УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

КАТЕДРА ЗА УПРАВЉАЊЕ СИСТЕМИМА

ПРОЈЕКТНИ РАД

ПРЕДВИЂАЊЕ ТРЕНДА КРИПТОВАЛУТА НА ОСНОВУ ОБЈАВА НА ДРУШТВЕНИМ МРЕЖАМА

Ментор:
доц. др Александар Ракићевић

Студенти:
Јанко Гашић 134/2016
Иван Прелић 421/2016
Никола Ђорђиески 4/2017
Анђелија Милутиновић 12/2017
Никола Керезовић 1053/2017

Београд, јун 2021.

Садржај

1. Увод 4

2. Прикупљање и претпроцесирање података 6

2.1 Начин прикупљања 6

2.2 Припрема података 7

3. Модел тржишног сентимента 10

3.1 Рачунање сентимента 10

3.2 Спајање и груписање података 11

3.2.1 Рачунање претходног тренда 13

3.2.2 Рачунање будуће промене 14

3.2.3 Нове колоне 15

3.3 Статистика над коначном табелом 16

4. Преглед коришћених метода рачунарске интелигенције 22

4.1 Методе класификације 22

4.2 Неуронска мрежа – вишеслојни перцептрон 23

4.3 Рекурентна неуронска мрежа ЛСТМ 24

5. Предвиђање ценовног тренда – проблем класификације 25

5.1 Стабла одлучивања 25

5.2 К-најближих суседа 25

6. Предвиђање ценовног тренда – неуронске мреже 27

6.1 Примена неуронске мреже вишеслојни перцептрон 27

6.1.1 Резултати мреже 28

6.1.2 Анализа резулата 29

6.2 Примена рекурентне неуронске мреже – LSTM 29

6.2.1 Предикција цене Биткоина 30

6.2.2 Предикција цене Етеријума 33

6.2.3 Предикција цене Рипла 34

7. Унакрсна корелација промена цене и тржишног сентимента 36

8. Подаци са осталих друштвених мрежа 40

8.1 Сентимент анализа за Твитер објаве 40

9. Закључак 43

10. Литература 44

1. Увод

Промене вредности Биткоина (енг. *Bitcoin*) последњих година су привукле велику пажњу како инвеститора тако и академске заједнице. Основно питање којим се аналитичари баве јесте да ли ће вредност Биткоина пасти или порасти, односно, да ли треба купити или продати токене. Постоје бројна истраживања, а на некима се и данас ради, која покушавају да одговоре на ово питање. Велика волатилност и сама флуктуација вредности ове криптовалуте створила је изузетну заинтересованост према овом тржишту. Због своје непредвидивости и немогућности да се нађе директна веза између промене цене и дешавања на тржишту, као нови аспект у овој анализи додато је поверење друштвене заједнице у ову криптовалуту. Управо том аспекту је посвећена посебна пажњу у овом раду.

Овај рад представља проширење и надоградњу на завршни рад Матије Милекића [3]. Разлика у односу на тај рад је то што је сада проширено истраживање са анализе друштвеног сентимента са тридескуп најбољих објава у том дану на све објаве у том дану, додате су криптовалуте Етеријум (енг. *Ethereum*) и Рипл (енг. *Ripple*). Поред тога, урађена је и сентимент анализа квалитативних података како би се боље схватиле објаве на друштвеној мрежи Редит (енг. *Reddit*).

Предмет овог рада је креирање система за предвиђање тренда, односно алгоритамско трговање на тржишту криптовалута. Пројекат је базиран на вештачким неуронским мрежама, техничкој и сентимент анализи.

Циљ овог рада је да се истражи да ли постоји веза између сентимета откривеног на друштвеној мрежи Редит и промене цене код Биткоина, Етеријума и Рипла. Овај рад тежи да покаже везу Редита и наведених криптовалута и то у периоду 01.12.2020.-01.05.2021, на основу дневних података који су прикупљени са интернета. У резултатима се може видети да је систем успео да оствари предикције које се подударају са стварним трендом што нам може донекле потврдити нашу претпоставку да је психологија тржишта битан фактор на овом тржишту.

Креирање система, обучавање и тестирање мреже су извршени у развојном окружењу и програмском језику Пајтон (енг. *Python*), а и скрипта за извлачење података је

писана уз помоћ Пајтон програмског језика. Подаци су преузети са друштвене мреже Редит и сајта *Coinmarketcap*.

2. Прикупљање и претпроцесирање података

За потребе пројекта подаци су прикупљани са друштвене мреже Редит (енгл. Reddit). Подаци су прикупљани у периоду од 1. децембра 2020. до 1. маја 2021, а криптовалуте за које су прикупљани подаци су Биткоин (*BTC*), Етериум(*ETH*) и Рипл(*XRP*). Скрипта за прикупљање података је написана у програмском језику Python, а коришћена су и два АПИ-ја, *Pushishift* и *PRAW*.

За сваки дан у наведеном периоду је скрејповано (енгл. *scrape*) 100 најпопуларнијих објава са сваког сабредита (енгл. *subreddit*) за наведене крипотвалуте. Све три табеле су сачињене од 13 колона, али због разлике у величини и популарности сабредита свака табела има различити број редова. Табела за Биткоин се састоји од 13099 опсервација, табела за Етеријум од 11336 опсервација, а табела за Рипл од 4499 опсервација. Редови који се налазе у скуповима података садрже следеће атрибуте:

- *Post id* - представља јединствену идентификацију поста додељену од стране Pushshift-a.
- *Title* - предстаља наслов објаве.
- *Date and Time* - представља датум и време када је објава постављена на Редит.
- *Date* - представља датум када је објава постављена на Редит.
- *Time* - представља време када је објава постављена на Редит.
- *Hour* - представља сат када је објава постављена на Редит. Ова колона је извучена како би се касније лакше вршила обрада података.
- *Upvotes* - представља укупан број позитивних гласова на објави.
- *Compound Sentiment* - укупан сентимент добијен помоћу VADER лексикона.
- *Voted Score* - коначни сентимент.
- *Positive Sentiment* - укупан број позитивних речи у једној објави.
- *Negative Sentiment* - укупан број негативних речи у једној објави.
- *Sentiment Score* - разлика између позитивних и негативних речи у једној објави.

У оквиру поглавља везаног за тржишни сентимент, приказана је структура самог скупа података.

2.1 Начин прикупљања

Као изабрани програмски језик, помоћу ког су се подаци преузимали изабран је Пајтон. Неопходне библиотеке приказане су на исечку кода испод.

```
import requests
import json
import datetime
import time
import re
import pandas
import openpyxl
import praw
import nltk

nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

Дефинише се функција која као параметре прима назив сабредита, почетно и крајње време одабраног периода, и *boolean* вредност која ће у овом случају увек бити подешена на *false*. У оквиру ове функције дефинише се упит (енг. *Query*), помоћу којег АПИ прикупља податке.

```
api_search_query = ("https://api.pushshift.io/reddit/search/
submission/?subreddit={}&sort_type=score&sort=desc&after={}&before="
    "{}&category=best&size=500")
if len(list_of_ids) != 0:  # If we don't have ANY data, just return an
empty dictionary.
    # Get info for each object. `info()` is quite fast and accepts a list
of fullname IDs.
    reddit_submissions = reddit.info(fullnames=list_of_ids)
    for submission in reddit_submissions:
        # post_score = submission.score
        final_dictionary[post_id] = {}
        final_dictionary[post_id]['id'] = submission.id
        final_dictionary[post_id]['score'] = submission.score
        final_dictionary[post_id]['title'] = submission.title
        final_dictionary[post_id]['link'] = submission.permalink
        final_dictionary[post_id]['created_utc'] =
int(submission.created_utc)
        final_dictionary[post_id]['author'] = submission.author
        final_dictionary[post_id]['num_comments'] =
submission.num_comments
        final_dictionary[post_id]['self_text'] = submission.selftext
        final_dictionary[post_id]['compound_sentiment'] = 0
        final_dictionary[post_id]['voted_score'] = 0
        final_dictionary[post_id]['positive_sentiment'] = 0
        final_dictionary[post_id]['negative_sentiment'] = 0
        final_dictionary[post_id]['sentiment_score'] =
final_dictionary[post_id]['positive_sentiment'] - \

final_dictionary[post_id]['negative_sentiment']
        post_id += 1
```

2.2 Припрема података

Вредноти атрибута се чувају у речник (енг. *dictionary*) из ког ће подаци бити учитани у *data frame*.

Анализа тржишног сентимента је извршена уз помоћ VADER лексикона, који је део NLTK библиотеке. Детаљно објашњење ове анализе приказано је у наредном поглављу. Анализа је извршена одвојено на наслову и тексту објаве, а затим је спојена у променљиву *text_and_title_score*. Укупан број позитивних реакција на објаву је представљен променљивом *post_score*.

```
# Dodavanje sentimenta svakom postu
j = 0
while j < 100:

    # TITLE ANALYSYS
    title = final_dictionary[j]['title']
    title_score = sid.polarity_scores(title)['compound']
    # print(title_score)

    text = final_dictionary[j]['self_text']
    text_score = sid.polarity_scores(text)['compound']
    # print(text_score)

    text_and_title_score = title_score + text_score
    # print(text_and_title_score)

    post_score = final_dictionary[j]['score']
    # print(post_score)
    voted_score = text_and_title_score * (1 + post_score * 0.2)
    # print(voted_score)

    final_dictionary[j]['compound_sentiment'] = text_and_title_score
    final_dictionary[j]['voted_score'] = voted_score
```

Наредна петља која се креира позива функцију једном по дану и прикупља 100 најпопуларнијих објава тог дана. Променљива *start* прима *integer* вредност која представља 1. децембар 2020. 00:00:00, а *end* променљива представља 1. децембар 2020. 23:59:59. Обе променљиве су у формату *epoch-time* (UNIX време) . Променљива *p* представља број дана који желимо да обухватимо.

