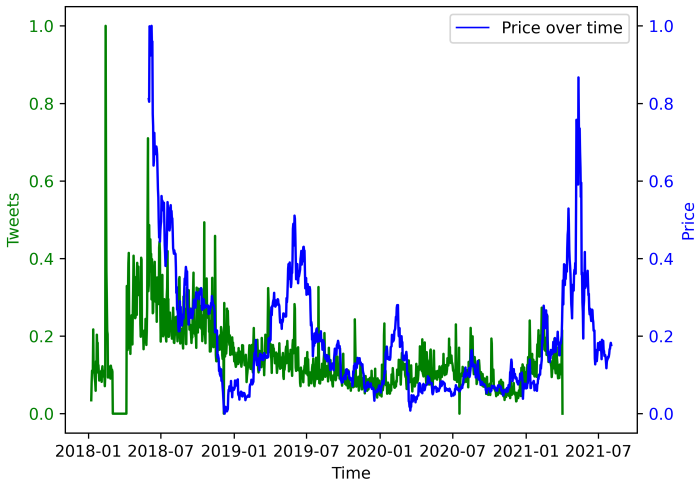
Slika 5.4: BTC - tweets / price



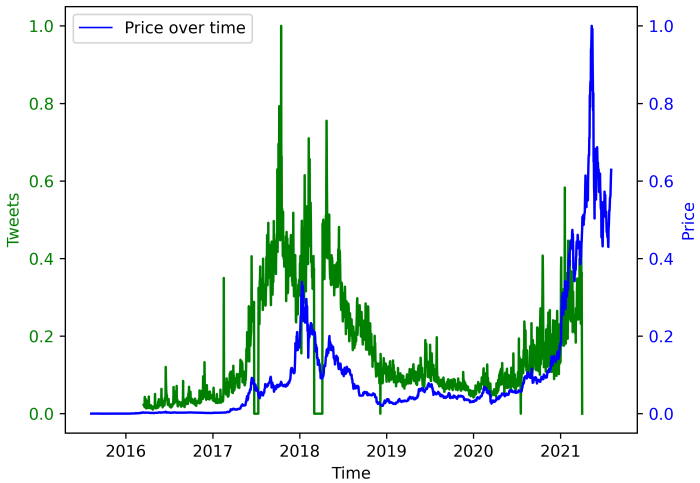
Slika 5.5: BTT - tweets / price



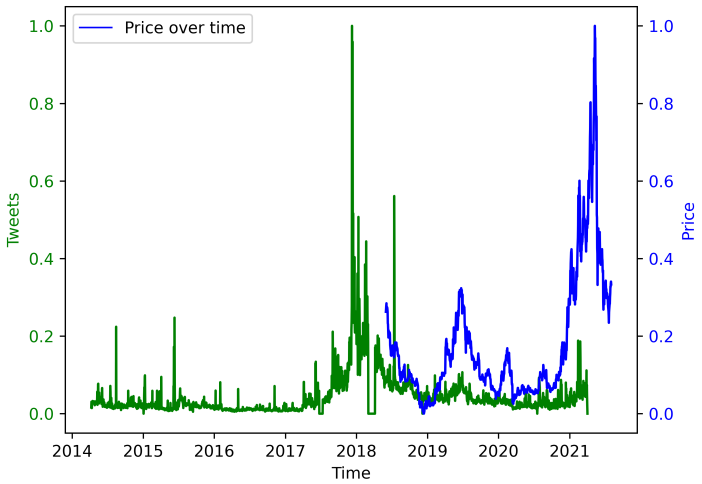
Slika 5.6: DOGE - tweets / price



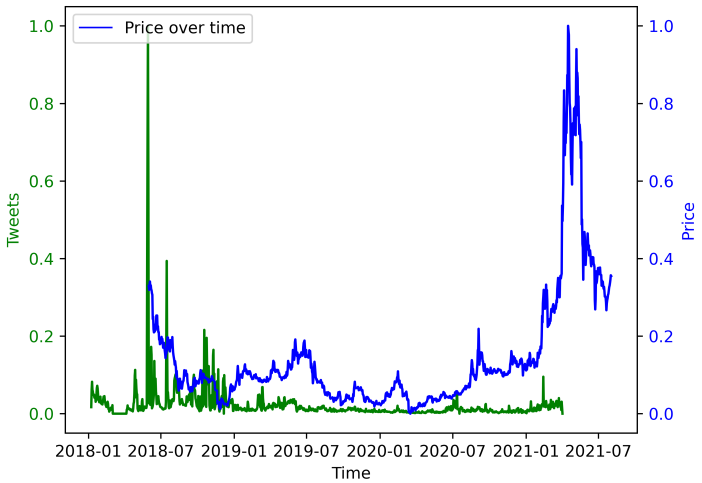
Slika 5.7: EOS - tweets / price



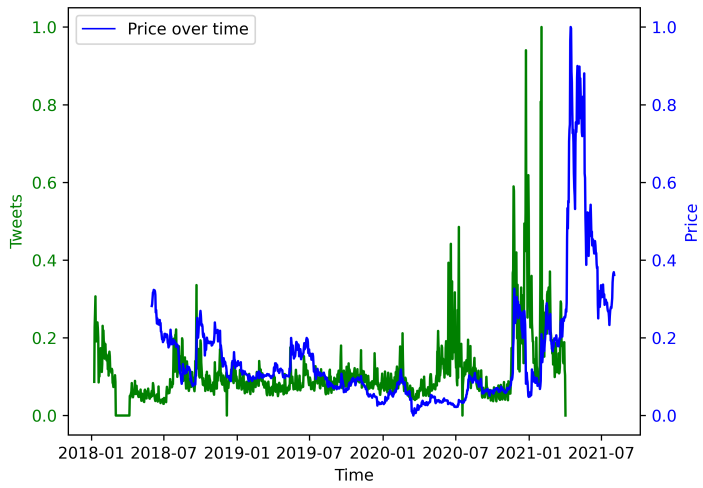
Slika 5.8: ETH - tweets / price



Slika 5.9: LTC - tweets / price



Slika 5.10: TRX - tweets / price



Slika 5.11: XRP - tweets / price

v predhodnjem koraku podatke že normaliziral, bo ta naloga občutno lažja. Razpon obeh naborov podatkov (cena in število zapisov) je že med 0 in 1, kjer 0 predstavlja najmanjšo in 1 največjo zgodovinsko vrednost. Tako sem podobnost teh dveh grafov izračunal kot absolutni razliko ploščin obeh krivulj (po spodnji enačbi).

$$\frac{|\sum_{i=1}^{n} \sqrt{(Xi - Yi)^2}|}{n}$$

Vrednost `n` je v tem primeru število vseh zapisov (dnevov) na katere imamo podatke za ceno in število Twitter zapisov določene kriptovalute. `Xi` je vrednost normalizirane cene kriptovalute na dan `i` , `Yi` pa je normalizirano število Twitter zapisov o tej kriptovaluti na dan `i`.

Rezultat takšne analize je število med 0 in 1, ki nam pove koliko sta si grafa podobna. Število, bližje 0 pomeni, da sta si grafa zelo podobna in obratno.

