Faculty of Mathematics and Information Science
Warsaw University of Technology

Project Milestone 4

Group name: Crypto Busters

Subject: Big Data Analytics

Authors:
Dominika Umiastowska, Laura Bąkała, Mateusz Wójcik, Patryk Wrona

Warsaw
2022/2023

Contents

1. Introduction

The project's goal is to model different cryptocurrency prices based on sentiment analysis of Reddit posts using Big Data solutions.

1.1. Benefits from Business Perspective

The business user will be able to tell if given cryptocurrency prices depend on the sentiments of Reddit users concerning respective cryptocurrencies. The dependence is still to be investigated – perhaps the fact of fluctuations in the sentiment of Reddit posts could lead to an increase or decrease in the value of analyzed cryptocurrency. The main benefit for the business user will potentially be the possibility to obtain a return on investment on cryptocurrencies as well as to make a profit from it.

1.2. Functional Requirements

The user should be able to:

— analyze the trends of cryptocurrency prices,
— analyze the sentiment of Reddit posts for a given cryptocurrency,
— obtain predictions of a cryptocurrency price based on Reddit sentiment analysis, i.e., whether the price is going to increase or decrease in the near future,
— assess model accuracy on historical data, making price predictions for a past date,
— export the prices of a selected cryptocurrency within a specified time period

1.3. Non-functional Requirements

— The price should be available for the user within half an hour of publishing it in the source API.
— The dashboard should handle at least 20 concurrent users without noticeable loss of performance.
— Security of shared responsibility model offered by Google Cloud Platform. Google is responsible for network, storage, encryption, and physical security. The team will be responsible for the security of the data pipeline and the application.

— Scalability – GCP offers a platform for building data ingestion and processing pipelines to support a wide range of streaming, batch, and near-real-time data sources. To ingest data reliably, serverless auto-scaling Google Pub/Sub service was used. Additionally, Cloud Dataproc provides a serverless Apache Spark integration for the scalable processing of terabytes of data.

— Fast recovery without disruption can be achieved by using instance templates for Compute Engines. Additionally, Cloud Storage was used for backups.

2. Data Sources

2.1. Cryptocurrency Prices

The analyses of cryptocurrencies were limited to 6 chosen cryptocurrencies, but other cryptocurrencies are also accessible in the storage:

— Bitcoin
— Ethereum
— Dogecoin
— Cardano
— XRP
— Solana

2.1.1. Real-time Cryptocurrency Prices

The streaming data API [Coia] provides the user with real-time selected cryptocurrency prices. The list of available cryptocurrency identificators [Coib] used in API is publicly available.

The streaming data consists of the following:

— **product_id** [Coib] – cryptocurrency identificator
— **cryptocurrency price**
— **timestamp**

The data can be accessed via WebSocket protocol; after subscribing for a given product_id, its price is sent to the subscriber.

Analyzed cryptocurrencies have the following product_ids (depending on the currency, USD or EUR):

— **Bitcoin** – BTC-USD, BTC-EUR
— **Ethereum** – ETH-USD, ETH-EUR
— **Dogecoin** – DOGE-USD, DOGE-EUR
— **Cardano** – ADA-USD, ADA-EUR
— **XRP** – XRP-USD, XRP-EUR
— **Solana** – SOL-USD, SOL-EUR

Data Format

The data is transmitted in JSON format. Example of such JSON:

```
{
  "type": "l2update",
  "product_id": "BTC-USD",
  "changes": [
    [
      "side": "buy",
      "price": "22356.270000",
      "size": "0.00000000"
    ],
    [
      "side": "buy",
      "price": "22356.300000",
      "size": "1.00000000"
    ]
  ],
  "time": "2022-08-04T15:25:05.010758Z"
}
```

All values are strings. Important fields are in **bold**:

— type – channel used by API, 'level2', 'level2_batch' (to reduce traffic), used by WebSocket protocol's subscribe message
— **product_id** – API special identificator; reference to cryptocurrency and currency, in this case, bitcoin and united states dollar
— changes - an array of changes at a given timestamp
— **side** – 'buy' or 'sell' for a given cryptocurrency
— **price** – exchange rate of given cryptocurrency-currency
— size – the amount transferred on the last trade
— **time** – timestamp

API Limitations

WebSocket connections are rate-limited at **8 requests every second per IP and up to 20 requests for bursts.**. Depending on the channel, the frequency of incoming data may vary, e.g., 20 requests per second are to be expected on *level2_batch* channel.

2.1.2. Historical Cryptocurrency Prices

To obtain historical cryptocurrency price datasets across the years, we used another API named CoinGecko API [Coic], which is public and free of charge.

We used it to extract datasets for each cryptocurrency with two columns: **date** and **price** in USD. The interval between each entry is one day, where prices are noted at 00:00 UTC. In this way, we obtained datasets:

— **bitcoin_data** - 3455 entries, from 28.04.2013 to 13.10.2022,
— **cardano_data** - 1823 entries, from 18.10.2017 to 13.10.2022,
— **dogecoin_data** - 3223 entries, from 15.12.2013 to 13.10.2022,
— **ethereum_data** - 2625 entries, from 07.08.2015 to 13.10.2022,
— **solana_data** - 917 entries, from 11.04.2020 to 13.10.2022,
— **xrp_data** - 3354 entries, from 04.08.2013 to 13.10.2022.

The overall size of the data is 460 KB.

2.2. Reddit Data

2.2.1. Reddit Real-Time Submissions

Reddit real-time streaming data has been collected via HTTP GET requests to the subreddit URL [Official Reddit API].

Data Format

The data is transmitted in JSON format.

```
{
"kind": "string",
"data": {
    "modhash": "string",
    "dist": int,
    "children": [{
        "kind": "string",
        "data": {
            "approved_at_utc":"string",
            "subreddit": "string",
            "selftext": "string",
            "title": "string",
            ...,
            "is_video":"boolean"
```

```
            }],
        "after":"",
        "before:""
        }
    }
```

All the data of the output JSON file is given in `'children'` object of the `'data'` key. There are about 110 keys with attributes of each data point, for example `'title'`, `'author_fullname'`, `'url'`, `'score'`, `'num_comments'`, `'children'`, `"created_utc"` and the most important one, `'selftext'`.

There are 6 types of `"kind"` attribute, namely:

— **t1** – comment
— **t2** – account
— **t3** – link
— **t4** – message
— **t5** – subreddit
— **t6** – award

API Limitations

The API can be used with the need for authentication (*OAuth2*). There are also traffic rate limitations, i.e., clients can make up to 60 requests per minute. Reddit API is free of charge if the developed code is made open-source. Moreover, one must state in the application's documentation how you collect, use, store, and disclose data collected by using the API.

Using the API

The data can also be collected by using official wrappers or libraries developed in programming languages, for example, PRAW in Python. PRAW has full access to the API. The example HTTP GET request is presented as follows.

```
https://www.reddit.com/r/{subreddit}/{listing}.json?limit={count}&t={timeframe}
```

2.2.2. Historical Reddit Submissions

Archives of historical Reddit submissions[stu] were downloaded and preprocessed with the use of Big Data solutions, and for financial causes, only the data from year 2022 was stored in HDFS. Exemplary filtering scripts could be found in repository[Bus22] in the data acquisition directory.

The filtered data is in tabular form; exemplary filtering in figure 2.1

The columns shown in 2.1 are:

— **subreddit** – Reddit 'topic', in our case e.g. 'bitcoin'

	subreddit	title	selftext	created_utc
0	ethereum	The DAO Is Almost Totally Apathetic: It Should...		1464739795
1	ethereum	GOLDMAN SACHS: 5 practical uses for blockchain...		1464744052
2	ethereum	Microsoft to develop an open source platform a...		1464745936
3	ethereum	Need help buying ethereum	[removed]	1464748845
4	dogecoin	So who approved Dodge to be in an Optimum come...		1464749216
5	ethereum	Difference between tokens and balances inside ...	So I bought some DAO tokens, and set up both c...	1464752690
6	ethereum	Started reading the "Blockchain Revolution" bo...	First I wanted to state that this is not an en...	1464753826
7	ethereum	Buying coca leaves through Ethereum		1464755277
8	dogecoin	Ha I just realized I was putting an I at the e...	[Oy.](http://i.imgur.com/GqKZ1eH.jpg)	1464758412
9	ethereum	cant find alethone in the ethereum webthree um...	[deleted]	1464761853
10	ethereum	Need help with my Wallet, balance becomes zero	It's been awhile since I last open Mist, now I...	1464764525

Figure 2.1. Exemplary historical Reddit submissions

— **title** – post's title
— **selftext** – post's content
— **created_utc** – timestamp in ms

Today, all Reddit submissions' volume is 5 TB, but taking into account only cryptocurrency *subreddits*, the data size is 500 MB.

Acquired Data Time Interval

Due to limitations on the collected data (subscription costs on Google Cloud) but also because of changing nature of cryptocurrency opinion, we stored historical Reddit data since January 2022. The final (considered at the project delivery date) raw data volume is about 500 GB.

2.3. Historical Sentiment Data

We've found multiple sources of historical sentiment data for various companies present in the stock market, all in batch format.

2.3.1. Stock-Market Sentiment Dataset

One such data is a dataset available on Kaggle named "Stock-Market Sentiment Dataset" [Cha20]. The data was collected from several (non-specified) Twitter handles and labeled. 5791 tweets were harvested in that process. They are stored in a single CSV file with two columns: **Text** and **Sentiment**. The former contains the tweet text, unmodified; no duplicates were found. The latter has two possible values, -1 and 1, corresponding to a negative and positive sentiment

respectively. These two classes are not balanced since there are 3685 positive tweets and 2106 (i.e. about 43% less) negative ones.

We've analyzed tweet lengths to see if there are any differences in tweet lengths (Fig. 2.2). Except for the one bar for negative tweets of a length of about 140, the distribution is basically the same, meaning that there are no statistically important differences in tweet lengths.

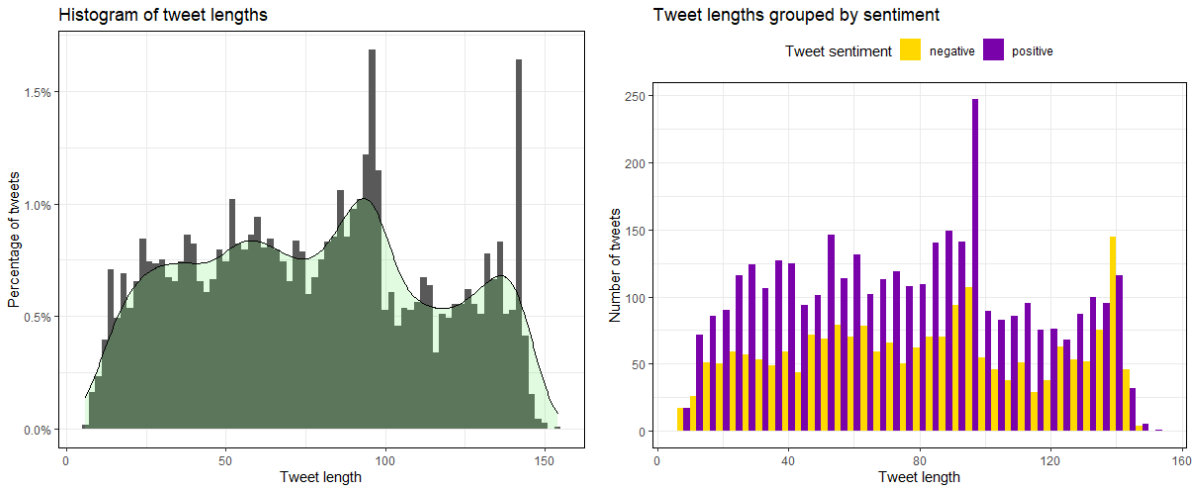

Figure 2.2. Distribution of tweet lengths

This dataset has two significant drawbacks. One is that there are no timestamps, making the tweets impossible to place on the timeline. The other is that the companies are only mentioned by their stock codes, which have to be isolated from the tweet text. This makes using this dataset rather inconvenient.

2.3.2. Stock Sentiment Analysis Dataset

The next dataset was published on GitHub by Surge AI [Sur22b]. The data was, again, collected from Twitter handles and labeled. This one dataset, though, is merely a sample with 500 observations. The strength of this dataset comes from its additional columns (compared to the Kaggle dataset). Bar the **Tweet Text** and **Sentiment** variables, the URL source of the tweet is stored in **Tweet URL** column, which may allow us to scrape the date of posting. Not only that, but a **Stock Ticker** is included too, meaning that all the relevant data is already extracted, rendering tweet text redundant.

The sentiment seem to have no effect on tweet lengths, again (Fig. 2.3). The tweets are significantly longer than those in the Kaggle dataset, however.