```
start = 1606777200  # 1.dec2020
end = 1606863599
p = 0

while p < 151:
    d = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime(start))
    print(d)
    try:
        subreddit_pushshift_time_top_retriever('Bitcoin', start, end,
last_month_mode=False)
    except:
        print("Greska 1")
    start += 86400
    end += 86400
    p += 1
```

На самом крају, подаци се чувају у *data frame* уз помоћ *Pandas* библиотеке, а затим се уписују у *Excel* фајл.

```
new_dataframe = pandas.DataFrame(
    {
        "Post id": ids,
        "Epoch time": epoch_times,
        "Title": titles,
        "Date": dates,
        "Upvotes": scores,
        "Compound Sentiment": compound_sentiments,
        "Voted Score": voted_scores,
        "Positive Sentiment": positive_sentiments,
        "Negative Sentiment": negative_sentiments,
        "Sentiment Score": sentiment_scores,
        "Username": authors,
        "Num of comments": comments
    }
)

writer = pandas.ExcelWriter('outputBITCOIN.xlsx')
try:
    new_dataframe.to_excel(writer, 'Sheet1')
except:
    print("Greska 2")
writer.save()
```

3. Модел тржишног сентимента

У овом поглављу биће објашњен поступак моделовања сентимента који се даље користио за предвиђање тренда криптовалута.

Одлучено је да коначни (тежински) сентимент обједињује два елемента објаве на Редиту: текст објаве и разлику између позитивних и негативних гласова (енг. *Upvotes and Downvotes*). [1] Наиме, на текст објаве примењена је сентимент анализа, а гласови на објави на Редиту додају тежину објави, с обзиром да ова друштвена мрежа рангира објаве са великом разликом између позитивних и негативних гласова на вишу позицију на страници и оне постају све популарније са повећањем те разлике.

Коментари на објавама нису узети у обзир при рачунању коначног сентимента, иако је почетна идеја била да се и они укључе у модел и додају тежину објавама. Након истраживања о коментарима као факторима сентимента на објавама, разлог за овакву одлуку је био тај што би такво додавање тежина прилично утицало на усложњавање модела, а то је требало избећи у овом раду.

3.1 Рачунање сентимента

За рачунање сентимента текстова користи се VADER лексикон (енг. *Valence Aware Dictionary and sEntiment Reasoner*) у оквиру NLTK библиотеке у програмском језику *Python*. Уз помоћ VADER лексикона, као излаз процеса рачунања сентимента добија се такозвани речник (енг. *Dictionary*) у *Python*-у који садржи четири вредности. Наиме, први елемент речника означава негативни сентимент, други елемент позитивни сентимент, трећи је неутрални сентимент и четврти елемент је сложени сентимент (енг. *Compound sentiment*). Сложени сентимент је најзначајнији у овом случају и он је коришћен за даље рачунање. Ова вредност представља управо агрегацију остале три вредности.

За одређени текст, VADER лексикон даје сложени сентимент у виду нумеричке вредности у распону [-1, 1]. Наиме, негативне вредности означавају негативни сентимент, позитивна вредност означава позитивни сентимент, док 0 означава да се текст не може категоризовати ни као позитиван ни као негативан. Такође, што је

негативна вредност ближа -1, то и текст има негативнији сентимент, и обрнуто што је позитивна вредност ближа броју 1 и текст има позитивнији сентимент.

```
j = 0
    while j < 100:
        positive_counter = 0
        negative_counter = 0
        # TITLE ANALYSYS
        title = final_dictionary[j]['title']
        title_score = sid.polarity_scores(title)['compound']

        text = final_dictionary[j]['self_text']
        text_score = sid.polarity_scores(text)['compound']

        text_and_title_score = title_score + text_score
```

За сваку објаву, израчунат је прво сентимент наслова објаве и засебно текста објаве, затим се ови сентименти сабирају и користе даље у формули. То представља сложени сентимент. Већ је поменуто да је друга варијабла у нашем моделу разлика позитивних и негативних гласова, а у скупу података ова варијабла већ постоји као таква и именована је са *score*. Коначни, односно тежински сентимент формира се на основу претходна два израчуната, а математички и програмски део су наведени испод.

```
post_score = final_dictionary[j]['score']
voted_score = text_and_title_score * (1 + post_score * 0.2)
```

Дакле, ознаке за све елементе у нашем скупу података потребних за рачунање коначног сентимента су:

- *Compound score* - сложени сентимент
- *Score* - разлика позитивних и негативних гласова
- *Voted score* - коначни (тежински) сентимент

На основу датих ознака, формула за рачунање коначног (тежинског) сентимента [1]:

$$Voted\ score\ = Compound\ score \times\ (\ 1\ +\ Score\ \times 0.2)$$

3.2 Спајање и груписање података

Након што је израчунат тежински сентимент и додат у скуп података, следећи корак био је да се споје скупови података у један скуп: подаци о објавама са Редита са израчунатим сложеним сентиментом и историјски подаци о ценама криптовалута и то по сатници у којој је објава настала. Направљене су нове колоне у ова два скупа

података које су садржале сатнице за сваки ред. У програмском језику Пајтон креирана је скрипта која спаја ова два скупа у јединствени скуп на основу колона: датум и време.

Након спајања скупова података, било је потребно структурирати податке како би се на основу њих правила предвиђања. Дакле, податке је било неопходно груписати по сатима, тако да за сваки сат у дану постоје следеће вредности:

- Укупан број објава за тај сат;
- Сума свих сентимената за тај сат – сума сама по себи садржи више информација од просечне вредности сентимената за један сат, јер се на основу суме може закључити да ли је било више од једне објаве у том сату и то даје већи значај сентименту;
- Почетна вредност криптовалуте за тај сат;
- Крајња вредност криптовалуте за тај сат;

Табела 1 – Добијене колоне након спајања

Row Labels	Count of Posts	Sum of Voted Score	BTC open value	BTC close value
2020-12-01 00:00:00	1	1.25554	19695.87	19565.47
2020-12-01 01:00:00	2	0.72744	19565.47	19605.75
2020-12-01 02:00:00	5	-0.10266	19605.75	19680.95
2020-12-01 03:00:00	5	1.40806	19680.96	19419.74
2020-12-01 04:00:00	4	2.72938	19419.73	19354.31
2020-12-01 05:00:00	7	1.87596	19352.64	19483.73
2020-12-01 06:00:00	5	2.53004	19483.73	19338.34

За груписање података користи се *Pivot* табела у оквиру Excel-а.

Последњи корак у припреми података за неуронску мрежу био је израчунавање следећих колона:

- Промена – ова колона представља разлику вредности криптовалуте на крају и на почетку сата (енг. *close and open value*);
- Знак промене – знак претходне колоне;
- Претходни тренд – показатељ кретања цена у претходна 24 часа;
- Знак претходног тренда – знак претходне колоне;
- Будућа промена – процентуална вредност промене цене у 12. сату од посматраног тренутка;
- Знак будуће промене – знак претходне колоне;

3.2.1 Рачунање претходног тренда

За разумевање модела претходног тренда увешћемо ознаке:

- cv_{end}: вредност криптовалуте на затварању у 24.часу
- cv_{start} : вредност криптовалуте на затварању у 1. часу
- $\sum_{start}^{end} signs$: збир знакова промене
- T: тренд

Да бисмо промене у претходна 24 часа сматрали позитивним трендом морају бити исупуњена два услова:

- Услов 1: $cv_{end} - cv_{start} > 0$
- Услов 2: $\sum_{start}^{end} signs > 0$

Да бисмо промене у претходна 24 часа сматрали негативним трендом морају бити исупуњена два услова:

- Услов 1: $cv_{end} - cv_{start} < 0$
- Услов 2: $\sum_{start}^{end} signs < 0$

У оба случаја, за вредност претходног тренда се поставља:

$$T = \frac{cv_{end} - cv_{start}}{cv_{start}} \times 100$$

У сваком другом случају, односно уколико не постоји ни позитиван ни негативан тренд каже се да тренд не постоји и тада се за вредност претходног тренда поставља 0.

Табела 2 – Неопходни услови за рачунање тренда и вредности тренда

Previous trend calculation	$\sum_{start}^{end} signs > 0$	$\sum_{start}^{end} signs < 0$	$\sum_{start}^{end} signs > 0$
$cv_{end} - cv_{start} > 0$	T	0	0
$cv_{end} - cv_{start} < 0$	0	T	0
$cv_{end} - cv_{start} = 0$	0	0	0

На основу претходне табеле, уколико су збир знакова и израчунати тренд истог знака, вредност претходног тренда се узима у обзир, у супротном се сматра да до промене тренда није дошло.

3.2.2 Рачунање будуће промене

Будућа промена се рачуна на основу временског периода од 12 часова из прошлости. За објашњење будуће промене, потребне су нам следеће ознаке:

- cv_{end2}: вредност криптовалуте на затварању у 12. часу;
- cv_{start2}: вредност криптовалуте на затварању у 1.часу.
- P: будућа промена

Вредност будуће промене биће:

$$P = \frac{cv_{end2} - cv_{start2}}{cv_{start2}} \times 100$$

3.2.3 Нове колоне

Након рачунања неопходних метрика, скуп података добија додатних 6 колона. Њихова структура приказана је у табели испод.

Табела 3 – Израчунате колоне

Shift	*Shift sign*	*Previous trend*	*Previous trend sign*	*Future trend*	*Future trend sign*
-130.05	-1	-4.09144273	-1	1.8	1
71.55	1	-3.923542838	-1	1.54	1
17.51	1	-4.201728067	-1	0.46	1
-177.61	-1	-3.827754646	-1	2.4	1
-58.14	-1	-3.803080554	-1	1.57	1
-67.04	-1	-4.787430333	-1	1.84	1
103.46	1	-3.536653094	-1	1.08	1

3.3 Статистика над коначном табелом

Над коначном табелом података, применили смо дескриптивну статистику и израчунали смо метрике тачности. Направљена је још једна колона у нашем скупу података – *Voted Score Sign*, као знак збира коначног сентимента.

Наиме, одлучили смо да проверимо у каквој су вези збир коначног сентимента и претходни тренд са будућом променом, па смо тако направили категорије: број тачно и број нетачно позитивно предвиђених будућих промена, број тачно и број нетачно негативно предвиђених будућих промена, број тачно и број нетачно неутрално предвиђених будућих промена, број тачно предвиђених будућих промена само на основу коначног сентимента и на крају број тачно предвиђених будућих промена само на основу претходног тренда.

Увели смо следеће ознаке за потребе лакшег рачунања:

- *brojacTP* – број тачно предвиђених позитивних будућих промена
- *brojacFP* - број нетачно предвиђених позитивних будућих промена
- *brojacTN* - број тачно предвиђених негативних будућих промена
- *brojacFN*- број нетачно предвиђених негативних будућих промена
- *brojacTNe* - број тачно предвиђених неутралних будућих промена
- *brojacFNe* - број нетачно предвиђених неутралних будућих промена
- *brojacTScore* – број тачно предвиђених будућих промена само на основу коначног сентимента
- *brojacTTrend* - број тачно предвиђених будућих промена само на основу претходног тренда
- *scoreSign* – знак коначног сентимента
- *previousTrendSign* – знак претходног тренда
- *futureSign* – знак будуће промене
- *score* – коначни сентимент
- *previousTrend* – претходни тренд
- *futureValue* – будућа промена

Поред ових променљивих, направили смо матрице за сваку од наведених категорија, у којима смо чували вредности коначног сентимента, претходног тренда и будуће промене за сваку опсервацију која испуњава услове за одређену категорију.

На основу бројача које смо увели, израчунали смо колики је удео сваке од наведених категорија у целом скупу података.