```
tweets-ada.json
Similarity: 0.1147976904568234

tweets-bch.json
```

```
Similarity: 0.6738463855538037

tweets-bnb.json
Similarity: 0.019814367060063627

tweets-btc.json
Similarity: 0.007913032727385801

tweets-btt.json
Similarity: 0.0045624861032861124

tweets-doge.json
Similarity: 0.0006364729290899146

tweets-eos.json
Similarity: 0.941404916736788

tweets-eth.json
Similarity: 0.02598846186013913

tweets-ltc.json
Similarity: 0.3436510973299928

tweets-trx.json
Similarity: 0.3815699428504037

tweets-xrp.json
Similarity: 0.3451674347483545
```

V zgornji datoteki so zapisani izračuni za vse grafe. Kot smo že prej omenili, imata Bitcoin in Ethereum tudi po izračunu visoko podobnost med grafoma (nizko vrednost blizu 0). Na tej točki je potrebno pogledati tudi vse ekstreme, ki imajo lahko popačen izračun. Pri podobnosti zelo iztopa kriptovaluta DogeCoin, ki je primer relativno mlade kriptovalute, čigar večino rasti so povzročili internetni forumi in sobe, kjer so se investitorji dogovarjali za masovno kupovanje te kriptovalute. Posledično lahko na grafu tudi opazimo močno rast cene in omemb te kriptovalute v samo eni točki, kar povzroči

velik razkorak med minimalno in maksimalno vrednostjo.