We've also analyzed stock tickers, also grouping them by sentiment (Fig. 2.4). Few stocks deviate from the average by much; with such low numbers, it's hard to draw meaningful conclusions about stock value. "$GME" and "$AUVI" have the highest fraction of positive tweets, while "$WMT" has the lowest... among the most commonly commented stocks, since many stocks have only one or two related tweets.

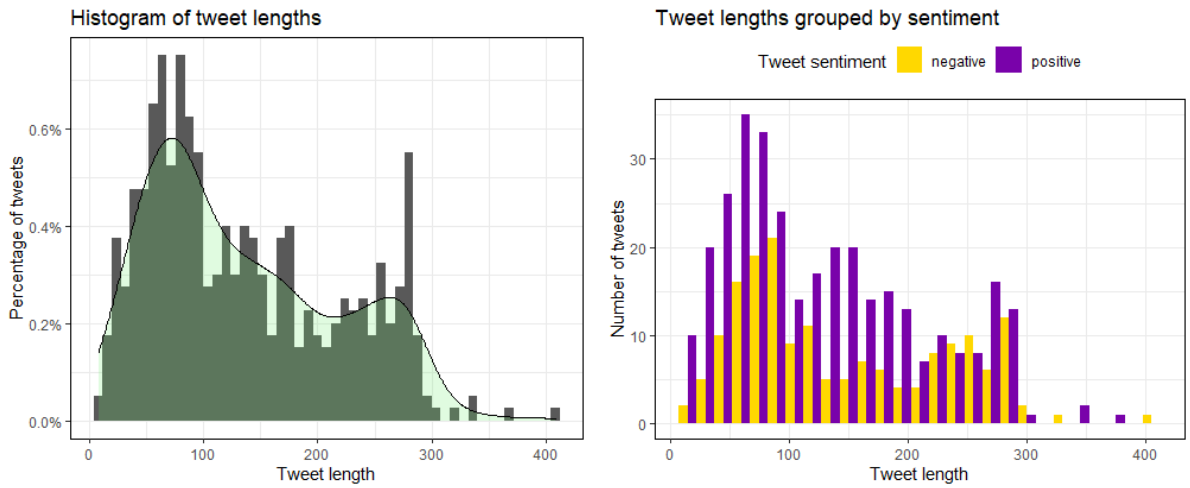

Figure 2.3. Distribution of tweet lengths

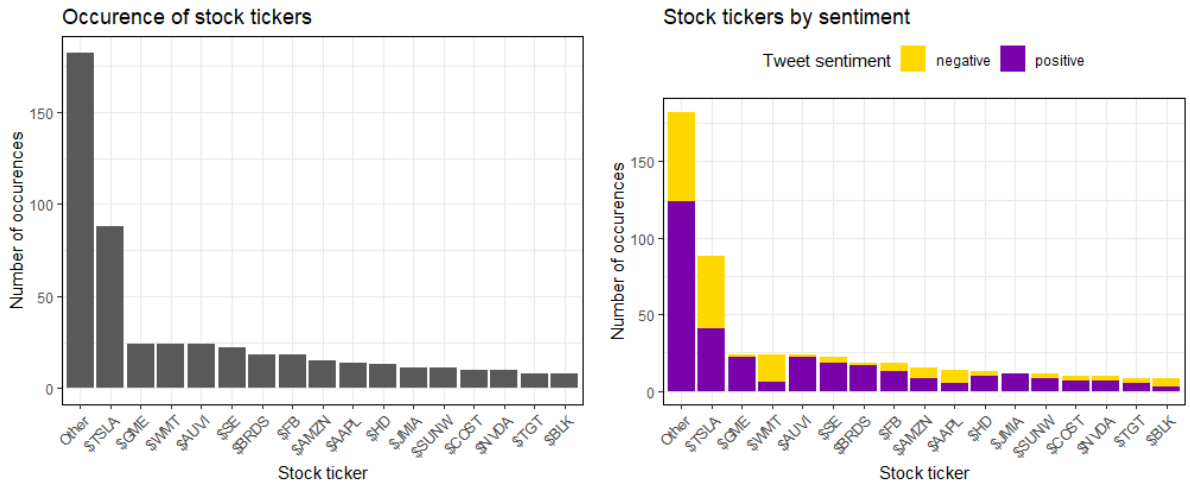

Figure 2.4. Distribution of stock tickers

2.3.3. Crypto Sentiment Analysis Dataset

Finally, there's a third dataset, again published on GitHub by Surge AI [Sur22a], which differs from the previous data in that its source is Reddit comments and the stocks are limited to cryptocurrencies. The data is rather small, with 562 observations. There are three columns, **Comment Text**, **Sentiment**, and **URL**. What is missing, though, is the currency name the comment is about. Sometimes the currency can be inferred from the subreddit name (`r/Bitcoin`, `r/dogecoin`, `r/solana`...). Sometimes currency name is mentioned in the comment. Some comments, however, are neither of those (e.g. "good time to accumulate"). It would take time to provide exact numbers, but a quick inspection shows that a significant portion of comments falls in the last group, which would further reduce the amount of available data.

If all of these fail, manual scrapping of Reddit or Twitter is also a possibility, but it would

involve setting up our own sentiment analysis pipeline, which is by far the most consuming operation here.

2.4. Data Quality Summary

The data concerning cryptocurrency seems the cleanest of all data, whereas any text-related data source must be cleaned before one can work with it. Especially Reddit submissions concerning cryptocurrency are messy – there are plenty of posts where people post the code to investing APIs, asking for help. On the other hand, there are plenty of symbols and emoticons. All this text needed to be filtered out to contain only English words used later in NLP models.

2.5. Format of Source Data

The source data is stored or arrives in given formats (if it was not stated before in this document, we point it out here):

— Historical Reddit – JSON format in .zstd archives
— Historical Crypto – JSON format
— Sentiment Data – CSV format
— Real-time Reddit – JSON format
— Real-time Crypto – JSON format

2.6. Purpose of Each Dataset in Machine-Learning Tasks

The **sentiment dataset** was used to learn the classifier – that is based on predictor 'Text' (which was preprocessed using only English words and then tokenized), the aim of the classifier would be predicting target variable 'Sentiment'. Such a classifier was used in the future on Reddit submissions to augment the prediction level of regressors for predicting cryptocurrency prices.

The column of **cryptocurrency datasets** 'price' is of main interest; based on past prices and Reddit submissions, the model of regression was built to predict cryptocurrency prices in the near future.

As far as **reddit data** is concerned, the main concern was to clean the post's contents (column 'selftext') as well as the post's title (column 'title'). The data itselfwas auxiliary in predicting crypto prices and served for analyzing the influence of Reddit posts on these prices.

3. Exploratory Data Analysis

The data collected can be divided into three main categories, according to their roles in the project and their source: Reddit posts from cryptocurrency subreddits, cryptocurrency stock data, and economy-related texts with sentiment labels.

3.1. Reddit data

Except for XRP, there isn't too much difference in the amount of posts from each cryptocurrency-focused subreddit. The most active r/Bitcoin subreddit only has about twice as much posts in the analysed period (year 2022) as the second-to-last subreddit, r/solana. This is good for training models, as the imbalance of the amount of data between models is negligible.

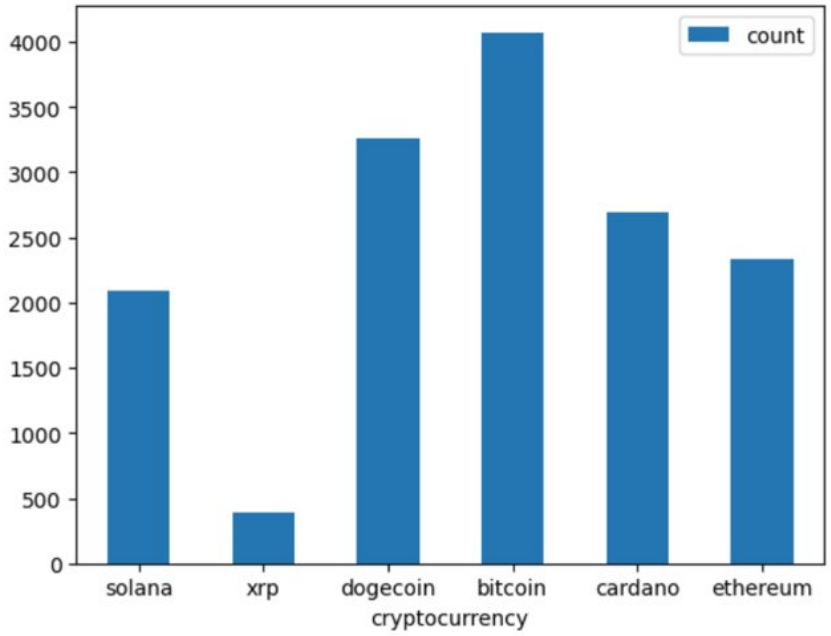

Figure 3.1. Activity of subreddits

There are visible patterns of user activity (understood here as the number of posts submitted). There's a peak of user activity at around 16 o'clock and the low point at 7 o'clock... except the sharp, abrupt peaks around this time (or a little earlier), perhaps related to automated posts being published by bots.

We may also observe a significant decrease of activity between the beginning and the end of the year. In some cases the number of posts fell by two-thirds (the cases of r/cardano and

r/solana subreddits). The cause of that is unknown to us, but may cause the cryptocurrencies to fade into irrelevance because of low popularity.

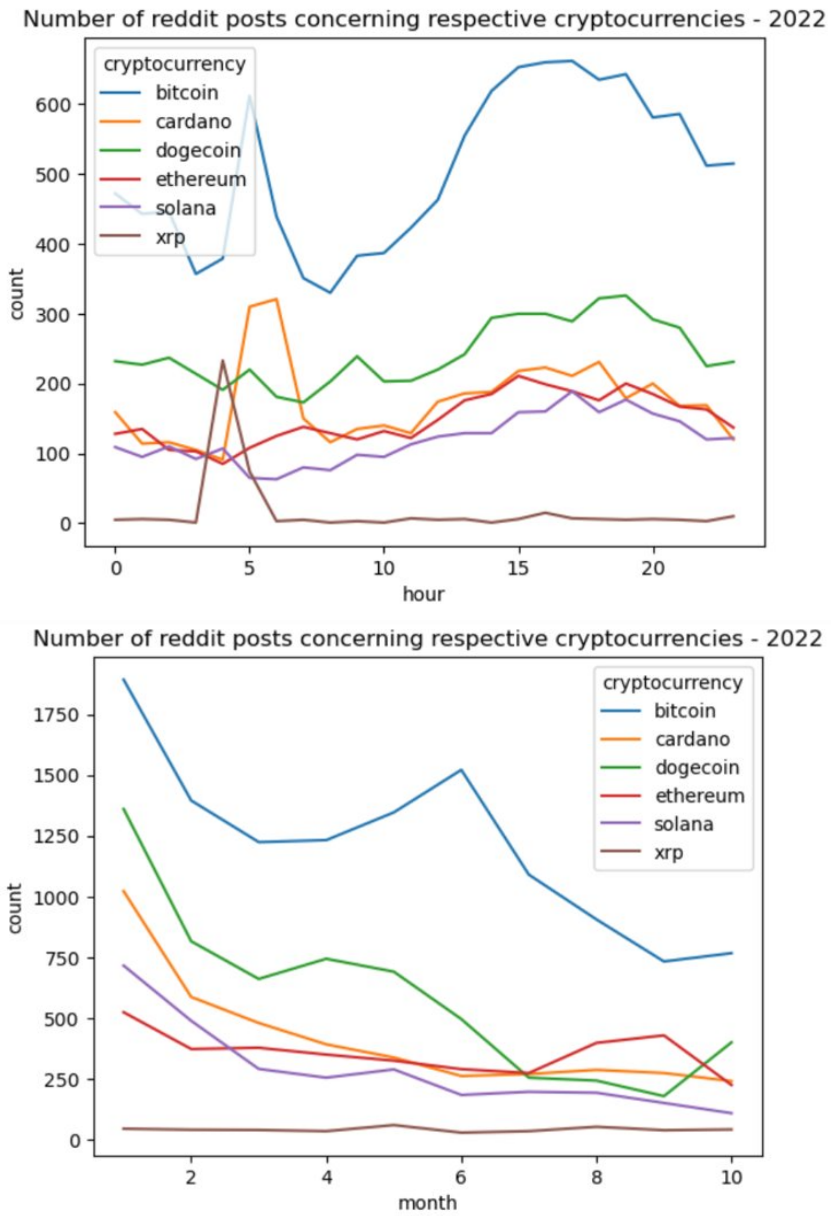

Figure 3.2. Subreddit activity by hour and by month

3.2. Cryptocurrency data

We've first analysed short-term variability of cryptocurrency prices using a randomly selected 24-hour period acquired through streaming Coinbase data. The difference of cryptocurrency values can be expressed in the order of magnitudes, thus the comparison of absolute differences would be meaningless. The exchange rates vary somewhat for all currencies, bitcoin and ethereum

being the most stable (and the most costly). Solana is an outlier here, with the difference between minimum and maximum price being as high as 20% of the minimum value in this period.