```
while i < len(df):
    scoreSign = df.at[i, 'Voted Score Sign']
    previousTrendSign = df.at[i, 'Znak Prethodni']
    futureSign = df.at[i, 'Znak Buduca']
    score = df.at[i, 'Sum of Voted Score']
    previousTrend = df.at[i, 'Prethodni Trend']
    futureValue = df.at[i, 'Buduca Promena']
    if (scoreSign > 0) and (previousTrendSign > 0) and (futureSign> 0):
        brojacTP+=1
        matTP[(brojacTP-1), :] = np.array([score, previousTrend,
futureValue])
    elif (scoreSign < 0) and (previousTrendSign < 0) and (futureSign <
0):
        brojacTN+=1
        matTN[(brojacTN-1), :] = np.array([score, previousTrend,
futureValue])
    elif (scoreSign > 0) and (previousTrendSign > 0) and (futureSign <=
0):
        brojacFP+=1
        matFP[(brojacFP-1), :] = np.array([score, previousTrend,
futureValue])
    elif (scoreSign < 0) and (previousTrendSign < 0) and (futureSign >=
0):
        brojacFN+=1
        matFN[(brojacFN-1), :] = np.array([score, previousTrend,
futureValue])
    elif (scoreSign == 0) and (previousTrendSign == 0) and (futureSign ==
0):
        brojacTNe+=1
        matFNe[(brojacFNe-1), :] = np.array([score, previousTrend,
futureValue])
    elif (scoreSign == 0) and (previousTrendSign == 0) and (futureSign !=
0):
        brojacFNe+=1
        matFNe[(brojacFNe-1), :] = np.array([score, previousTrend,
futureValue])
    elif scoreSign == futureSign and scoreSign!=previousTrendSign:
#pogodjen znak buduceg samo na osnovu voted score
        brojacTScore+=1
        matTScore[(brojacTScore-1), :] = np.array([score, previousTrend,
futureValue])
    elif previousTrendSign == futureSign and
previousTrendSign!=scoreSign: #pogodjen znak buduceg samo na osnovu
trenda
        brojacTTrend+=1
        matTTrend[(brojacTTrend-1), :] = np.array([score, previousTrend,
futureValue])
    else:
        brojacOstalo+=1
    i+=1
```

Након рачунања ових метрика, у Екселу је направљен табеларни приказ удела ових категорија за све три криптовалуте.

Табела 4 – Приказ метрика за све три криптовалуте

	TP	***FP***	***TN***	***FN***	***TNe***	***FNe***	***TScore***	***TTrend***	***Ostali slučajevi***
BTC	0.127	0.121	0.016	0.029	0.000	0.034	0.326	0.124	0.224
ETH	0.120	0.128	0.014	0.026	0.000	0.063	0.316	0.146	0.187
XRP	0.076	0.069	0.032	0.035	0.000	0.125	0.249	0.161	0.252

Направљене матрице смо претворили у *data frame*, преко кога смо их уписали у одвојене странице (*sheets*) у Екселу. Изглед сваког sheet-а и објашњење биће дати испод.

Табела 5 – Приказ тачно предвиђене позитивне будуће промене

	Sum of Voted Score	**Previous Trend**	**Future Change**
0	40.66146	0.24	0.65
1	1.73364	1.32	1.22
2	1.53638	1.84	1.32

Код табеле тачно предвиђене позитивне будуће промене, видимо да су вредности у колонама *Sum of Voted Score* и *Previous Trend* позитивне, и да је будућа промена такође позитивна.

Табела 6 – Приказ нетачно предвиђене позитивне будуће промене

	Sum of Voted Score	**Previous Trend**	**Future Change**
0	1.79276	1.36	-0.4
1	0.11352	1.99	-2.69
2	0.132	2.04	-0.31

Код табеле нетачно предвиђене позитивне будуће промене, видимо да су вредности у колонама *Sum of Voted Score* и *Previous Trend* позитивне, и да је будућа промена негативна.

Аналогно овим примерима, код табеле тачно предвиђене негативне будуће промене, вредности у колонама *Sum of Voted Score* и *Previous Trend* су негативне, и будућа промена је негативна.

Табела 7 – Приказ тачно предвиђене негативне будуће промене

	Sum of Voted Score	Previous Trend	Future Change
0	-0.50568	-1.76	-0.59
1	-22.45344	-1.46	-0.4
2	-0.81886	-0.26	-1.49

У табели нетачно предвиђене негативне будуће промене, вредности у колонама *Sum of Voted Score* и *Previous Trend* су негативне, и будућа промена је позитивна.

Табела 8 – Приказ нетачно предвиђене негативне будуће промене

	Sum of Voted Score	Previous Trend	Future Change
0	-5.9319	-4.09	1.8
1	-0.19714	-2.82	1.83
2	-1.3886	-2.43	1.39

Табеле тачно предвиђених неутралних промена су празне за Биткоин и Етеријум, односно нема ситуација у којима су и *Sum of Voted Score*, *Previous Trend* и *Future Change* били неутрални. Код Рипла, постоји само једна опсервација у овој категорији.

Табела 9 – Приказ тачно предвиђене неутралне будуће промене

	Sum of Voted Score	Previous Trend	Future Change
0	0	0	0

У табели нетачно предвиђене неутралне будуће промене, вредности у колонама *Sum of Voted Score* и *Previous Trend* су неутралне, и будућа промена није неутрална, односно различита је од нуле.

Табела 10 – Приказ нетачно предвиђене неутралне будуће промене

	Sum of Voted Score	Previous Trend	Future Change
0	0	0	-2.73

1	0	0	-2.66
2	0	0	0.29

У табели тачно предвиђених будућих промена само на основу знака коначног сентимента, знак вредности у колонама *Voted Score* и *Future Change* је исти, и различит од знака вредности у колони *Previous Trend.*

Табела 11 – Приказ тачно предвиђене будуће промене на основу знака коначног сентимента

	Sum of Voted Score	**Previous Trend**	**Future Change**
0	-0.10266	0	-4.68
1	1.70616	0	1.15
2	-1.1357	0	-2.24
3	0.69146	0	0.53
4	0.65328	0	0.82
5	2.86696	-3.92	1.54

Аналогно са претходном табелом, у табели тачно предвиђених будућих промена само на основу знака претходног тренда, знак вредности у колонама *Previous Trend* и *Future Change* је исти, и различит од знака вредности у колони *Voted Score.*

Табела 12 – Приказ тачно предвиђене будуће промене на основу знака претходног тренда

	Sum of Voted Score	**Previous Trend**	**Future Change**
0	-0.30008	2.89	1.38
1	-1.2839	0.07	0.85
2	1.6838	-0.57	-0.91
3	0.10224	-1.09	-0.89
4	1.65134	-0.84	-2.11
5	0	-2.25	-0.81

Након прављења ових табела, израчунали смо и просечне вредности сваке од ових варијабли за сваку категорију, као и стандардне девијације за сваку варијаблу и у ту сврху користили смо Пивот табелу у Екселу.

Приказ табеле са свим израчунатим дескриптивним статистика је дат испод.

Табела 13 – Дескриптивне статистике

True Positive		
Average Of Sum Of Voted Score	*Average Of Previous Trend*	*Average Of Future Change*
16.450945	**5.114215116**	**2.737761628**
StdDev Of Sum Of Voted Score	*StdDev Of Previous Trend*	*StdDev Of Future Change*
49.0286512	**3.948553925**	**2.562662369**
False Positive		
Average Of Sum Of Voted Score	*Average Of Previous Trend*	*Average Of Future Change*
22.71852031	**5.340552147**	**-1.952576687**
StdDev Of Sum Of Voted Score	*StdDev Of Previous Trend*	*StdDev Of Future Change*
74.11606775	**5.087514823**	**1.966596619**
True Negative		
Average Of Sum Of Voted Score	*Average Of Previous Trend*	*Average Of Future Change*
-25.29898381	**-4.407142857**	**-3.014285714**
StdDev Of Sum Of Voted Score	*StdDev Of Previous Trend*	*StdDev Of Future Change*
97.221558	**4.017998704**	**2.908245756**
False Negative		
Average Of Sum Of Voted Score	*Average Of Previous Trend*	*Average Of Future Change*
-12.50138727	**-5.706493506**	**2.586493506**
StdDev Of Sum Of Voted Score	*StdDev Of Previous Trend*	*StdDev Of Future Change*
44.76848745	**4.959226321**	**2.455869321**
False Neutral		
Average Of Sum Of Voted Score	*Average Of Previous Trend*	*Average Of Future Change*
0	**0**	**1.138817204**
StdDev Of Sum Of Voted Score	*StdDev Of Previous Trend*	*StdDev Of Future Change*
0	**0**	**4.321497287**
True Prediction According To The Voted Score		
Average Of Sum Of Voted Score	*Average Of Previous Trend*	*Average Of Future Change*
16.20174865	**-0.684084187**	**1.58817975**
StdDev Of Sum Of Voted Score	*StdDev Of Previous Trend*	*StdDev Of Future Change*
105.542943	**3.585954351**	**3.588164245**
True Prediction According To The Previous Trend		
Average Of Sum Of Voted Score	*Average Of Previous Trend*	*Average Of Future Change*
7.308021138	**0.281766467**	**-0.173023952**
StdDev Of Sum Of Voted Score	*StdDev Of Previous Trend*	*StdDev Of Future Change*
75.83389324	**5.344253757**	**4.232244893**

4. Преглед коришћених метода рачунарске интелигенције

Током израде пројекта коришћене су различите методе рачунарске интелигенције ради предикције ценовног тренда. У зависности од конретне методе, предвиђа се одговарајући тип вредности.

Иако је конкретан проблем потребно решити коришћењем неуронских мрежа, као додатне методе коришћене су најпримењивије методе класификације. Ове методе послужиће за упоређивање са осталим резултатима.

4.1 Методе класификације

У сврху упоредне анализе са добијеним резултатима, коришћене су и методе класификације, тј. метода стабла одлучивања (енг. *decision tree learning*) и метода к-најближих суседа (енг. *k-nearest neighbors algorithm*).

Метода стабла одлучивања један је од најкоришћенијих приступа предиктивног моделирања који се користи у многим областима статистике и машинског учења. Овај модел користи стабло одлучивања (као предиктивни модел) за анализу података, тренирање и на крају од наученог примењује на подацима намењеним за тестирање. Ова метода важи за врло интуитиван (читљив) и једноставан начи машинског учења.

Метода к-најближих суседа представља тип учења који важи за најједноставнији метод учења и заснован је на примерима. Овај метод омогућава повећање поузадности саме класификације. За креирање предикција је неопходно навести скуп података за тренирање и тестирање, као и број суседа који ће се посматрати.

Оба модела као резултат дају предвиђене категоричке променљиве, тј. у овом случају резултат је -1, 0 или 1, односно знак будуће промене тренда. Као метрика поузданости ових метода користиће се тачност (енг. *accuracy*) која се добија дељењем погођених категорија са укупним бројем предикција.

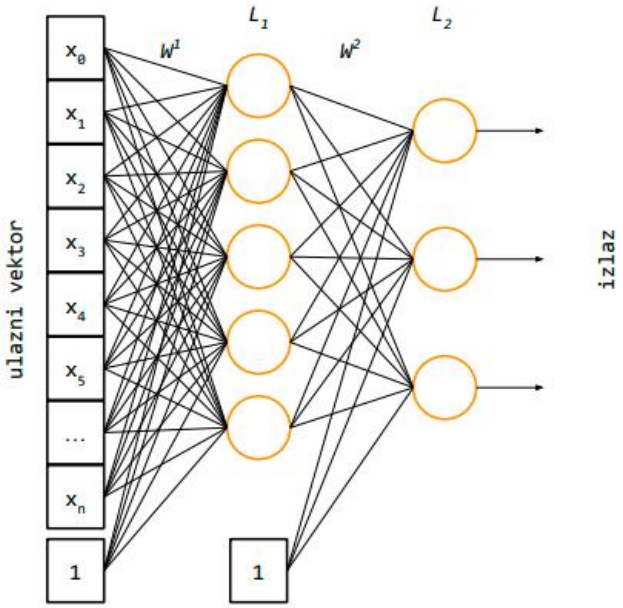
$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives}$$

За главни скуп података користе се колоне укупан сентимент, претходни тренд и будући тренд. Скуп се дели на тренинг и тест скуп у односу 4:1. У зависности од

метода, применом крос-валидације добија се одговарајућа вредност параметара (*ср или к*). Матрицом конфузије се обједињују добијени подаци са очекиваним и рачуна се тачност применом наведене формуле.

4.2 Неуронска мрежа – вишеслојни перцептрон

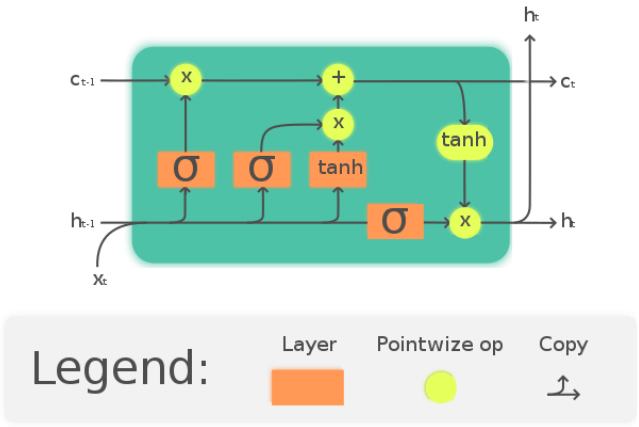
Вишеслојни перцептрон (енг. *Multilayer Perceptron*) је напредна вештачка неуронска мрежа која генерише скуп излаза из скупа улаза. Вишеслојни перцептрон карактерише више слојева улазних чворова повезаних као усмерени граф између улазног и излазног слоја - више појединачних неурона се групише у слојеве, док се излази једног слоја користе као улази за следећи – због тога се овај тип мреже још назива и feed-forward мрежа. Овај перцептрон користи *backpropagation* за обуку мреже и он је метода дубоког учења (енг. *deep learning*). Вишеслојни перцептрон се широко користи за решавање проблема који захтевају супервизовно учење, као и истраживање рачунске неуронауке и паралелно дистрибуиране обраде. Апликације укључују препознавање говора или слике и машински превод.



Слика 1 - Пример вишеслојног перцептрона са једним скривеним и једним излазним слојем

4.3 Рекурентна неуронска мрежа ЛСТМ

ЛСТМ (енг. *Long short-term memory*) је вештачка рекурентна неуронска мрежа која се користи у области дубоког учења. За разлику од стандардне *feedforward* неуронске мреже, наведена користи повратну спрегу. Осим процесирања појединачних података помоћу ње је могуће процесирати и читаве секвенце података. LSTM се користи у предикцији заснованој на временским серијама [4].



Слика 2 - Пример рекурентне неуронске мреже

Код помоћу којег је извршено тестирање података је приказано испод.