5.5 Pridobivanje aktualnih podatkov

V Python programu `scraper.py` je napisana logika za pregledovanje spletne strani BitInfoCharts [1].

```
# function for web scraping
def scrape(coins):
    date = datetime.today().strftime('%Y-%m-%d')
    tweets_current = { 'tweets': [], 'date': date}
    results = []
    browser = webdriver.Chrome()

    for coin in coins:
        name = coin['name']
        symbol = coin['symbol']

        url = 'https://bitinfocharts.com/' + name + '/'
        browser.get(url)

        element = browser.find_element_by_xpath('//tr[td[a[
    contains(text(), "Tweets per day")]]]/td[2]/a')
        tweets = element.text
        print(tweets)

        results.append({ "name": name, "tweet": tweets })

    browser.close()
    tweets_current['tweets'] = results
    return tweets_current
```

Zgoraj je prikazana funkcija za dejansko ekstrakcijo podatkov s spleta. V prvih vrsticah so nastavljeni vsi potrebni podatki, nato pa se Chrome webdriver definira v novo spremenljivko. Za vsak kovanec posebej odpremo ustrezen url, ki ga sestavimo z njegovim imenom, nato pa s pomočjo xPath lociramo podatek o številu Twitter zapisov v tekočem dnevu. Rezultat nato

samo še dodamo v seznam in vrnemu iz funkcije. Program vse zbrane podatke nato zapiše v JSON datoteko, ki se hrani v mapi, dokler do nje ne dostopa strežnik.

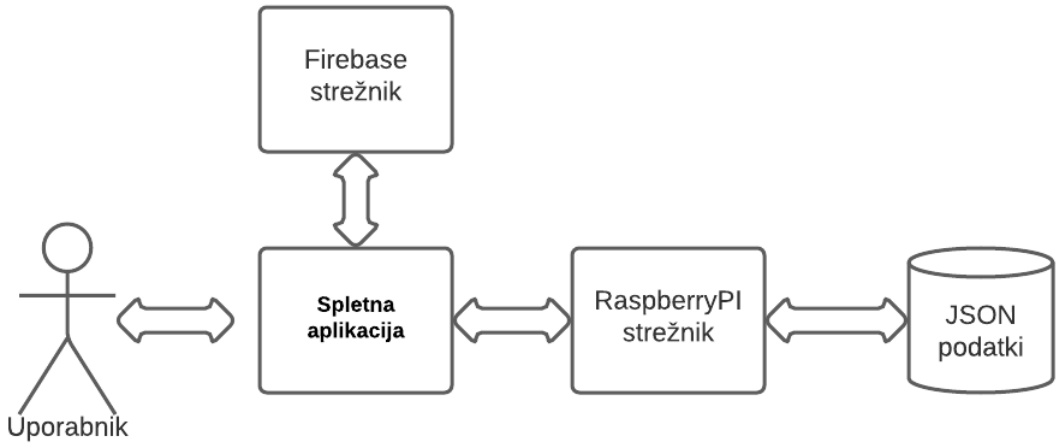
5.6 Strežnik

Podatkovni strežnik z API točko se nahaja znotraj lokalnega omrežja na napravi RaspberryPi. Napisan je v jeziku Javascript in se izvaja v okolju NodeJS. Za lažjo implementacijo sem uporabil še tehnologije Express, REST in NPM. Strežnik je sorazmeroma preprost, njegova naloga je pravilno klasificirati 2 API klica (get in post), ter v obeh primerih opraviti predpisane ukaze. Prvi klic je REST klic GET, pri katerem se sproži branje JSON datotek na strežniku in pripravljanje odgovora. Ko je ta pripravljen se ga zapiše v REST odgovor in pošlje nazaj. Drugi klic je REST klic POST, pri katerem se sproži osveževanje podatkov. Ko pride klic na strežnik, se požene Python program za branje ekstrakcijo podatkov o Twitter zapisih. Ko je to delo opravljeno, se na strežniku nahaja posodobljena datoteka s podatki, ki se jo lahko takrat ponovno pridobi s klicem GET na strežnik.