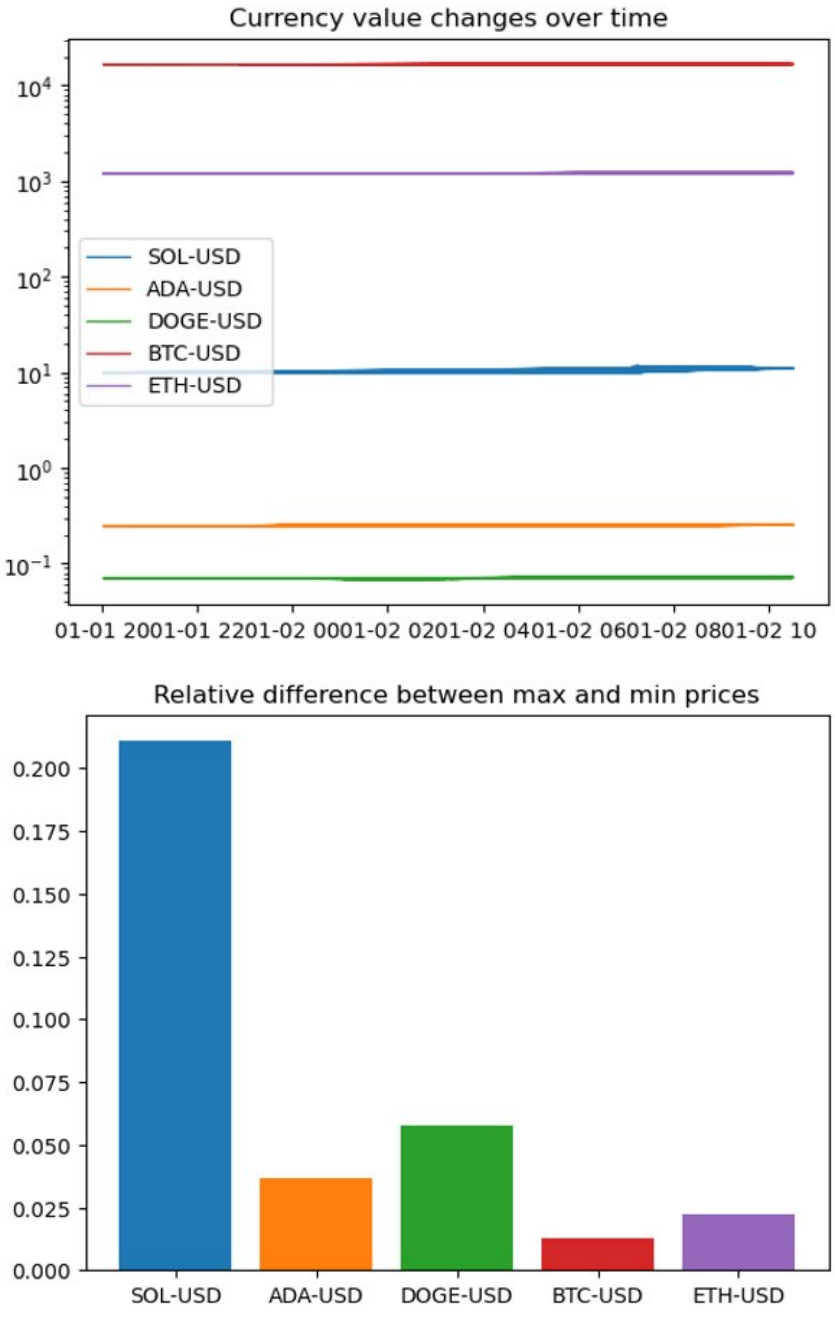

Figure 3.3. Cryptocurrency value analysis over 24 hours

Similar conclusions can be drawn from the three-month analysis on historical crypto data. Solana lost a lot of value in November, with the minimum value being just a quarter of the maximum value. Dogecoin went the other way, gaining value at the beginning of November, almost tripling its exchange rate. It dropped somewhat since then, though.

Moreover, a slight loss of value was present the same day of November for all analysed

currencies. This is probably related to FTX Exchange, one of the largest cryptocurrency markets, filing for bankruptcy protection on 11th of November after a case of embezzlement by its owner.

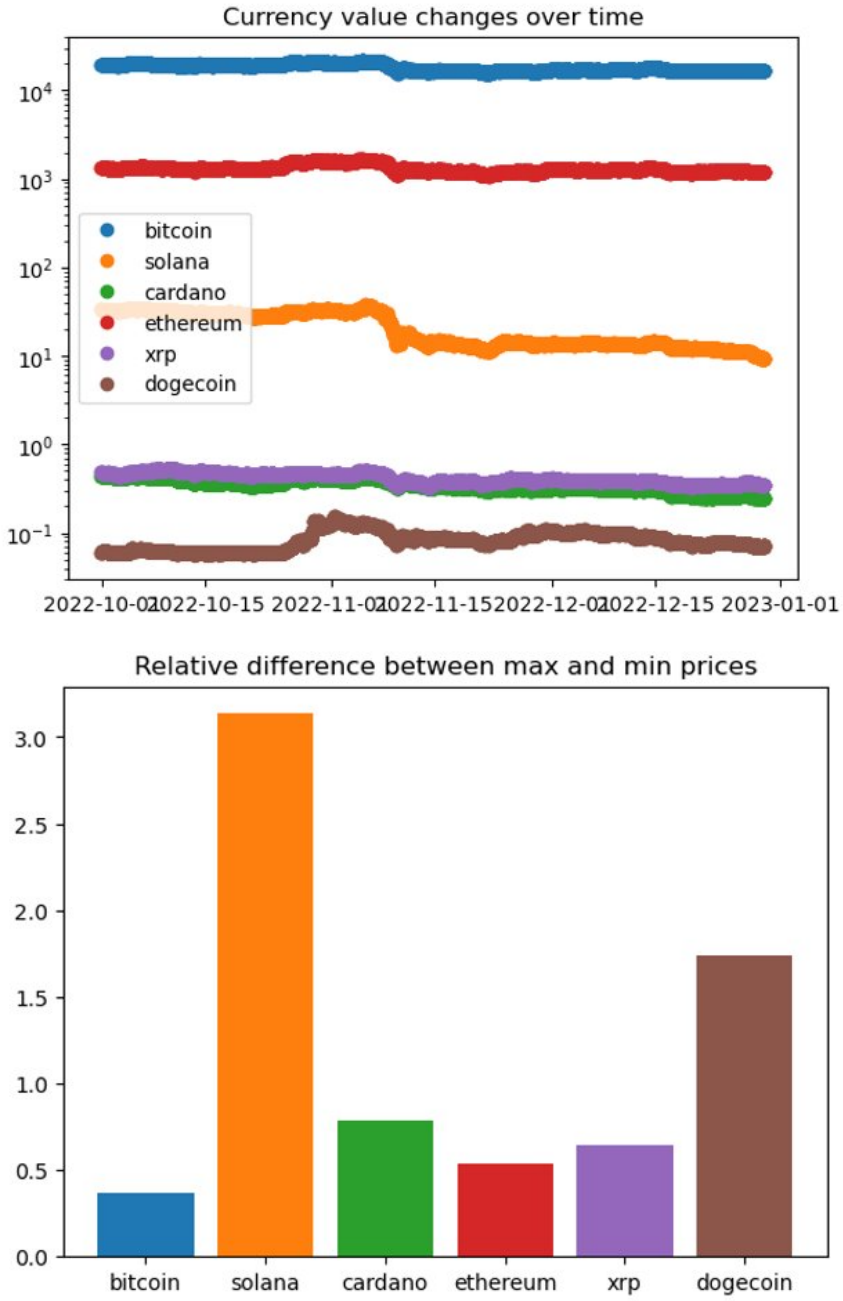

Figure 3.4. Cryptocurrency value analysis over 3 months

3.3. Sentiment data

There's a reasonable balance of negative and positive tweets, although the former is somewhat more numerous. The distribution of sentiment positivity seem not to depend on tweet length, as both distributions look very much alike.

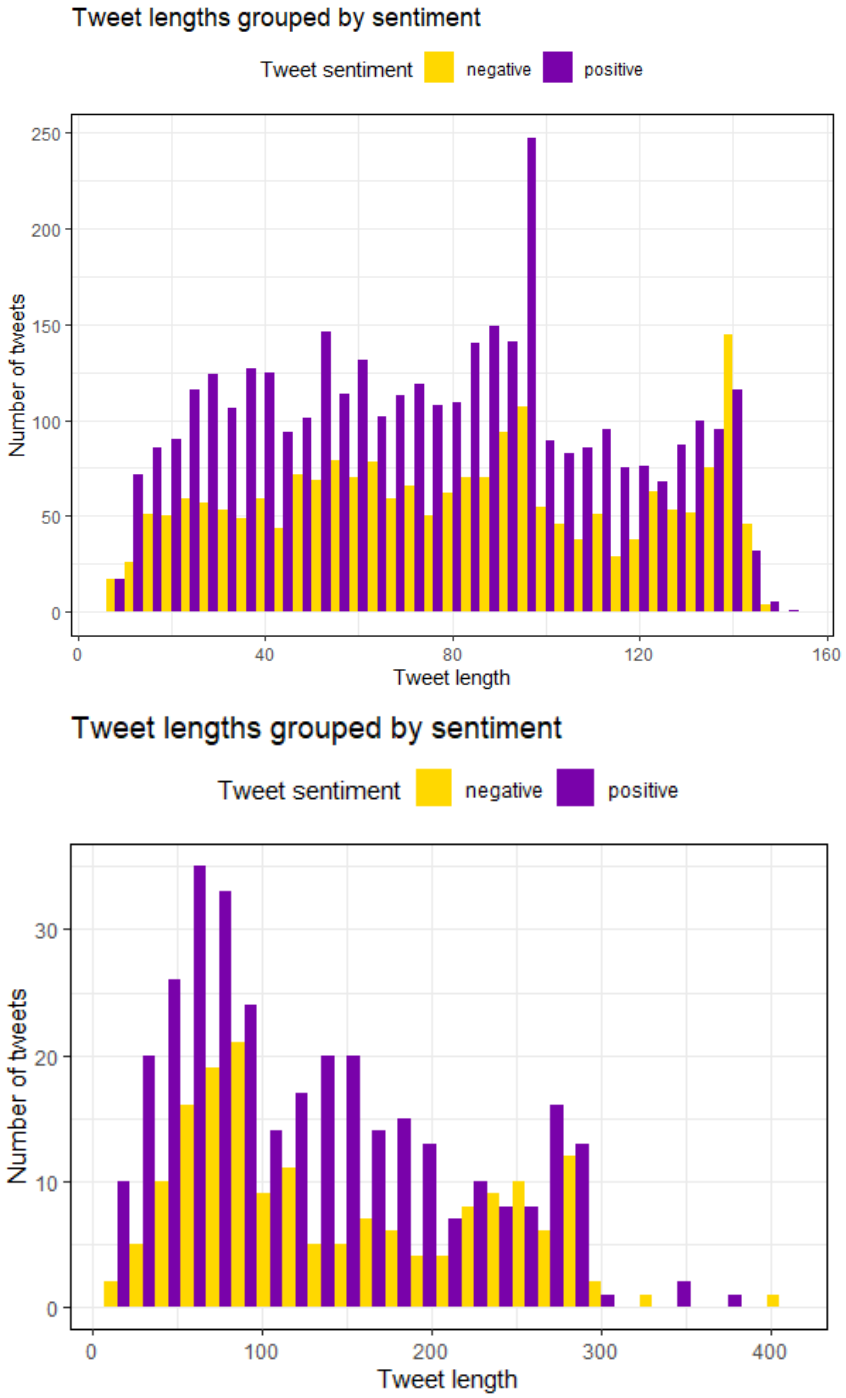

Figure 3.5. Relation of text length and sentiment label

4. Data Preprocessing

4.1. Sentiment Data Transformation and Storage

The sentiment dataset, as a small, not changing dataset, is processed right after being downloaded. The main step of transformations is unifying the target variable (Sentiment) so that it contains the values 1 or -1. Then, using python *nltk* and *re* libraries, the text column (Text) is transformed, leaving out only English words text (that are meaningful to NLP models, but we still do not tokenize the text column). As it is a relatively small and auxiliary dataset, it is stored in hive table *sentiment_table* in CSV format and does not involve bucketing nor partitioning.