```
# punimo ulazni i izlazni trening set
for i in range(0, len(train_data)-24):
    x_train.append(train_data[i:i+24, 0:2])
    y_train.append(train_data[i+24, 1])

# LSTM model kome dodajemo 2 LSTM sloja i 2 Dense sloja
model=Sequential()
model.add(LSTM(50, return_sequences=True,
input_shape=(x_train.shape[1],2)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

#Kompajliranje modela
model.compile(optimizer='adam', loss='mean_squared_error')

# pravimo test data set pa ga delimo na ulazni i izlazni set
test_data = scaled_data[training_data_len - 24: , :]
x_test = []
y_test = dataset[training_data_len: , 1 ]
for i in range(24, len(test_data)):
    x_test.append(test_data[i-24:i, 0:2])

# predikcija
predictions = model.predict(x_test)

# racunanje korena srednje kvadratne greske (root mean square error)
mse = np.sqrt(np.mean(predictions - y_test) **2)
```

5. Предвиђање ценовног тренда – проблем класификације

Коришћењем метода класификације, предвиђа се знак тренда (1, -1 или 0). На основу добијених резултата код обе методе, рачуна се матрица конфузије, затим се на основу погођених вредности израчунава тачност.

5.1 Стабла одлучивања

Тренирањем података за све скуп податке, израчунава се тачност. У табели испод излистан је број тачно погођених вредности, као и укупан број вредности који се предвиђао. Такође, на крају сваке колоне, израчуната је тачност.

Табела 14 - Резултати класификације помоћу методе стабла одлучивања за сва 3 скупа података

	BTC	**XRP**	**ETH**
Број погођених вредности	261	185	293
Укупан број вредности за предвиђање	532	396	549
Тачност	**0.49**	**0.48**	**0.53**

На основу добијених тачности, закључује се да овај метод класификације није дао задовољавајуће резултате, односно успео је да погоди само 50% трендова у сваком скупу података за тестирање. Уколико би се овим резултатом тестирала стратегија куповине и продаје криптовалута, резултати би највероватније дали негативан исход.

5.2 К-најближих суседа

Као и у претходној методи, за сваки скуп података израчунава се тачност, број погођених и број укупних трендова.

Табела 15 – Резултати класификације помоћу метода к-најближих суседа за сва 3 скупа података

	BTC	**XRP**	**ETH**
Број погођених вредности	262	181	295
Укупан број вредности за предвиђање	532	396	549
Тачност	**0.49**	**0.46**	**0.54**

Слична ситуација се догодила као и код претходне методе. Резултати су за нијансу бољи код Биткоина и Етеријума, међутим тестирањем стратегије би исход такође био лош.

Овим методама долази се до закључка да методе класификације, односно алгоритми машинског учења као што су стабло одлучивања и к-најближих суседа не представљају адекватне алгоритме за предвиђање тренда криптовалута. Додатним анализама и додавањем улазних вредности добијених на основу анализа би се можда дошло до бољих резултата.

6. Предвиђање ценовног тренда – неуронске мреже

Као што је у прегледу метода наведено, користиће се 2 типа неуронских мрежа. Помоћу вишеслојног перцептрона предвиђаће се знак тренда, док ће се помоћу рекурентне неуронске мреже предвидети сама цена криптовалуте.

6.1 Примена неуронске мреже вишеслојни перцептрон

За потребе пројектног рада се подаци сва три скупа података убацују у вишеслојни перцептрон. Коришћен је *Neuroph Framework* који садржи имплементрацију ове мреже. Почетни скуп података дели се на тренинг и тест скуп у односу 85:15, али је претходно потребно нормализовати податке методом *MaxNormalizer*. Окружење у ком је имплементиран пројекат и поменути фрејмворк је *Apache NetBeans*.

Оптимизацијом и прилагођавањем мреже скупу података, одређене су следеће ставке:

- Број слојева је 3: један улазни, један скривени и један излазни.
- Активациона функција која је најадекватнија за скуп података је подразумевана за ову класу, односно сигмоидна функција (сигмоид). Такође, за овај тип података могуће је користити и функцију хиперболичког тангенса.
- За број неурона скривеног слоја узима се 10.
- Као правило учења користи се *Momentum Backpropagation*, такође имплементиран у оквиру класа.
- Метода евалуације, односно рачунања тачности предикција мреже имплементирана је коришћењем готових метода, односно аутоматским тестирањем над тест скупом података, рачунањем матрице конфузије и тачности мреже. Уколико би се предвиђала конкретна вредност будућег тренда, користила би се средња квадратна грешка (енг. *mean squared error*).
- Из разлога смањења времена „учења“ модела, постављена је граница броја итерација на 25000.

У фази прилагођавања мреже, испробаване су различите комбинације броја скривених слојева са различитим бројевима неурона у оквиру истих. С обзиром на мали број улаза (улазних променљивих) и извршавањима кодова, закључено је да је један скривени слој довољан.

Остали параметри су узети као подразумевани, као што су *Learning Rate = 0.1, Max Error = 0.01* и *Momentum Value = 0.6*. Поставке мреже дате су у исечку кода испод.