5.7 Spletna stran

Spletna stran služi kot mesto za prikaz aktualnih podatkov o gledanih kriptovalutah. Te so razvrščene po relevantnosti (tiste kriptovalute, ki so imele faktor podobnosti nižji, so na seznamu vpisane višje).

Spletna stran je napisana s pomočjo JavaScript ogrodja Angular. Uporablja tudi tehnologije HTML, CSS, Javascript in Bootstrap. Na zgornji sliki lahko vidimo postopek komunikacije med deli aplikacije. Uporabnik v spletnem brskalniku najprej dostopa do spletne aplikacije, tako da pošlje zahtevek na strežnik. Ta mu odgovori s spletno aplikacijo, ob nalaganju ta pošlje zahtevno na strežnik RaspberryPi, ki pa ob tem dogodku prebere datoteko z aktualnimi podatki o številu omemb kriptovalut na Twitterju. To pošlje kot



Slika 5.12: Diagram interakcij spletno aplikacijo in strežnikom

odgovor nazaj spletni strani, ki pa nato prejete podatke prikaže uporabniku.

Poglavje 6

Delovanje aplikacije

Celotna koda diplomske naloge je dostopna na GitHub repozitoriju na naslovu `https://github.com/tilenmiklavic/Diploma`. V mapi `website` se nahaja koda celotne spletne strani, ki jo lahko lokalno zaženemo, tako da sledimo spodnjim navodilom:

1. Repozitorij prenesemo na svoj lokalni računalnik z uporabo ukaza `git clone https://g`
   Po uspešni izvedbi tega ukaza, se bo koda iz mape `website` prenesla v trenutno lokacijo na lokalnem računalniku

2. Premaknemo se v delovno mapo projekta z ukazom `cd website/src/app/`

3. Ukaz `npm install` bo poskrbel, da se prenesejo vsi paketi, ki so nujni za delovanje aplikacije.

4. z ukazom `ng serve --host 0.0.0.0` pa poskrbimo, da se spletna stran požene na naslovu `htpp://localhost:4200/`

Ker sem za izdelavo spletne aplikacije uporabljal Javascript ogrodje Angular, imam v osrednji datoteki `main.component.ts` možnost uporabe funkcije `ngOnInit()`, ki mi omogoča izvajanje dela kode vedno ko se omenjena komponenta ponovno naloži.

```
// function for calling service for coins data
private populateTable(): void {
  this.coinService.getCoinsData()
    .then(odgovor => {
      console.log(odgovor)

      this.coins = odgovor.coins
    })
    .catch(napaka => {
      console.error("Napaka:", napaka)
    })
}

ngOnInit(): void {
  this.populateTable()
}
```

V zgoraj omenjeni funkciji sem poklical funkcijo `populateTable()`, ki skbri za to, da se pošlje zahteva za pridobitev podatkov o kovancih na podatkovni strežnik. To se zgodi v asinhronem klicu, kjer se ob uspehu izvede en del kode (odgovor se zapiše v lokalno spremenljivko in prikaže uporabniku), ob neuspehu pa drug (uporabnika se obvesti o napaki).