4.1.1. Final processed tabular data format

— **Text** – STRING
— **Sentiment** – INT $(-1, 1)$

4.2. Cryptocurrency Data Transformation and Storage

The cryptocurrency dataset is also processed right after being downloaded. The transformation steps include merging the data concerning different cryptocurrencies (creating a new column discerning the cryptocurrency). Another transformation step is to create new columns from the *date* column – year, month, day and hour in order to permit partitioning on year and month. This dataset is stored in hive table *crypto_table* as a serving layer.

4.2.1. Final processed tabular data format

— **date** – STRING
— **price** – DOUBLE
— **cryptocurrency** – STRING
— **year** – INT
— **month** – INT
— **day** – INT
— **hour** – INT

4.3. Reddit Submissions Data Transformation and Storage

The Reddit submissions dataset, as a very huge dataset, cannot be processed right after downloading it – it would constitute a bottleneck in our architecture. Rather, a bash script collects a list of files in the HDFS master data set directory and passes each file, one by one, to a python script, which purpose is to filter out subreddits that are interesting to us (but the business model can change at this moment), to remove submissions having empty post's content (or [deleted], [removed]), and then to use *nltk* and *re* python libraries to leave out only English words in each Reddit post. After these steps, the resulting DataFrame is saved as JSON and stored in other HDFS directory. From these JSON files, a Hive table *reddit_table* is created to serve analytical and modelling purposes.

4.3.1. Final processed tabular data format

— **cryptocurrency** – STRING
— **title** – STRING
— **selftext** – STRING
— **created_utc** – FLOAT
— **year** – INT
— **month** – INT
— **day** – INT
— **hour** – INT

4.4. Cryptocurrency Prices in Speed Layer

In the speed layer, cryptocurrencies are preprocessed to be shown to an end user. Apache NiFi flow both puts the data in HDFS and publishes messages to the corresponding Pub/Sub Lite topic, which are then read by using *pubsublite* format in Apache Spark Structured Streaming Python API.

The messages are decoded and preprocessed using the JSON schema. All variables are cast to proper types, shown for debugging purposes, and then transformed back to the Pub/Sub Lite messages using the format given in the Google Cloud documentation. Finally, the processed messages are published to another Pub/Sub Lite topic and then read using Google Cloud Python API to be appended to the BigTable database.

4.5. Reddit Submissions in Speed Layer

In the case of Reddit data, the flow is similar to the cryptocurrency streaming data. Nonetheless, more transformations are applied to the data – subreddit name has to be matched with

Fields	
key	bytes The key used for routing messages to partitions or for compaction (e.g., keep the last N messages per key). If the key is empty, the message is routed to an arbitrary partition.
data	bytes The payload of the message.
attributes	map<string, AttributeValues> Optional attributes that can be used for message metadata/headers.
event_time	Timestamp An optional, user-specified event time.

Figure 4.1. The fields present in Pub/Sub Lite message. The data is of binary type.

relevant currency (it is enough to make it all lowercase, fortunately), word count is applied on merged title and content data, and, finally, sentiment model prediction is included.

5. System Architecture

5.1. Data Flow Description

The whole data pipeline is served via the Google Cloud Platform. Data ingestion is automated with Apache NiFi on GCP's Compute Engines. NiFi is also used to connect to separate cloud services on Google Cloud. The data is pulled from the mentioned Reddit and Coinbase APIs. Additionally, a messaging system Cloud Pub/Sub is fed with data and used for real-time streaming.

Historical data is directed to HDFS, then it is processed to the serving layer (Hive tables). The real-time data is split into the speed layer (including BigTable) and the batch layer (saved in HDFS). In the case of the batch layer, it is further processed to be stored in the serving layer. There is also error handling in NiFi. The whole flow (especially real-time data submissions) can be investigated in 5.1.

There are two NiFi workflows accessing Coinbase, one for subscribing and digesting websocket data and one for periodic API access. The latter was created due to the unreliability of Coinbase websocket. Error messages received from both pipelines are directed to error handling section, having extracted their relevant message contents. Reddit flow is much simples, as it only requires removing certain characters irrelevant to later analysis, which couldn't be stored in JSON files, and splitting bulk post data. Every piece of data from all these flows is saved to an appropriate directory in HDFS.

5.2. System Architecture - Lambda

The following Lambda architecture elements and their components were used in the project:

1. **Batch layer** (master dataset storage) – Apache HDFS together with Apache Hive for storing the raw data. Also, the processed data was stored in **Avro format** (because updates are made to rows and there are no columns to be added, it's more like transactional data – both Reddit submissions and crypto prices have timestamps).
2. **Serving layer** – Apache Hive as the NoSQL database has been used to store aggregated and preprocessed data. It works on top of HDFS of Hadoop project.
3. **Speed layer** – Google Cloud Pub/Sub for queuing real-time messages and Cloud Dataproc with Apache Spark Structured Streaming for stream processing. For the final delivery, the

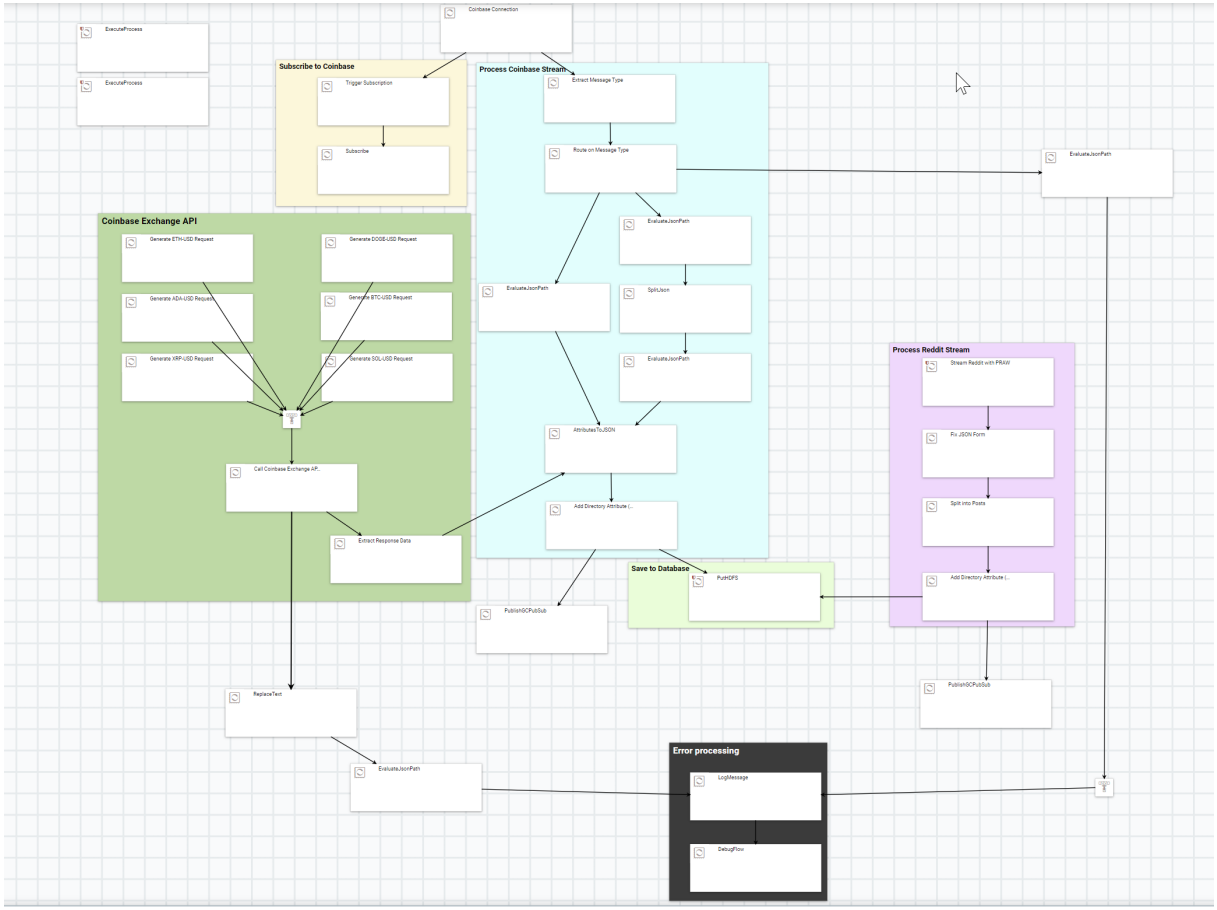

Figure 5.1. The final NiFi data flow

streaming data was stored in Google BigTable in order to allow presenting the most recent cryptocurrency price to the user.

5.3. Main Cluster on Google Dataproc

We have chosen the Google Dataproc service to build the main cluster that stores the master dataset. HDFS is provided by Apache Hadoop, which is the basis of all other functionalities. Google Dataproc had already installations of two NoSQL databases – Apache Hive and HBase. The cluster was used to plan Hadoop, Hive, and PySpark jobs. Thanks to these solutions, We have used the cluster for processing, analytical and machine-learning tasks.

The cluster type is standard, which consists of a master node and N workers (2 in our case). All workers are *n2-highmem-4* VMs with 4 vCPUs and 32 GB of memory. The software and technologies provided by the cluster are as follows:

— Ubuntu 18.04
— Apache Hadoop
— Apache NiFi

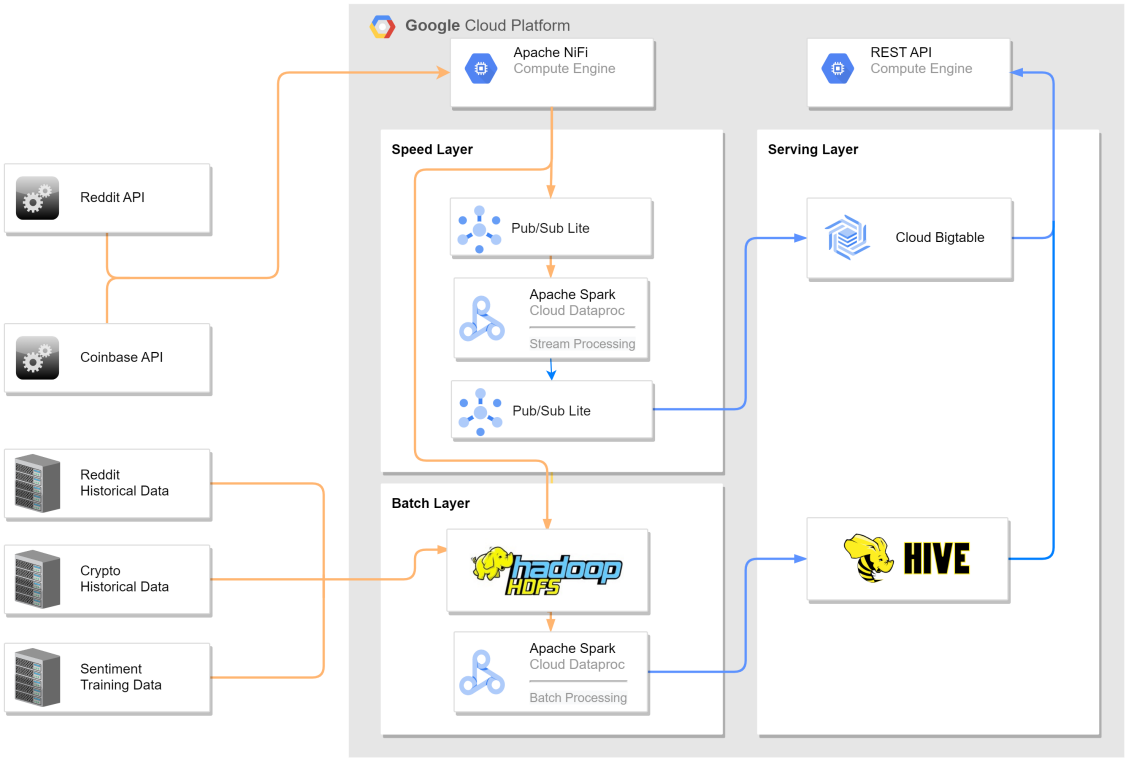

Figure 5.2. The diagram of the Lambda architecture implemented in Google Cloud Platform

— Apache Hive
— Apache Hbase
— Python 3.8 with Jupyter

6. Analytical Module and Machine Learning

For the analytical module, we have created two different machine learning models: one for predicting the sentiment of the Reddit posts, and a second based on past prices of the cryptocurrencies and aggregated sentiment of the Reddit posts from the past few days which will predict future cryptocurrency prices. Additionally, we wanted to create a similar model based solely on past cryptocurrency prices to see if adding sentiment would increase its performance.

For now, we concentrated on the sentiment classification model and model predicting prices solely on past prices, so we could next combine these two and create the final model.