```
DataSet[] ttset = dataSet.split(0.85, 0.15);
DataSet trainingSet = ttset[0];
DataSet testSet = ttset[1];

MultiLayerPerceptron
nnet = new
MultiLayerPerceptron(inputCount, 10, outputCount);
BackPropagation
learningRule = nnet.getLearningRule();
learningRule.addListener(this);
learningRule.setMaxIterations(25000);
nnet.learn(trainingSet);
```

6.1.1 Резултати мреже

Пре убацивања података у мрежу, битно је напоменути да је за овај тип мреже излазна варијабла подељена на две могуће вредности: 0 и 1, односно 0 уколико је тренд негативан и 1 уколико је тренд позитиван. Анализом скупова података, до неутралне промене тренда, односно до стагнирања тренда дошло је у занемарљивом броју у односу на позитивне или негативне промене.

Као што је већ речено, за рачунање тачности имплементирана је метода *evaluate* у којој је најбитније навести који се класификациони евалуатор користи, што је у овом случају *Binary Classifier Evaluator*, који вредностима (вероватноћама) већим од 0.5 додељује класу 1, односно позитиван тренд, док у супротном додељује класу 0, односно негативан тренд. Код ове методе приказан је у исечку кода испод.

```
private void evaluate(MultiLayerPerceptron nnet, DataSet testSet) {
        Evaluation evaluation = new Evaluation();
        evaluation.addEvaluator(new ClassifierEvaluator.Binary(0.5));
        evaluation.evaluate(nnet, testSet);
        ClassifierEvaluator evaluator =
evaluation.getEvaluator(ClassifierEvaluator.Binary.class);
        ConfusionMatrix cm = evaluator.getResult();
        ClassificationMetrics[] metrics =
ClassificationMetrics.createFromMatrix(cm);
        ClassificationMetrics.Stats average =
ClassificationMetrics.average(metrics);
        System.out.println("Accuracy: " + average.accuracy);
    }
```

Преглед резултата, односно тачности по скупу података су излистане у следећој табели.

Табела 16 - Тачност учења мреже по скупу података

	BTC	ETH	XRP
Тачност	**0.59**	**0.56**	**0.49**

6.1.2 Анализа резулата

Уколико се погледају добијене тачности модела, односно учења мреже, закључује се да резултати нису у великој мери подобни за позитивне исходе и даље анализе. Као разлог се могу узети разне претпоставке:

- Улазни скупови података немају довољан број инстанци – посматра се мали временски период, па је потребно проширити скупове, односно узети већи временски оквир (нпр. минимум годину дана само за скуп података за тестирање).
- Параметри мреже нису довољно оптимизовани – потребно одрадити вишепараметарску оптимизацију и на основу резултата одабрати најбољи скуп параметара.
- Не посматрају се добре улазне варијабле – променити формуле за израчунавање тренда (повећати или смањити период рачунања) или увести нову формулу за рачунање.

Уколико се ови резултати пореде са резултатима класификационих метода, закључује се да је коришћење неуронских мрежа сигурно бољи избор од алгоритама класификације.

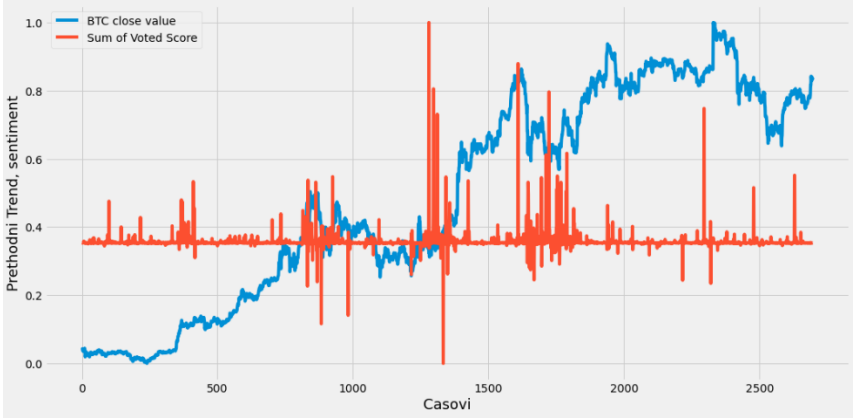
Уколико се резултати тумаче у односу на реалан проблем, закључује се да утицај тржишног сентимента има највећи утицај на кретање тренда Биткоина и Етеријума, док је за Рипл тај утицај видљиво мањи.

6.2 Примена рекурентне неуронске мреже – LSTM

За разлику од вишеслојног перцептрона, у овом случају се на односу цене на затварању и вредности сентимента предвиђа будућа цена коришћених криптовалута.

6.2.1 Предикција цене Биткоина

На графикону испод је плавом бојом представљено кретање цене Биткоина, док је црвеном представљено кретање сентимента тј. „позитивности“ објава на друштвеној мрежи Редит. Може се приметити да постоји добра, али не савршено јасна корелација између ове две вредности. На појединим местима одређени скокови у сентименту су се одразили на то да цена Биткоина расте у наредних неколико дана. Са друге стране, иако је на неким местима дошло до „трзаја“ сентимента, то није утицало на промену цене каква се интуитивно очекивала.



Слика 3 – Графички приказ односа цене Биткоина и тржишног сентимента

У конкретном примеру, ова мрежа предвиђа цену у 24. часу на основу вредности сентимента и цене од нултог до 23. часа. Да би предвидела наредну цену (у 25. часу), мрежа узима вредности сентимента и цене од 1. до 24. часа. Процес се наставља по истом шаблону до последњег часа у скупу података. На тај начин у мрежу улазе само вредности сентимента и цене које су се догодиле у претходном дану јер су оне релевантне за предвиђање вредности у наредном сату.

За тренинг се издвојаја 86% података док се за тест издваја последњих 12%, а преосталих 2% података се изоставља. Расподела има ову структуру због недостајућих вредности које би на резултате мреже негативно утицале.

Користи се Керас библиотека (енг. *Keras*) која ради са вештачким неуронским мрежама. Мрежа садржи два ЛСТМ слоја од по 50 неурона, где први слој има повратну спрегу, тј. враћа вредности последњег излаза. На ова два слоја надовезују се густи слојеви (енг. *dense layers*) са по 25, тј. једним неуроном. За компајлирање модела

користили смо оптимизатор под називом *adam,* а за рачунање одступања предикције од стварне вредности одабрана је средња квадратна грешка (енг. *mean squared error*).

За тренирање мреже користе се различите вредности параметра величине броја епоха и количине података која улази у мрежу (енг. *batch size*). У наредној табели су приказане добијене вредности средње квадрате грешке за различите вредности параметара.

Табела 17 – Корен средње квадратне грешке за различити вредности параметара

Epochs	**1**	**10**	**100**
Batch size			
1	270.67	439.97	768.75
10	117.67	40.62	288.21
100	73.62	108.86	149.55

Иако се чини да је најмања средња квадратна грешка 40.62 за вредности параметара 10 и 10, то није истина. Мрежа је тренирана више пута за исте вредности параметара и закључује се да број епоха не утиче знатно на вредности средње квадратне грешке, док је боље да batch size буде 10 или 100.