Sama logika za pošiljanje zahteve na podatkovni strežnik se nahaja znotraj datoteke `coins.service.ts`. V datotekah z oznako `service` se v Angular projektih nahajajo funkcije za komuniciranje z drugimi deli spleta. To prinese k dodatni zaščiti in smiselni strukturiranosti projekta.

```
export class CoinsService {

  constructor(private http: HttpClient) { }
  private url = environment.coins_url

  public getCoinsData(): Promise<any> {


```

```
    return this.http
      .get(this.url)
      .toPromise()
      .then(data => data as any)
      .catch(CoinsService.obdelajNapako)

  }
}
```

V zgornjem izseku lahko vidimo, da je v funkciji `getCoinsData()` definiran HTTP klic tipa GET, ki se izvede na naslov, definiran v `environment.coins_url`. Pričakujemo, da bo klic uspešen, odgovor pa nabor podatkov o kovancih, zato ta pričakovan odgovor vrnemo, v primeru da pride do napake, pa o tem obvestimo uporabnika z izpisom napake v konzolo.

Vse pridobljene podatke pa nato prikažemo uporabniku, kar je v ogrodju Angular močno poenostavljeno.

```
<tbody>
  <tr *ngFor='let coin of coins; let i = index'>
    <th class='align-middle' scope="row">{{i + 1}}</th>
    <td class='align-middle'><img class='icon' src="../../../
   assets/images/{{coin.symbol}}.png" alt="icon"> {{coin.coin
   }} <strong>{{coin.symbol | uppercase}}</strong></td>
    <td class='align-middle'>{{coin.today}}</td>
    <td class='align-middle'>{{coin.min_value | number:
   '1.0-0'}}</td>
    <td class='align-middle'>{{coin.max_value | number:
   '1.0-0'}}</td>
    <td class='align-middle'>{{coin.avg_value | number:
   '1.0-0'}}</td>
  </tr>
</tbody>
```

V zgornjem primeru si lahko ogledamo kako preprosto je vztavljanje in spreminjanje vrednosti v datoteki `main.component.html`. Z ukazom `*ngFor`

se sprehodimo čez celoten seznam `coins` in v vsaki iteraciji prebran element obravnavamo kot spremenljivko `coin`. Nato na določena mesta v HTML datoteki vztavimo atribute spremenljivke coin (coin.today, coin.minValue,...). Na pri tem pristopu je prikaz na spletni strani vedno sinhroniziran s stanjem spremenljivke coins, nam pa ni potrebno skrbeti za dodatno Jaascript kodo, ki bi ob pravem trenutno poskbela za menjavo podatkov v HTML datoteki.

Na spletni strani imamo tudi možnost ročnega osveževanja podatkov. V datoteki `coins.service.js` se nahaja funkcije `refreshCoinsData()`, ob izvedbi katere se s klicem na podatkovni strežnik podatki o kovancih osvežijo.

```
// function for refreshing coins data
public refreshCoinsData(): Promise<boolean> {

  return this.http
    .post(this.url, {"time": Date.now()})
    .toPromise()
    .then(odgovor => odgovor as any)
    .catch(CoinsService.obdelajNapako)
}
```

S HTTP klicem tipa POST na url podatkovnega strežnika sprožimo posodabljanje podatkov. V telo zahtevka pripenmo še trenutni datum za morebitno pregledovanje na strežniku. Ob uspešni izvedbi osveževanja na podatkovnem strežniku, se bo vrnila logična vrednost `True`, v nasprotnem primeru pa `False`. Temu primerno je potem tudi sporočilo za uporabnika.

Na podatkovnem strežniku, se v okolju NodeJS ozvaja datoteka `server.js`, ki vsebuje dve glavni funkciji.

```
app.get('/', (req, res) => {
    console.log("Coins data")
    fs.readFile('coin_data.json', function(err, data) {
        fs.readFile('results.json', function(err, results) {
            let coin_data = JSON.parse(data)
            let results_data = JSON.parse(results)
```

```
7
8             results_data.tweets.forEach(function(el1, ind1) {
9                 coin_data.coins.forEach(function(el2, ind2) {
10                    str1 = el1.name.toLowerCase().replace(/\s
    /g, '')
11                    str2 = el2.coin.toLowerCase().replace(/\s
    /g, '')
12
13                    if (str1 == str2) {
14                        coin_data.coins[ind2].today = el1.
    tweet
15                    }
16                })
17            })
18            res.send(coin_data)
19        })
20     })
21 });
```

Prva funkcija skrbi za obdelavo HTTP zahtev tipa GET. Ko pride takšna zahteva na strežnik, program odpre datoteki `coin_data.json` in `results.json`. V prvi datoteki so zapisani podatki o kovancih (največje in najmanjše število Twitter zapisov, ime, znak,...) v drugi pa rezultati ekstrakcije podatkov o aktualnih Twitter zapisih z internetne strani. Podatke o istem kovancu iz teh dveh datotek hočemo združiti, zato preberemo imen kovancev vseh zapisov iz obeh datotek in podatke z istim imenom združimo pod en zapis v nov seznam. Ta seznam na koncu pošljemo kot odgovor na zahtevo.