6.1. Sentiment Model

Already preprocessed and normalized text is first tokenized and then vectorized via converting words to hashing codes. Next, we use the Logistic Regression model to predict the sentiment of the text. The model is quite simple but seems suitable for given limited resources on the Virtual Machine.

The model uses a dataset with a total of $6'853$ observations which are all related to either cryptocurrency or stocks and the classes within this dataset are quite well balanced.

To validate the model we split the sentiment dataset into training and testing subsets in proportions 7 to 3 and calculated its accuracy which is equal to $74,06\%$. The final model we used for evaluating Reddit posts was trained on the whole dataset.

Sentiment in time

We predicted sentiment on all posts from Reddit, and according to one of the functional requirements, we created a plot of sentiment in time. The sentiment is aggregated by mean in daily intervals for readability.

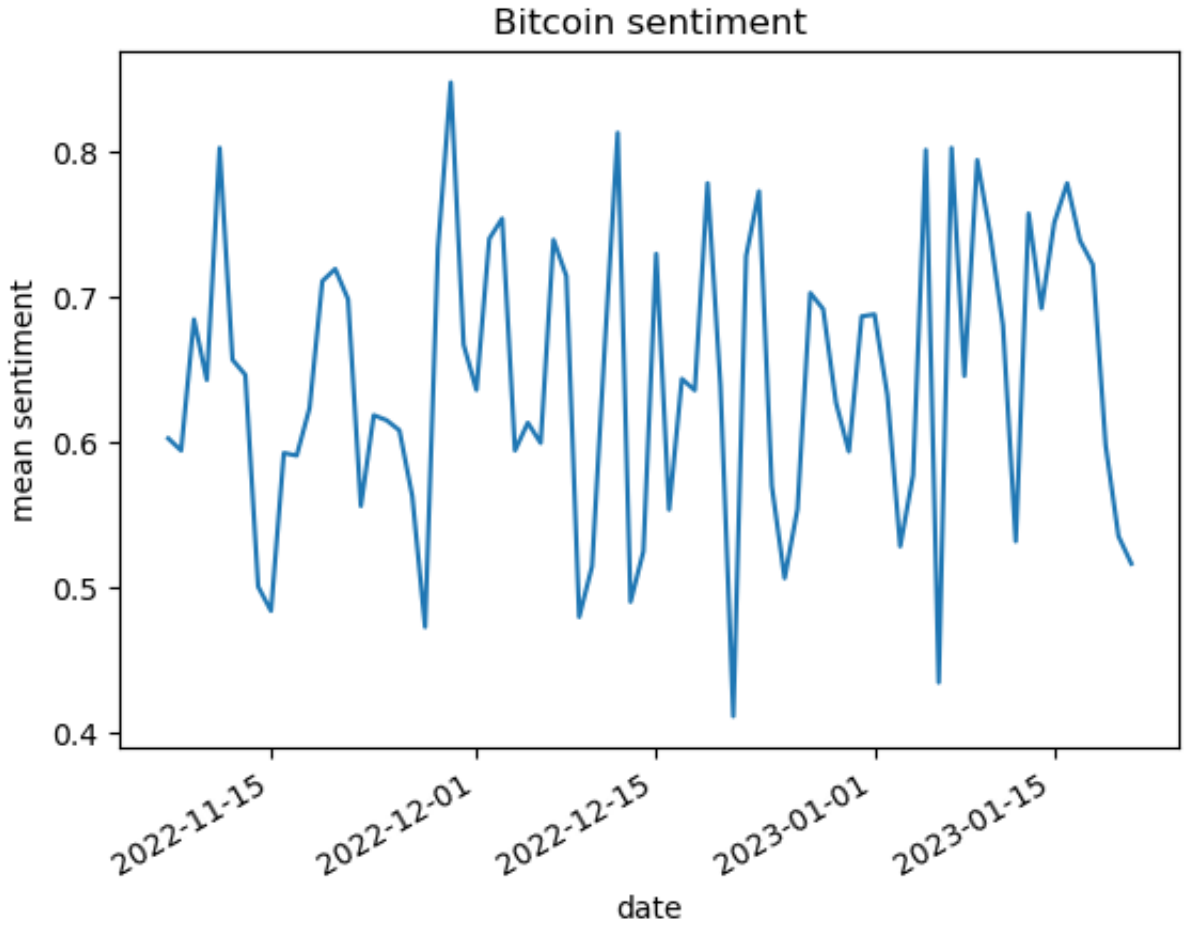

Figure 6.1. Sentiment in time for Bitcoin

6.2. Cryptocurrencies price forecasting based solely on past prices

For predicting cryptocurrency prices we decided to use the Prophet model, which is created for forecasting time series. We decided on this since it is one of the very few forecasting models compatible with PySpark and multivariate (prediction may be based on many variables) at the same time.

We trained the model on both prices from the historical dataset (from 01.10.2022 to 20.01.2023 in the hourly intervals) and prices accumulated via streaming data (which are currently fetched at least once a minute). Since the character of every cryptocurrency is different, we had to create separate models for each of them.

For evaluating the model, we trained it on the period from 01.10.2022 to 30.12.2022 and then tested it on the rest of the days (on January). We calculated three different measures: Mean Squared Error, Mean Absolute Error, and Coefficient of Determination.

We also visualized predictions using below plots. The black dots show training data. The blue line shows estimated prices, and the black line - true prices.

Bitcoin

The bitcoin model has $MSE = 7653884.57$, $MAE = 1881.16$ and $R2 = -0.81$.

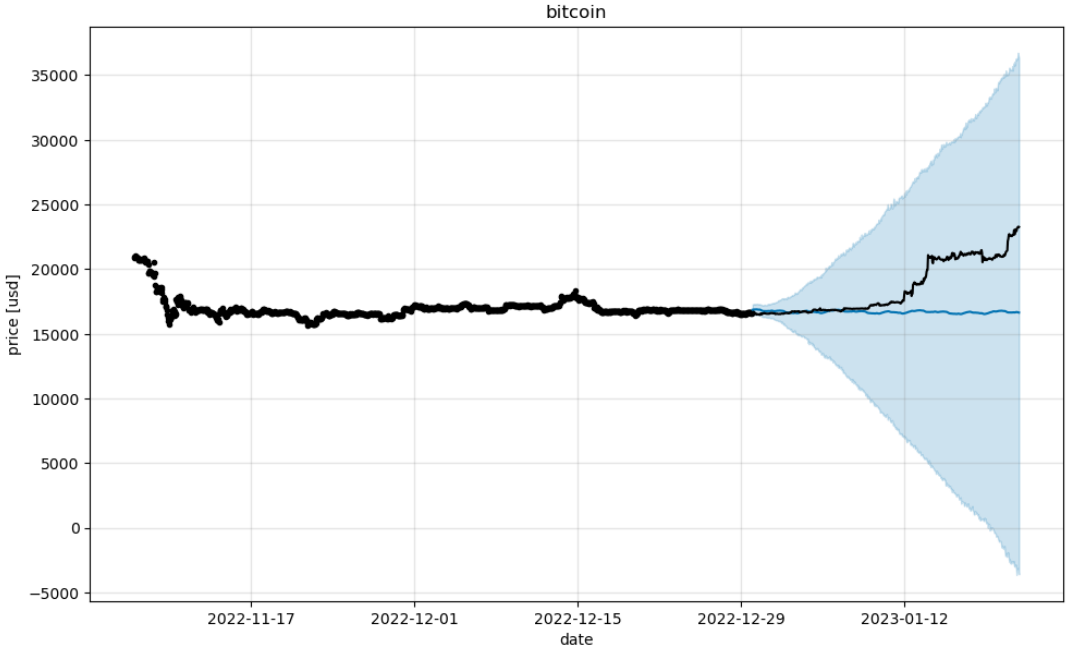

Figure 6.2. Bitcoin price predictions based on price only

Cardano

The Cardano model has $MSE = 0.015$, $MAE = 0.105$ and $R2 = -7.84$.

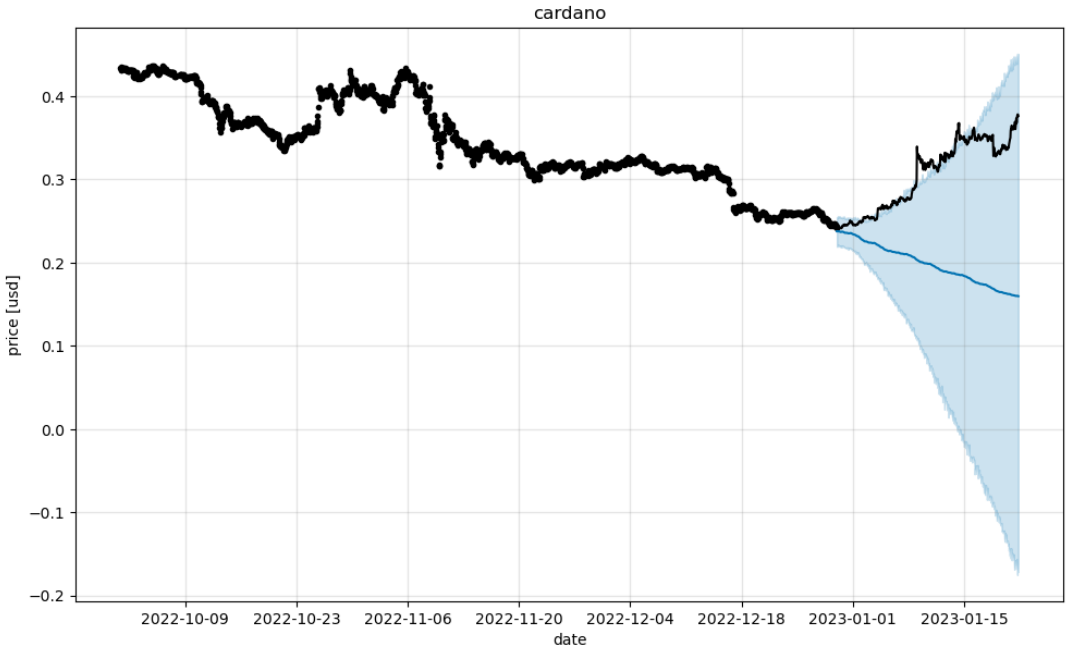

Figure 6.3. Cardano price predictions based on price only

Dogecoin

The Dogecoin model has $MSE = 0.00070$, $MAE = 0.022$ and $R2 = -17.57$.

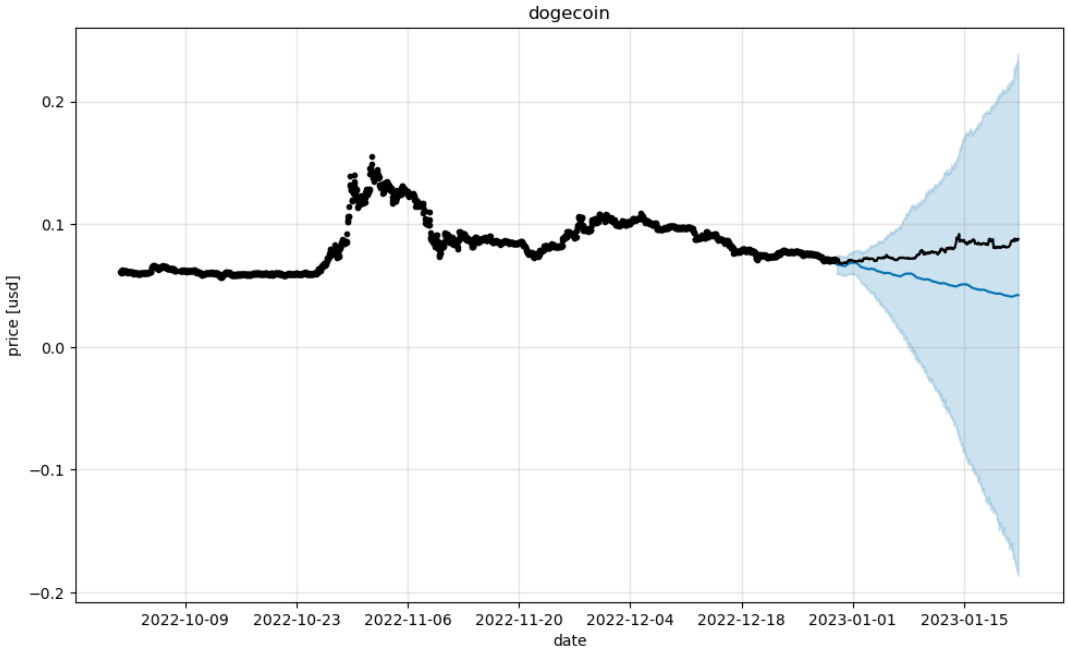

Figure 6.4. Dogecoin price predictions based on price only

Ethereum

The Ethereum model has $MSE = 82776.68$, $MAE = 227.48$ and $R2 = -2.63$.