Са аспекта ефикасности и брзине тренирања, што је *batch size* већи то је извршавање брже зато што ће се мање пута проћи кроз целу мрежу. Са друде стране, пораст броја епоха негативно утиче на брзину тренирања.

Мреже и њихова комплексност знатно могу утицати на предикције, зато смо покушали и са другачијом комбинацијом броја слојева и неурона. У следећој табели се налазе вредности корена средње квадратне грешке неколико модела са параметрима *batch_size=10* и *epochs*=10.

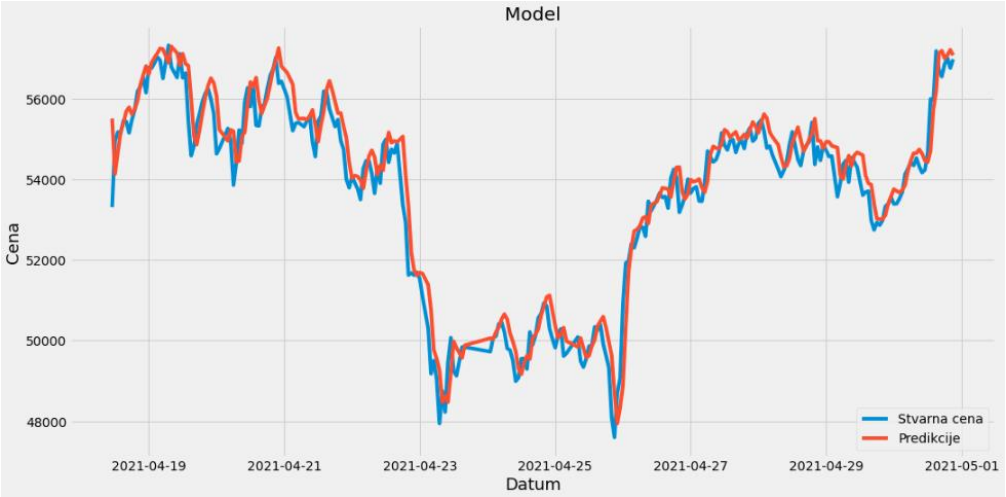
Табела 18 – Корен средње квадратне грешке у односу на број слојева и број неурона

Број слојева и неурона	Корен средње квадратне грешке
LSTM(10), LSTM(10), Dense(10), Dense(1)	628.88
LSTM(25), LSTM(25), Dense(10), Dense(1)	569.46
LSTM(50), LSTM(50), Dense(25), Dense(1)	40.62
LSTM(100), LSTM(100), Dense(50), Dense(1)	108.90
LSTM(200), LSTM(150), Dense(100), Dense(1)	181.18

Примећује се да исувише прости модели могу довести до веће грешке у предвиђању, али исто тако модели могу постати исувише комплексни што доводи до истог проблема.

У претходном тексту, на основу претходна 24 часа се предвиђа наредни (-24 -> +1). Поред тога, испробаване су још неке комбинације:

- Путем последњих 24 часа се предвиђа шта ће се десити наредног дана (тј. шта ће се десити за тачно 24 часа (-24 -> +25.)). Разултат који се добија за средњу квадратну грешку је 1134.04 тј. 892.31 (за параметре 10, 10, тј. 100, 10) што је далеко лошије од претходне комбинације часова.
- Путем последњих 48 часа се предвиђа шта ће се десити наредног часа (-48 -> +1.). Разултат који се добија за средњу квадратну грешку је 253.68 и 27.71 (за параметре 10, 10, тј. 100, 10) што је слично вредностима из табеле.



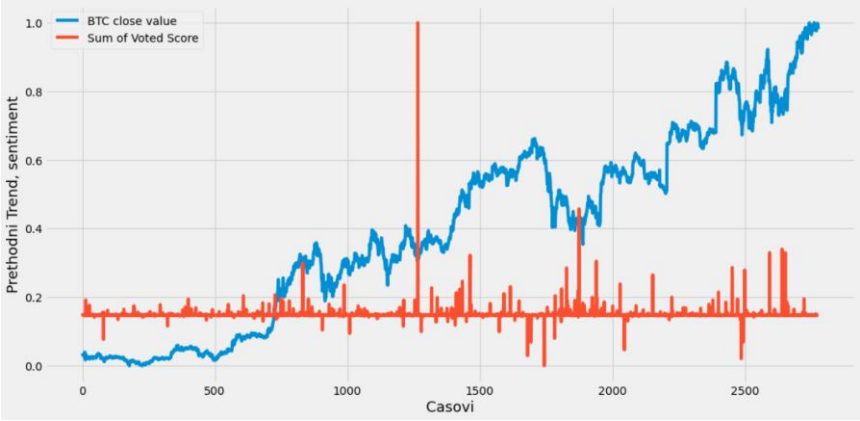
Слика 4 - -24 -> +1 sa parametrima 10, 10

На графикону плава линија представља стварну цену Биткоина, док црвена представља предикције модела. Иако се чини да је модел изузетно добар, заправо није. Предикције константно касне за стварним. Ако би се посматрао један временски тренутак, уочава се да промена предикције стално касни за променом цене. Ако би се направио софтвер за трговање на основу овог модела, готово сигурно би пословао у минусу.

Нажалост, модел који би остварио финансијски успех није креиран. Разлог томе је што претходна цена, која је један од два улаза у наш модел, не утиче на будућу. Са друге стране, сентимент готово сигурно утиче на будуће кретање цене, али није једини фактор. Претпоставка је да велики тржишни „играчи“ (богате корпорације, конгломерати, фондације...) обављају велике трансакције криптовалута које драстично утичу на цену. Такве промене у цени корисници Редита не могу да „испрате“, тј. те промене нису последица писања на овој друштвеној мрежи. Тржиште криптовалута је изузетно волатилно јер је и даље мало и нестабилно, мали број трговаца утиче драстично на кретање цена.

6.2.2 Предикција цене Етеријума

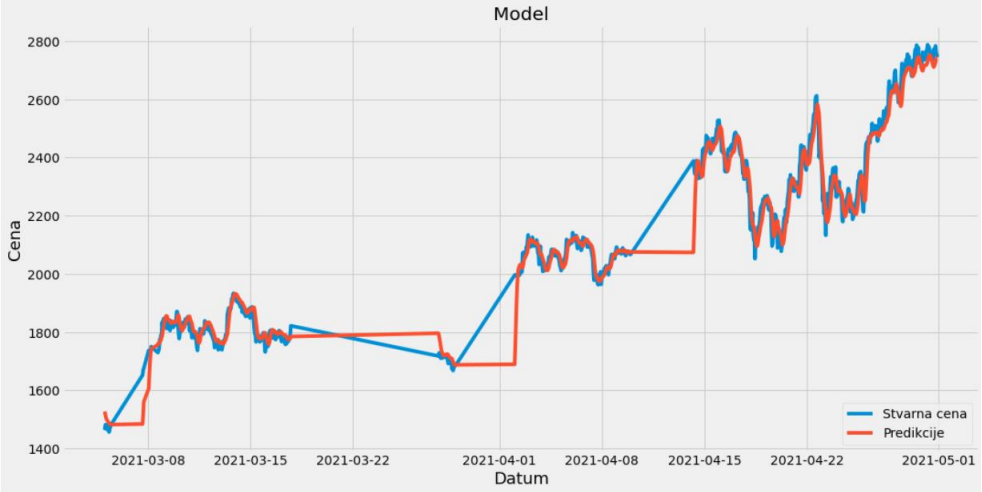
За разлику од односа цена Биткоина и тржишног сентимента за исти, ситуација код Етеријума је знатно другачија, тј. визуелно је немогуће приказати сам однос.



Слика 5 – Графички приказ односа цене Етеријума и тржишног сентимента

За Етеријум је скуп података подељен на 70% за тренинг и 30% за тест. И поред тога што недостаје доста података, неуронској мрежи то није сметало јер одступање предикције од стварних вредности које имамо готово исто као код Биткоина који има све вредности на тест скупу.

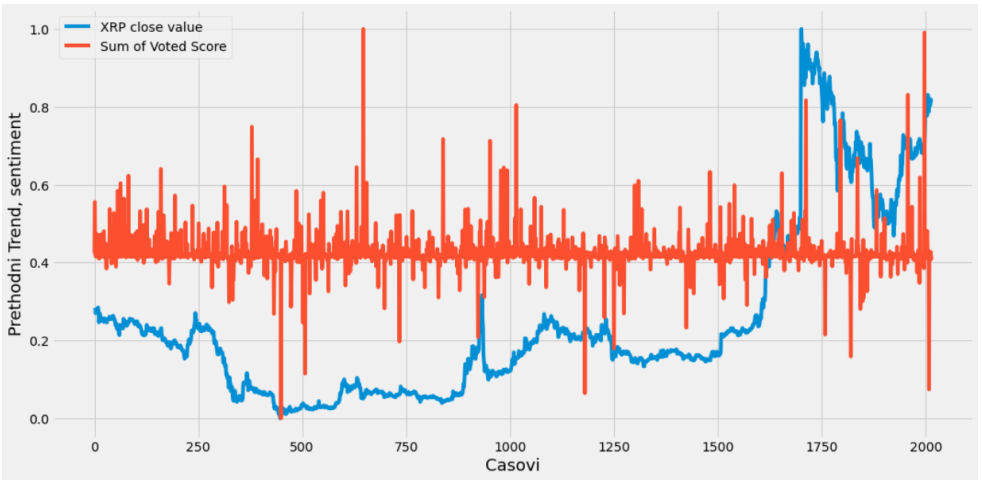
За *batch size* вредност 10 и 10 епоха вредност средње квадратне грешке износи 5.21. Делује да је грешка доста мања од Биткоина, али претпоставља се да је разлог томе што Биткоин има већу цену, па је квадрат разлике између предикције и стварне цене већи него у случају Етеријума.



Слика 6 – Предикција цене Етеријума

6.2.3 Предикција цене Рипла

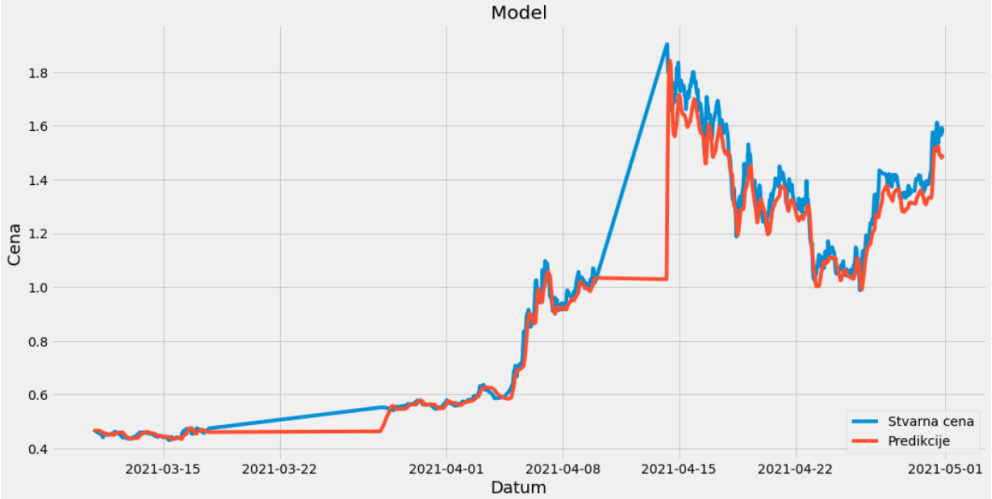
Као и код Етеријума, визуелно се не може уочити значајна корелација између сентимента и цене.



Слика 7 – Графички приказ односа цене Етеријума и тржишног сентимента

За Рипл је скуп података такође подељен на 70% за тренинг и 30% за тест. За исте параметре неуронске мреже као код Етеријума, средња квадратна грешка износи

0.0349. Постоји сумња да је недостатак података који се јавио мало пре 15. априла 2021. у некој малој мери утицао на предикције модела које су се јавиле након тога. Од тог тренутка модел је пратио скокове и падове цене али је константно почео да „потцењује“ (енг. *undershoot*) стварну вредност Рипла.



Слика 8 – Предикција цене Рипла

Графикони корелације сентимента и цене нису наведени за ове две криптовалуте. Разлог томе је то што су ова два графика још мање јасна од графика корелације Биткоина. Претпоставка је да је то зато што су далеко мање криптовалуте од Биткоина, а нарочито Рипл.

7. Унакрсна корелација промена цене и тржишног сентимента

С обзиром да предвиђања и тачности нису задовољиле критеријум „довољно добрих резултата“, разматрана је корелација између промена цене криптовалуте (разлика цене на затварању и отварању) и тржишног сентимента. У овом случају, разматра се веза између временске серије и децималне вредности.

Визуелно се ове променљиве представљају на два одвојена графикона уз помоћ Пајтон библиотека, док се уз помоћ одређених метода рачуна вредност корелације вредност кашњења (енг. *lag*). Кашњење у овом случају представља помак сентимента, односно тренутак када сентимент заправо утиче на промену цене. Кашњење се заправоодноси на то колико су серије померене и знак кашњења одређује која се серија помера. Вредност кашњења са највишим коефицијентом корелације представља најбоље прилагођавање две серије.

Учитавају се за сваку криптовалуту по два фајла – вредности промена цена и вредности сентимента од прве до последње опсервације. Начин учитавања приказан је у исечку кода.