Poglavje 7

Rezultati

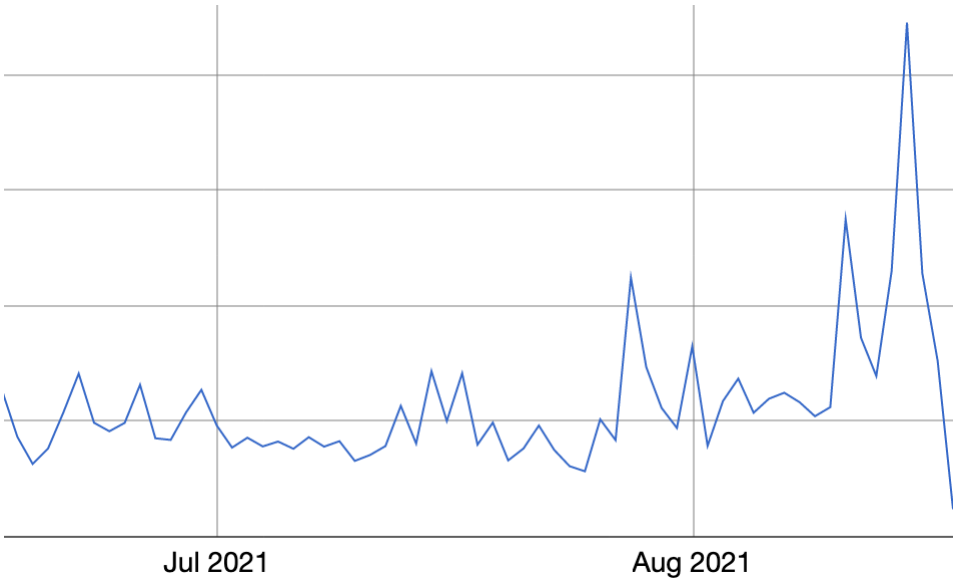
Kot sem ugotovil že pri predstavitvi in izrisu podatkov, nadpovprečno dogajanje na družabnem omrežju Twitter še ne pomeni nujno tudi nihanja cene kriptovalute. Vseeno pa bi se rad dotaknil nekaj primerov, ki sem jih tekom testiranja opazil.

7.1 XRP

Kriptovaluta Ripple je od sredine julija pa do sredine avgusta poskočila s cene 0.53$ na 1.34$. V tem obdobju je spletna stran nekajkrat zaznala nadpovprečno kočilino Twitter zapisov in na to tudi opozorila.

Na sliki 7.1 lahko opazimo postopno rast cene kovanca XRP med julijem in avgustom. Na sliki 7.2 pa število Twitter zapisov v tem istem časovnem obdobju. Tako cena in Twitter zapisi so doživeli lokalno maksimalno vrednost okoli 15. avgusta, med grafoma pa lahko opazimo tudi druge podobnosti (začetek dvigajočega trenda z začetkom avgusta in nadaljevanje tega do 15.8.). To pomeni, da je spletna stran na dogajanje opozorila večkrat med mesecem avgustom.

Slika 7.1: Cena kovanca XRP med julijem in avgustom



Slika 7.2: Twitter zapisi kovanca XRP med julijem in avgustom

Slika 7.3: Cena kovanca BTC med julijem in avgustom

7.2 BTC

Podobno kot XRP, je tudi Bitcoin zabeležil visoko rast tekom meseca avgusta. To lahko vidimo na sliki 7.3. Med potekom rasti kriptovalute, pa ni bilo moč zaznati abnormalnega dogajanja na družabnem omrežju Twitter, kar je razvidno iz slike 7.4.