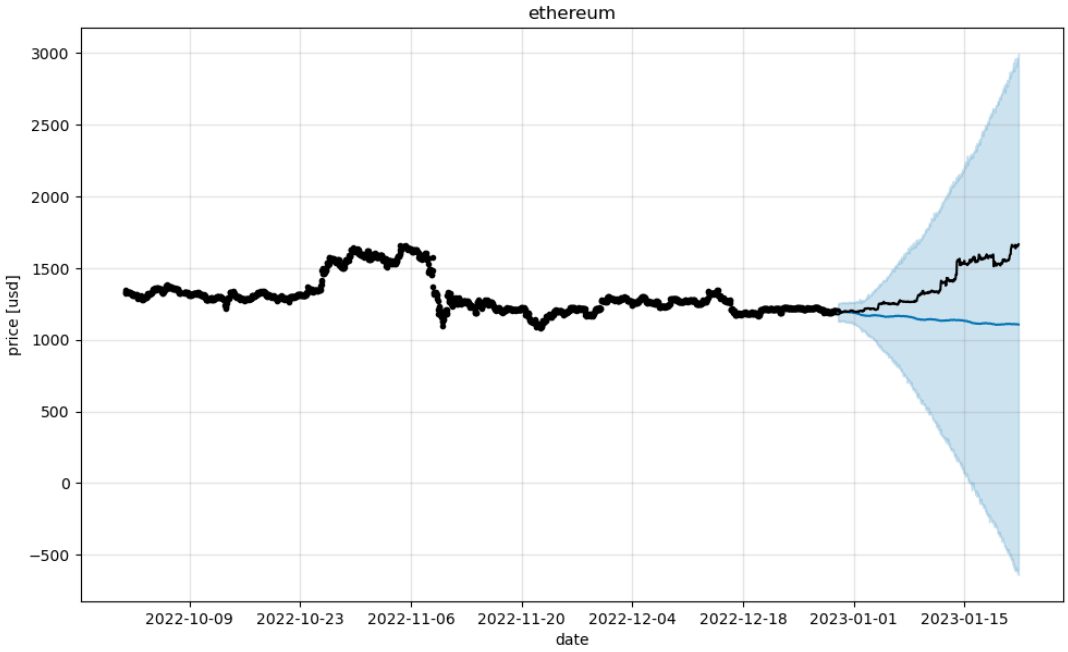

Figure 6.5. Ethereum price predictions based on price only

Solana

The Solana model has $MSE = 110.25$, $MAE = 8.55$ and $R2 = -3.46$.

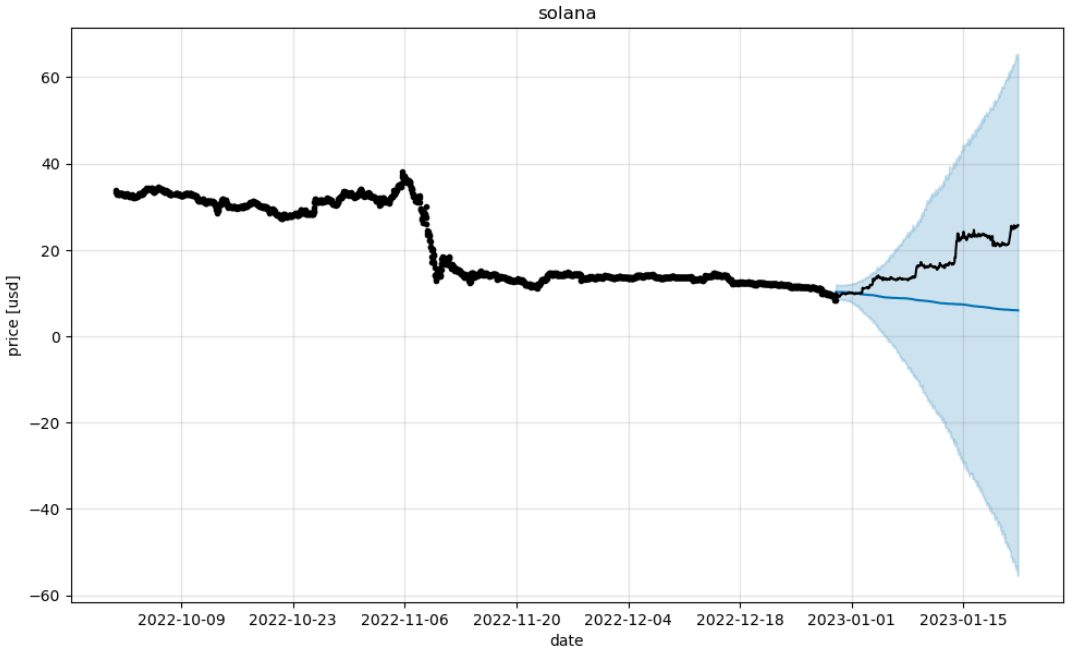

Figure 6.6. Solana price predictions based on price only

XRP

The XRP model has $MSE = 0.0032$, $MAE = 0.045$ and $R2 = -5.00$.

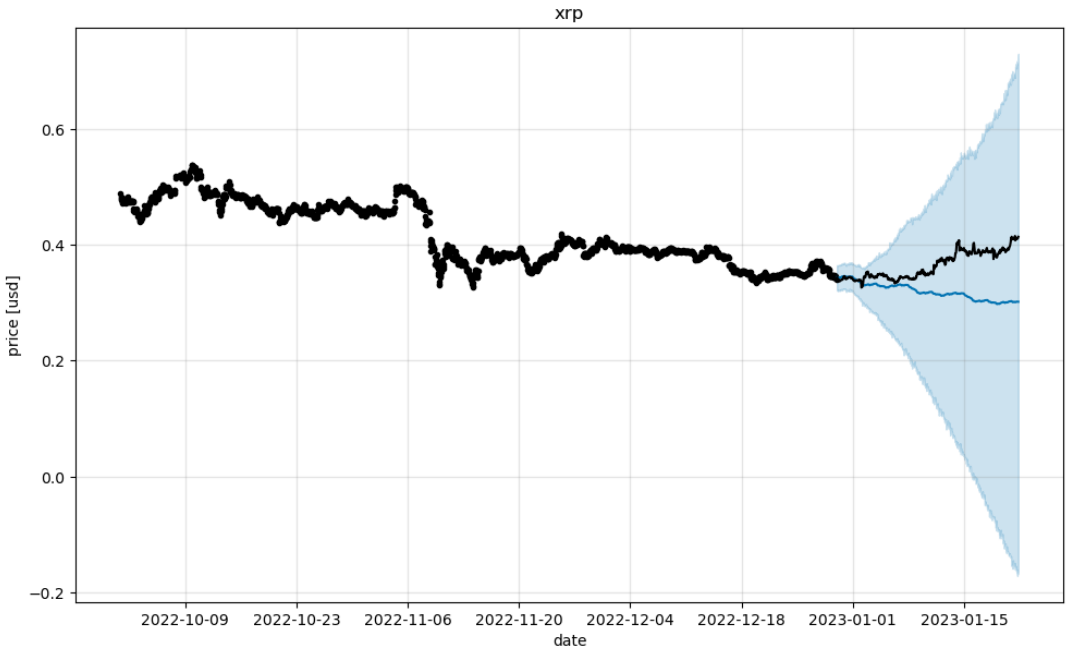

Figure 6.7. XRP price predictions based on price only

6.3. Cryptocurrencies price forecasting based on past prices and sentiment

We used the exact same model as before with the only difference being that this time we added a new regressor to the model (new variable) which is the sentiment and therefore the new model was a multivariate model. Thanks to this, model should use both date and sentiment to determine the price.

For the final model, we prepared the data with both date and sentiment. In order to achieve this, we followed below steps:

1. Predicting sentiment based on trained beforehand model on available Reddit data.
2. Changing all timestamps for cryptocurrency and Reddit data by rounding it to the full hour.
3. Aggregating Reddit sentiment results to one per date by prediction (also tried probability).
4. Shifting dates from Reddit by N days to future.
5. Aggregating columns by date and cryptocurrency (crypto-based).
6. If no price is available for a given hour, it is taken from neighboring rows and repeated
7. It is saved to HDFS as training data for the crypto-price model.
8. Since we shifted the data forward, there will be some "future sentiment", or rather based on the point of view "sentiment from N days ago" - this is needed because both for training and for predicting model needs both date and sentiment. Since it is not possible to have sentiment in the future we do that to get instead a column with past sentiment to make predictions on it.
9. Based on this sentiment and date, we predicted future crypto prices. It is saved separately as "testing" data.

The data for these tests is from the same period as for the price-only model evaluation tests that is trained on period 10.01.2022 – 30.12.2022 and tested on 31.12.2022 – 20.01.2023.

Bitcoin

The bitcoin model has $MSE = 7680873.13$, $MAE = 1886.19$ and $R2 = -0.81$.

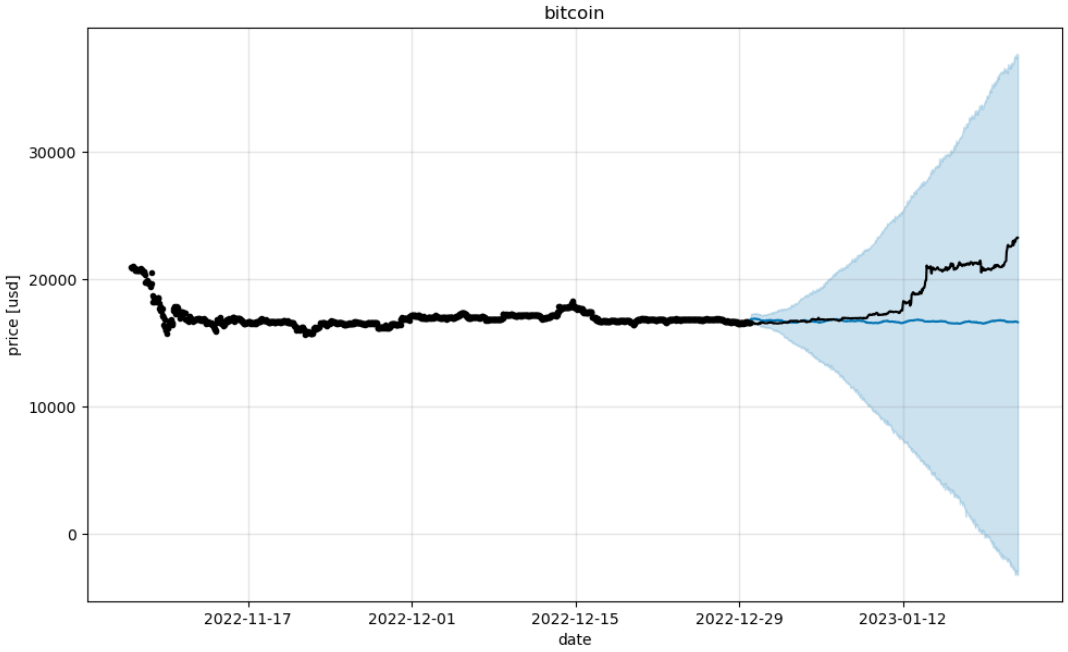

Figure 6.8. Bitcoin price predictions based on price and sentiment

Cardano

The Cardano model has $MSE = 0.015$, $MAE = 0.105$ and $R2 = -7.88$.

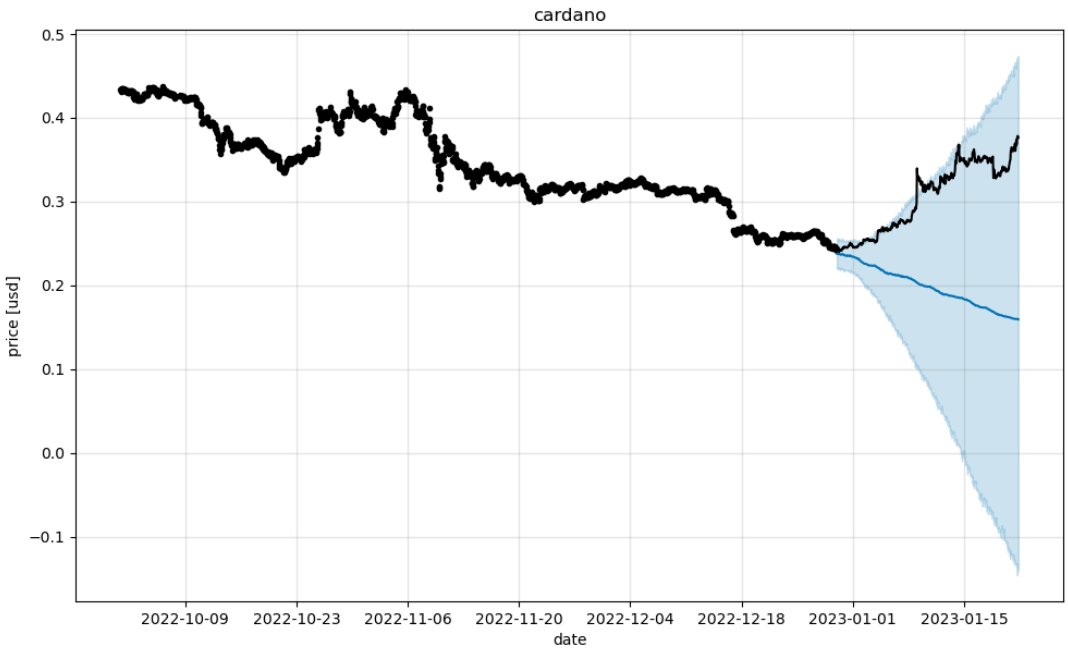

Figure 6.9. Cardano price predictions based on price and sentiment

Dogecoin

The Dogecoin model has $MSE = 0.00070$, $MAE = 0.022$ and $R2 = -17.59$.