```
time_series = ['btc', 'btcsent']
dirName = "data/"
fs = 2680
MR = len(time_series)
Y = np.zeros((MR, fs))
dictVals = {}
for ind, series in enumerate(time_series):
    filename = dirName + series + ".txt"
    df = pd.read_csv(filename, names=[
                    'time', 'U'], skiprows=1, delimiter='\s+')
    yvalues = []
    for i in range(1, fs+1):
        val = df.loc[df['time'] == i]['U'].values[0]
        yvalues.append(val)

    dictVals[time_series[ind]] = yvalues
```

Затим се на два одвојена графикона исцртавају ове вредности.

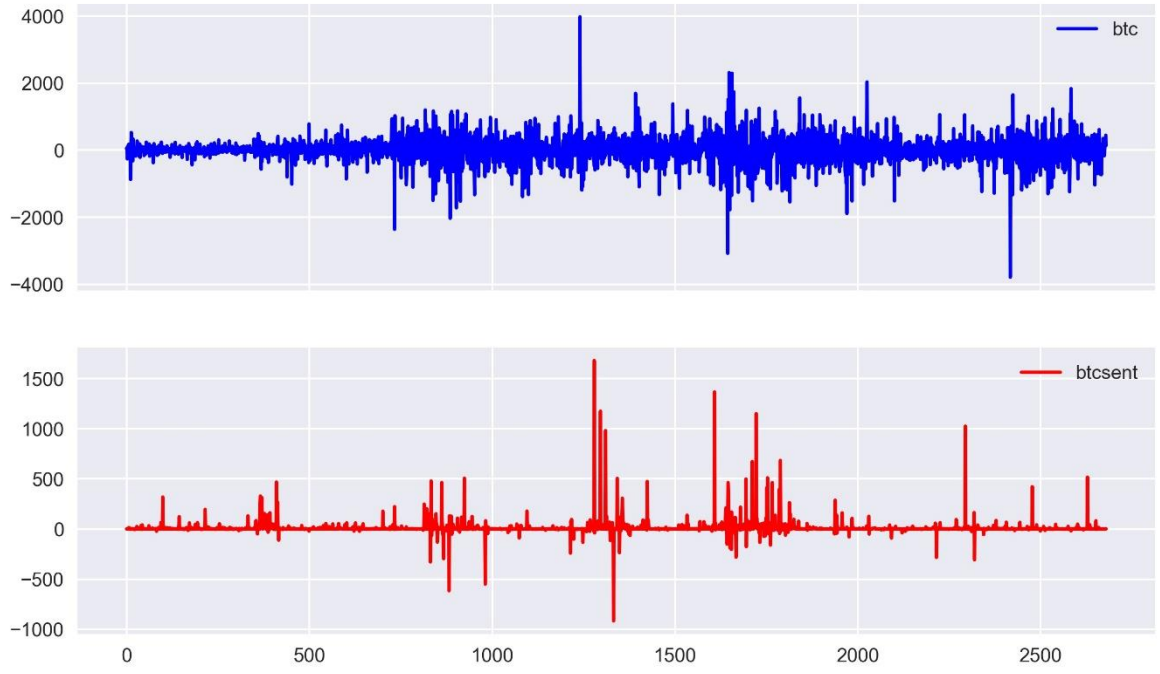
```
timeSeriesDf = pd.DataFrame(dictVals)
fig, ax = plt.subplots(2, 1, figsize=(10, 6), sharex=True)
ax[0].plot(timeSeriesDf[time_series[0]], color='b', label=time_series[0])
ax[0].legend()
ax[1].plot(timeSeriesDf[time_series[1]], color='r', label=time_series[1])
ax[1].legend()
plt.savefig('data_viz.jpg', dpi=300, bbox_inches='tight')
```

Рачун саме корелације и израчунавање кашњења приказано је испод:

```
def crosscorr(datax, datay, lag=0):
    return datax.corr(datay.shift(lag))
d1, d2 = timeSeriesDf[time_series[0]], timeSeriesDf[time_series[1]]
window = 10
lags = np.arange(-(100), (100), 1)  # contrained
rs = np.nan_to_num([crosscorr(d1, d2, lag) for lag in lags])