Do sedaj sem opazil, da Twitter zapisi ne bodo vedno sledili trendu rasti cene kriptovalut, medtem ko je obratno bolj pogosto res. Tekom testiranja sem pogosot lahko zaznal nihanje v ceni posamičnih kriptovalut, medtem ko je bila tudi na Twitterju povečana dejavnost. Iz tega predvidevam, da je bila moja začetna trditev o vplivu Twitter zapisov na ceno kriptovalute postavljena relativno natančno.

Slika 7.4: Twitter zapisi kovanca BTC med julijem in avgustom

Poglavje 8

Zaključek

Cilj diplomske naloge je bil razviti sistem za preverjanje v kakšni meri omembe kriptovalut vplivajo na ceno teh. Vsi ti podatki pa so potem tudi predstavljeni končnemu uporabniku brez tehničnega znanja. Menim, da je bil cilj naloge v veliki meri dosežen, saj sem se v okviru naloge dotaknil vseh predhodno zastavljenih področij. Zbral in analiziral sem potrebne podatke za napovedovanje, napisal sem program za pregledovanje in ekstrakcijo podatkov o številu Twitter zapisov z interneta in potem te podatke ponudil za dostop na podatkovnem strežniku, ki se nahaja v mojem lokalnem omrežju. Izdelal sem tudi spletno stran, ki uporabniku prikazuje vse potrebne podatke. Koda vseh delov naloge je dostopna na odprtem GitHub repozitoriju `https://github.com/tilenmiklavic/Diploma`. Ker se zavedam, da so izboljšave mogoče še na vseh delih kode, je repozitorij odprtokoden in lahko vsakdo prispeva in predlaga izboljšave. Sam vidim največ prostora za izboljšavo spletne strani, ki je trenutno v pretežno osnovnem izgledu in morda za končnega uporabnika ni dovolj privlačna. Predstavitev podatkov na izboljšan način, prikaz in shranjevanje dodatnih informacij (zadnje abnormalno dogajanje, delež današnjega skoka / padca cene,...). Ravno tako bi polek predstavitve informacij o dogajanju na družabnem omrežju Twitter lahko na spletni strani prikazal tudi aktualne informacije o cenah kriptovalut, dnevne razlike, tržni kapitalizaciji, aktualnih novicah in druge. Z več podatki na

enem mestu bi tako lahko uporabniki čutili manjšo težnjo po uporabi več virov za pridobivanje potrebnih informacij, kar bi jim eventuelno naredilo pogled in izkušnjo uporabe te spletne strani bolj pozitivno. Strežnik bi rad razširil z več API dostopnimi točkami, ki bi spletni strani in drugim odjemalcem ponudile širšo možnost uporabe. V primeru da bi se dnevno število obiskovalcev na spletni strani močno povišalo pa bi strežnik tudi prestavil na gostovanje na oblak, saj bi omejeni viri pri domačem gostovanju (manjša pasovna širina, omejena procesorska moč mikroračunalnika RaspberryPi,...) lahko povzročili slabšo uporabniško izkušnjo.

Literatura

[1] Bitinfo charts. URL https://bitinfocharts.com.

[2] Coin desk. URL https://www.coindesk.com/price/dogecoin.

[3] Google firebase. URL https://firebase.google.com/.

[4] Reddit - wall street bets, 2012. URL https://www.reddit.com/r/wallstreetbets/.

[5] Coin market cap, 2013. URL https://coinmarketcap.com/.

[6] Reddit - satoshi street bets, 2020. URL https://www.reddit.com/r/SatoshiStreetBets/.

[7] Daniel; Nelson John; Abraham, Jethin; Higdon and Juan Ibarra. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3), 2018.

[8] Olivier Kraaijeveld and Johannes De Smedt. The predictive power of public twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, 65: 101188, 2020. ISSN 1042-4431. doi: https://doi.org/10.1016/j.intfin.2020.101188. URL https://www.sciencedirect.com/science/article/pii/S104244312030072X.