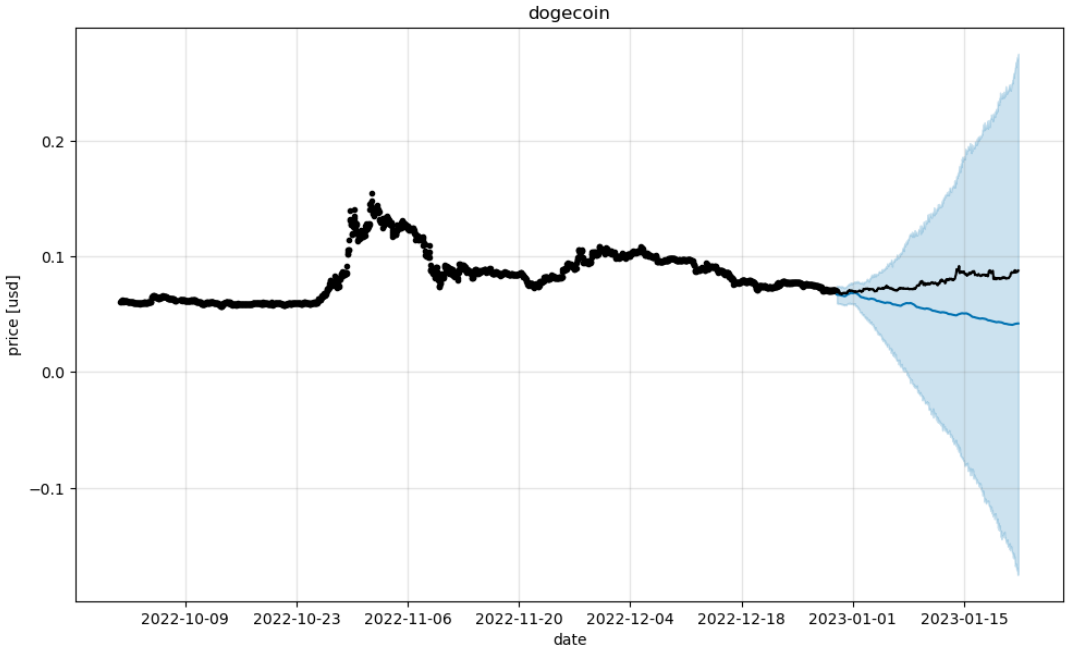

Figure 6.10. Dogecoin price predictions based on price and sentiment

Ethereum

The Ethereum model has $MSE = 83889.26$, $MAE = 229.75$ and $R2 = -2.68$.

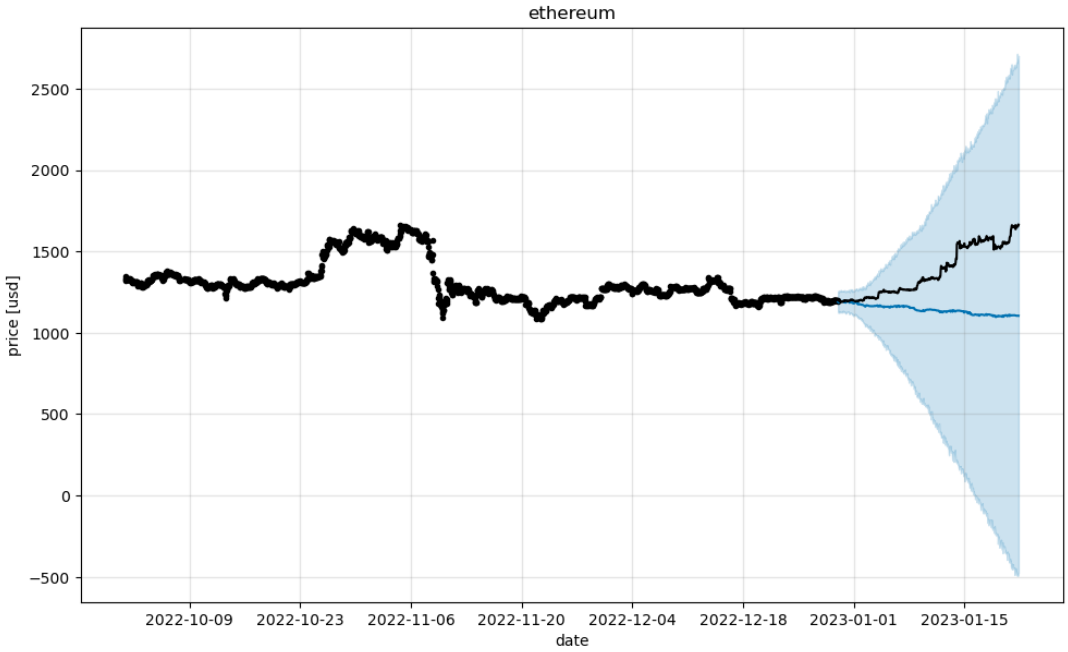

Figure 6.11. Ethereum price predictions based on price and sentiment

Solana

The Solana model has $MSE = 109.10$, $MAE = 8.50$ and $R2 = 3.41$.

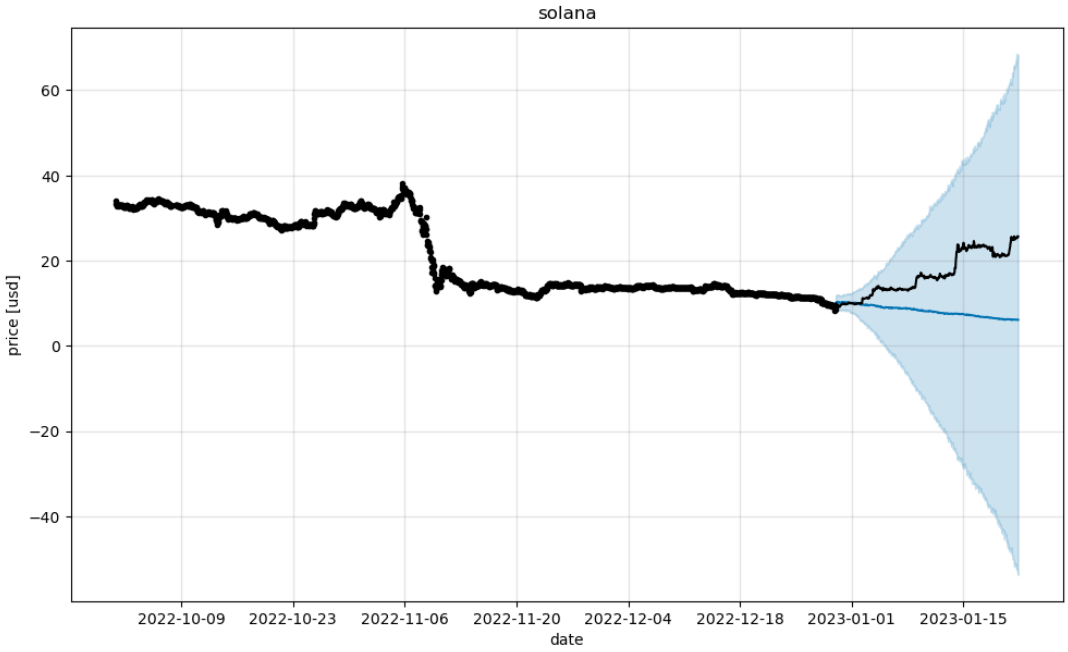

Figure 6.12. Solana price predictions based on price and sentiment

XRP

The XRP model has $MSE = 0.0036$, $MAE = 0.048$ and $R2 = -5.74$.

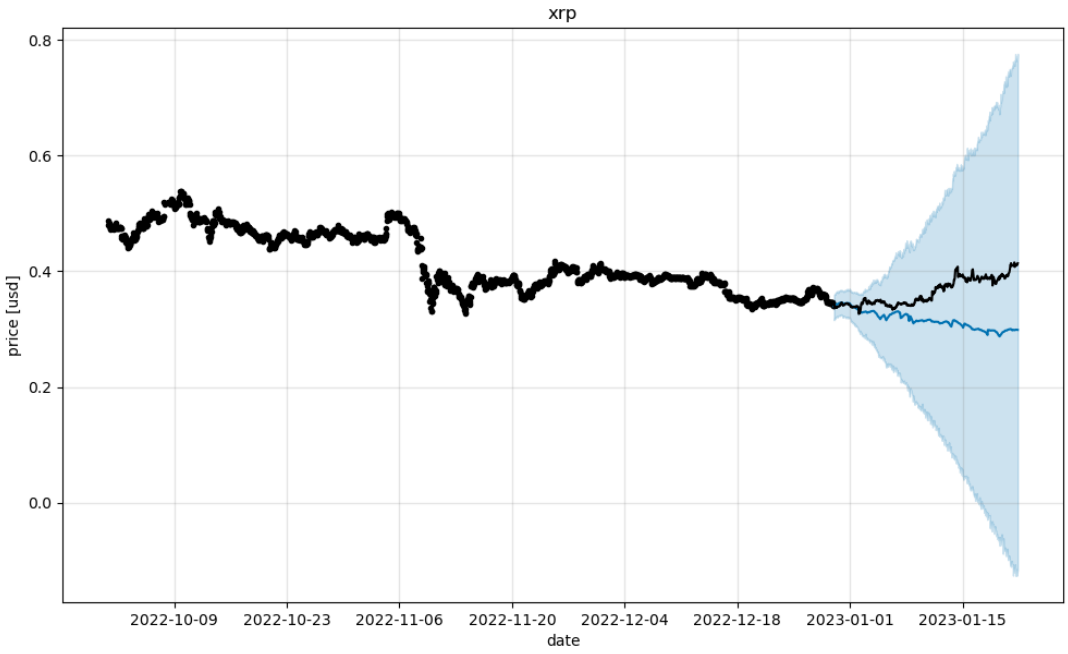

Figure 6.13. XRP price predictions based on price and sentiment

6.4. Predictions with final model

Because we needed more future data (for sentiment) and we didn't need test data for testing purposes, the training period changed to 05.11.2022 – 22.01.2023. The prediction goes 2 weeks in the future which is in the time of the tests: 24.01.2023 – 06.02.2023. The break between training and testing data is the result of the break between creating and training the model (which occurs only once in a while) and testing which goes always from the current moment onward.