print("xcorr {}-{}".format(time_series[0], time_series[1]),
lags[np.argmax(rs)], np.max(rs))
```

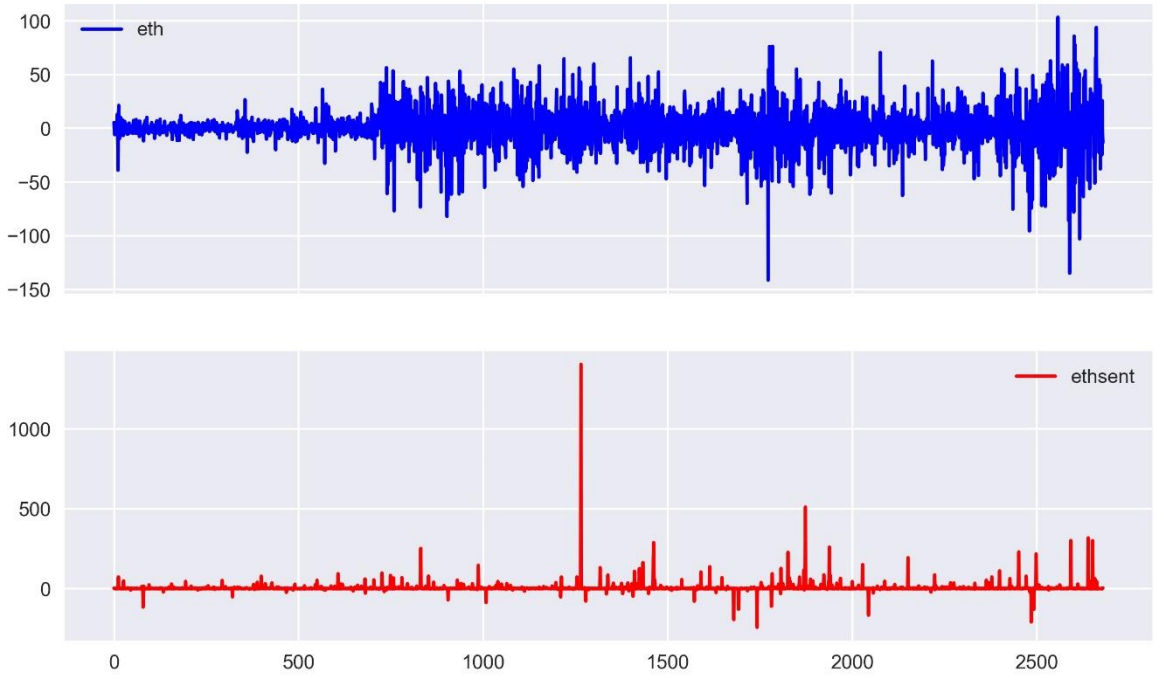
Код Биткоина, уочава се „визуелна“ корелација између промена цене и сентимента, односно примећују се позитивни и негативни скокови како промене цене, тако и сентимента.



Слика 9 – Упоредни приказ промене цене и тржишног сентимента Биткоина

Што се тиче корелације, она износи приближно 0.07, што за овај скуп од 2500 опсервација представља јако слабу корелисаност. Вредност кашњења је 66, односно значи да сентимент касни 66 сати у најбољој корелацији са променом цене.

Етеријума је значајно лошија ситуација што се може запазити на слици испод. Промене у ценама су биле честе и са великим скоковима и падовима, док је ситуација са сентиментом по том питању другачија, односно „визуелна“ корелација ова два параметра је врло слаба.

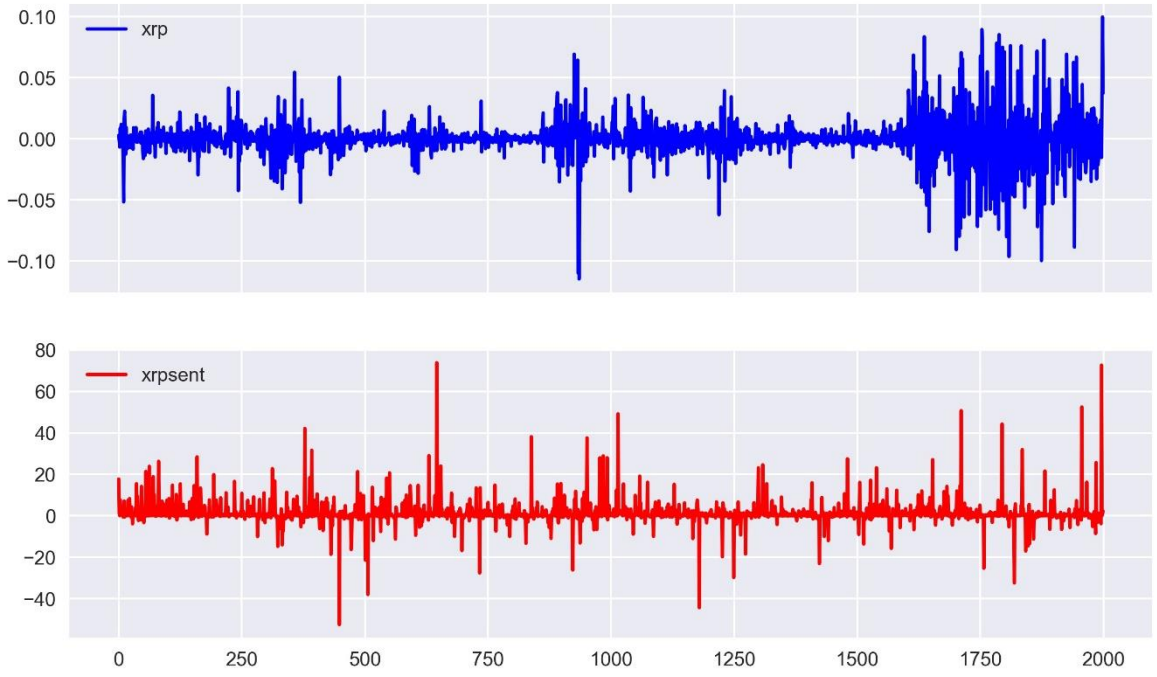


Слика 10 – Упоредни приказ промене цене и тржишног сентимента Етеријума

Вредност корелације је мања од 0.05, док је вредност кашњења -1336.

У односу на Биткоин и Етеријум, закључило би се да је корелација промене цене и сентимента код Рипла најбоља. Међутим, вредности параметара то не показују тако. Вредност корелације износи нешто мање од 0.06, док је вредност кашњења 786.

На основу унакрсне корелације, закључује се да се највећи утицај сентимента на промену вредности цене, односно разлику цена на затварању и отварању јавља код Биткоина. Ова чињеница и није изненадна, јер Биткоин важи за најпопуларнију, а уједно представља и најстарију крипто валуту, па је утицај „људи“ са друштвених мрежа знатно већи.



Слика 11 – Упоредни приказ промене цене и тржишног сентимента Рипла

8. Подаци са осталих друштвених мрежа

У фази прикупљања података, подаци који су били потребни за даљи рад су се преузимали са друштвених мрежа Редит (енг. *Reddit*) і Твитер (енг. *Twitter*). Разним алгоритмима покушано је да се дође до валидних података са обе друштвене мреже.

Проблем у раду настао је код прикупљених података са Твитера. Подаци су прикупљани на 2 начина: директно са Твитер профила и коришћењем *Selenium-a* – библиотеке програмског језика Пајтон која омогућава динамичко кретање кроз веб апликацију и директан приступ „живим“ подацима, али и преко Твипаја (енг. *Tweepy*) – библиотеке програмског језика Пајтон која омогућава приступ Твитеровом АПИ-ју. Анализом добијених података, закључено је да њихов број није довољан за даљи рад, односно резултати који би се добили даљим методама не би били валидни, због великих „јазова“ у самој структури података. Побољшавањем постојећих или креирањем нових алгоритама, могуће је доћи до новог (већег) броја података са ове друштвене мреже, што би самом пројектном раду дало на већем значају.

У овом примеру смо покушали то да постигнемо путем официјалног Твитеровог АПИ-ја користећи библиотеку под називом Твипај. Нажалост, скрипта је преузимала само око 60 твитова који су се класификовали као „популарни“ (велики број свиђања и пратилаца). Претпоставља се да је разлог томе што Твитер ограничава количину твитова коју је могуће скинути. Одлучено је да се овај метод изостави из пројекта због недовољне количине твитова.

Као и за податке са Редита, креиран је модел сентимента, односно формула која би на основу одређеног типа података креирала укупни сентимент скор, односно значајност ставова Твитер корисника на актуелна дешавања са ове друштвене мреже.

8.1 Сентимент анализа за Твитер објаве

У оквиру овог потпоглавља, објашњен је модел сентимента који је био одабран за објаве са друштвене мреже Твитер. С обзиром да код Твитера уместо позитивних и негативних гласова постоје лајкови и ритвитови (енг. *Retweet*), ове две варијабле би биле укључене у сентимент анализу Твитер објава. Такође, овај модел би се разликовао од модела за Редит објаве по варијабли број пратилаца корисника који је

поставио конкретну објаву [2]. Међутим, Твитер ограничава преузимање ове врсте података, стога се наилазило на препреке у вези са овом варијаблом.

Наиме, број пратилаца корисника за чије објаве се рачуна сентимент је веома битан фактор. Он указује на утицајну моћ корисника и досег његових објава, што додаје тежину објавама утицајних корисника.

Сентимент објава израчунат је на исти начин као и за Редит објаве, а то је коришћењем VADER лексикона. Када су израчунати сентименти објава, колоне које су нам биле потребне за модел сентимента су:

- Сложени сентимент: *Compound Score*;
- Број пратилаца корисника: *UserFollowerCount*;
- Број лајкова: *Likes*;
- Број ритвитова: *Retweets*:

На основу датих ознака, формула за рачунање коначног (тежинског) сентимента је [1]:

$$FinalScore = CompoundScore \times UserFollowerCount \times (Likes + 1) \times (Retweets + 1)$$

Након израчунатог сентимента, по истом принципу као и за Редит објаве, скупови података су спојени у јединствени скуп и затим су подаци груписани по сатима како би се припремила табела за неуронску мрежу.

Међутим, у овом тренутку је постало јасно да постоји велики број недостајућих вредности по сатима и такви подаци нису могли бити коришћени за предвиђање тренда криптовалута. Разлог за велики број недостајућих вредности је тај што подаци са Редита и Твитера нису преузимани на исти начин, а простим филтрирањем објава у току дана према броју лајкова и ритвитова не даје равномерно распоређене податке у току дана.

Начин на који се рачунала промељива *FinalScore* приказана је у исечку кода испод.

```
while i < 3023:
    compound_score = 0
    final_score = 0
    text = btc_tweettext[i]
    likes = btc_likes[i]
    retweets = btc_retweets[i]
    #followers = btc_followers[i]
    compound_score = sid.polarity_scores(text)['compound']
    final_score = compound_score * 1 * (likes + 1) * (retweets + 1)
    if final_score >= 0:
        final_sentiments[i] = math.sqrt(final_score)
    else:
        final_sentiments[i] = -1 * math.sqrt(abs(final_score))
    compounds_sentiments[i] = compound_score
    i+=1
df1['Compound']=compounds_sentiments
df1['Final']=final_sentiments
#Dodavanje sentimenta svakom postu
j = 0
while j < 3023:
    try:
        positive_counter = 0
        negative_counter = 0
        text = df1.TweetText[j]
        words_text = findall(r"[\w']+", text)
        for word in words_text:
            for positive_word in x:
                if positive_word == word.lower():
                    positive_counter = positive_counter + 1
                    break
            for negative_word in y:
                if negative_word == word.lower():
                    negative_counter = negative_counter + 1
                    break
        positive_sentiments[j] = positive_counter
        negative_sentiments[j] = negative_counter
        sentiment_scores[j] = positive_counter - negative_counter
        j +=1
    except:
        break
```

9. Закључак

Овим пројектом показује се целокупан процес предвиђања одређене врсте података – од самог прикупљања до анализе добијених резултата. Да би резултати били што боље протумачени, коришћене су различите методе и метрике зарад добијања адекватних одговора.

Анализом тржишног сентимента, успешно је додељена вредност једном периоду криптовалуте – 1 час. Овај моменат представља јачину тржишног опхођења према одређеној криптовалути – њихова мишљења, реакције, запажања и одлуке. Применама методе вештачке интелигенције и машинског учења, ови подаци су постали мерљиви и употребљиви за даљи рад.

Неуронске мреже су коришћене зарад тренирања и тестирања одређеног скупа података. Овим поступком предвиђао се ценовни тренд криптовалута и сама цена. Анализом добијених резултата запажа се слаба веза између сентимента и тренда свих криптовалута које су обухваћене овим пројектом. Због чега? Списак могућих разлога је дужи, али неки од њих су „више вероватни“:

- Мали временски период – мали број опсервација
- Постојање бољих метода за саму анализу резултата – средња квадратна грешка и тачност се могу заменити неким другим метрикама
- Сентимент са другог тржишта – као што је у раду напоменуто и прикупљање података са Твитера, можда се из тога може закључити боља повезаност сентимента тог тржишта у односу на тржиште Редита
- Подаци са тржишта нису валидни – прикупљени подаци могу бити „лажни“, тј. „власници“ објава нису довољно стручни у овој области

Могуће промене и побољшања се могу добити применом других неуронских мрежа, повећањем скупова података, анализом других криптовалута као што су *Dogecoin, Harmony, Litecoin* и остале.

Генерални закључак и утисак јесте да повезаност друштвених мрежа и вредности криптовалута сигурно постоји, само је потребно ухватити прави начин за њихово откривање.

10. Литература

[1] Gui Jr, H. (2019). Stock Prediction Based on Social Media Data via Sentiment Analysis: a Study on Reddit (Master's thesis).

[2] Mohapatra, S., Ahmed, N., & Alencar, P. (2019, December). KryptoOracle: A Real-Time Cryptocurrency Price Prediction Platform Using Twitter Sentiments. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 5544-5551). IEEE.

[3] Matija Milekić, Aleksandar Rakićević, Pavle Milošević (2018) Neural networks in market sentiment analysis for automated trading: The case of Bitcoin. In Symposium proceedings-XVI International symposium Symorg 2018: Doing Business in the Digital Age: Challenges, Approaches and Solutions.

[4] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014.

Коришћени сајтови:

1. https://www.cryptodatadownload.com/data/binance/
2. https://towardsdatascience.com/computing-cross-correlation-between-geophysical-time-series-488642be7bf0
3. http://ai.fon.bg.ac.rs/