Bitcoin

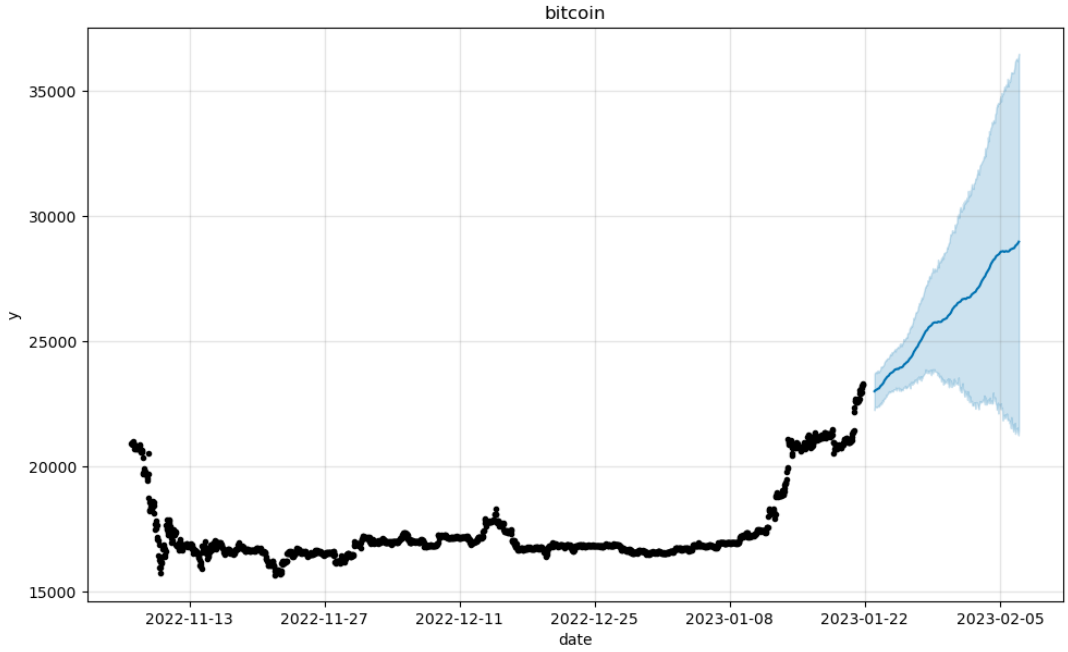

Figure 6.14. Bitcoin price predictions for next two weeks

Cardano

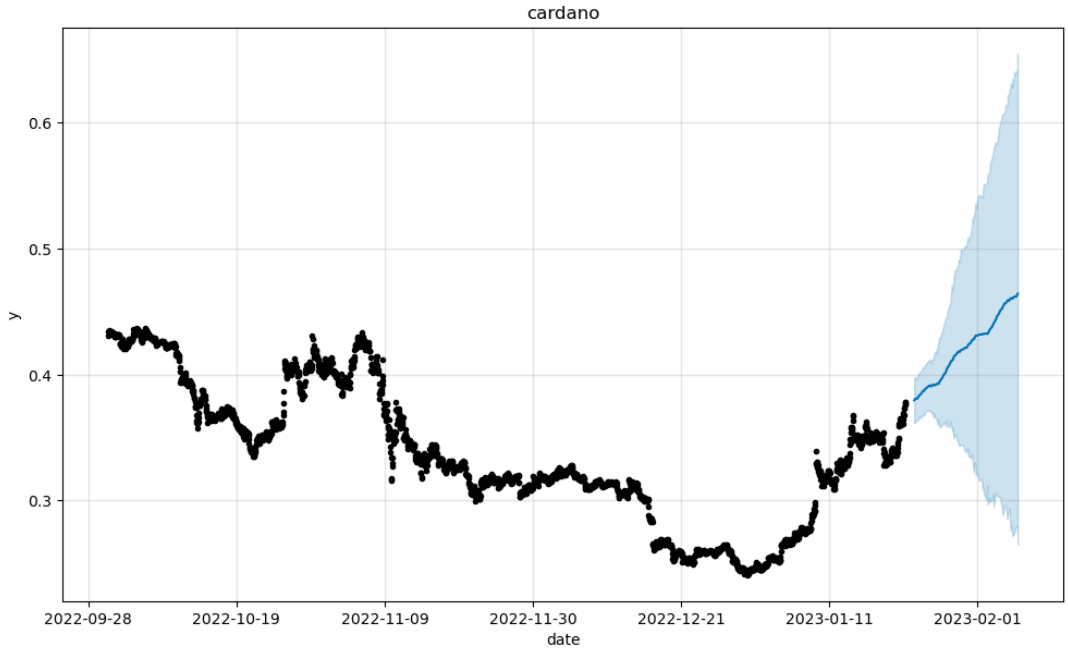

Figure 6.15. Cardano price predictions for next two weeks

Dogecoin

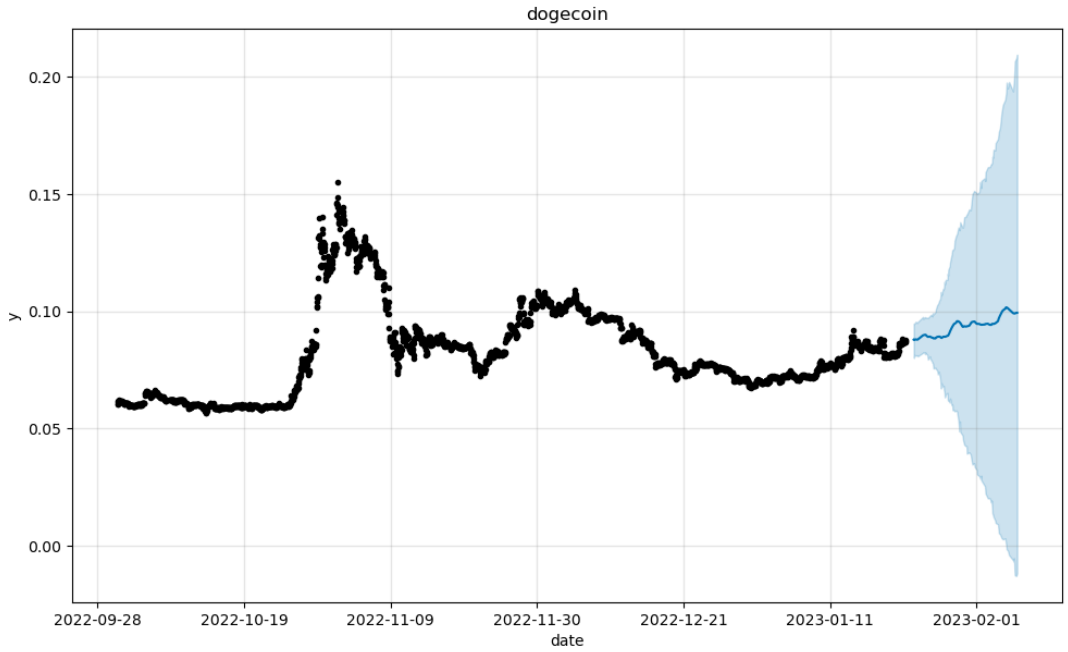

Figure 6.16. Dogecoin price predictions for next two weeks

Ethereum

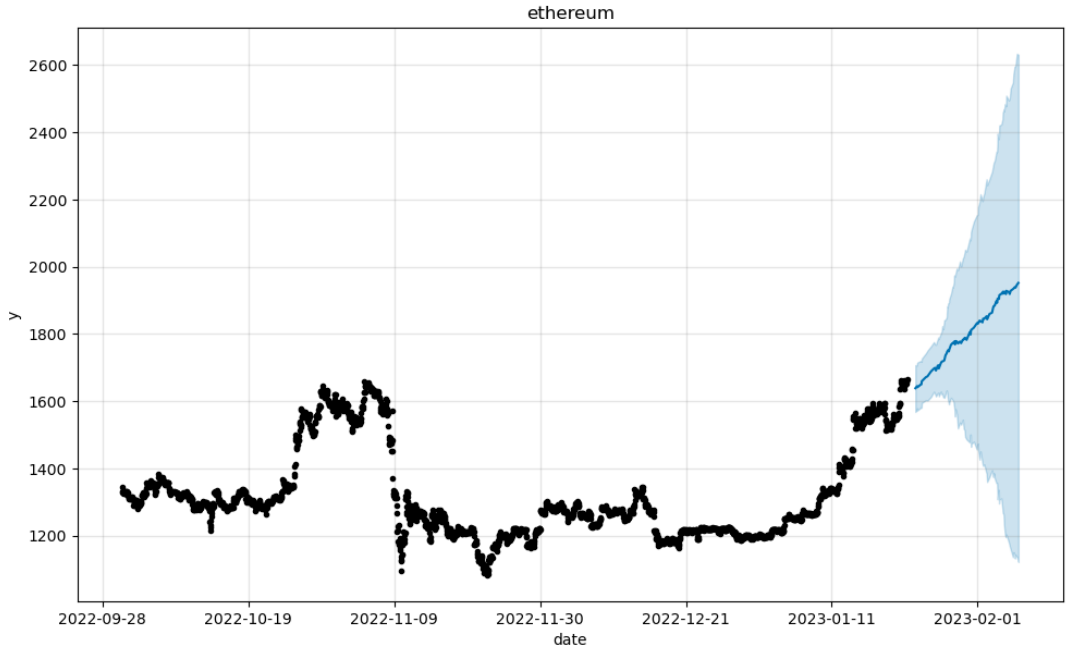

Figure 6.17. Ethereum price predictions for next two weeks

Solana

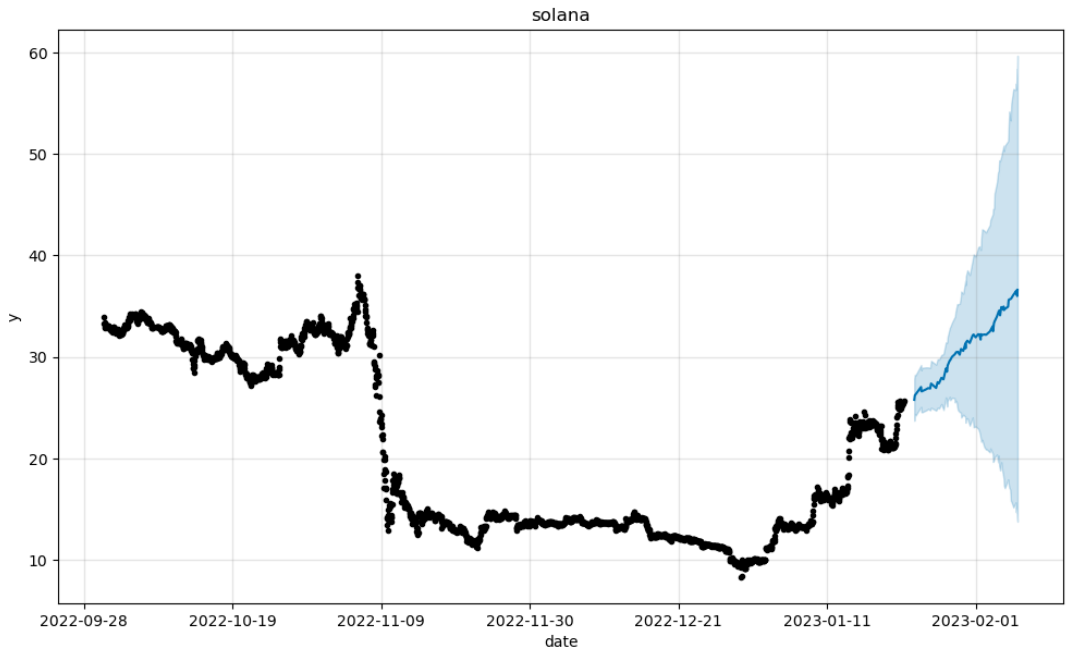

Figure 6.18. Solana price predictions for next two weeks

XRP

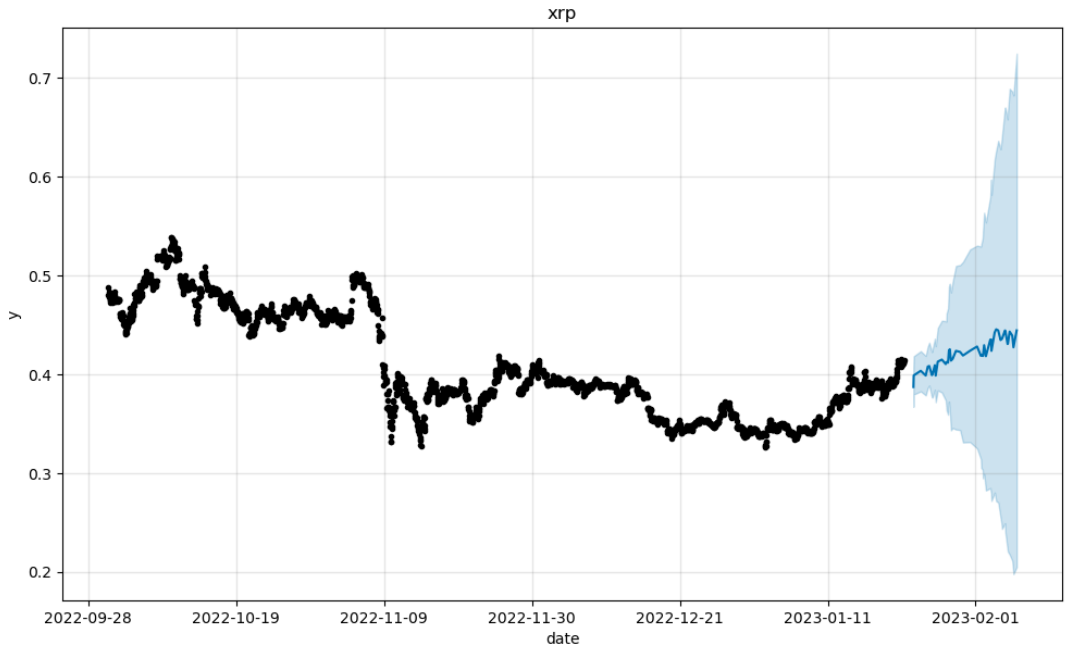

Figure 6.19. XRP price predictions for next two weeks

7. Project results and assessment

The solution uses Apache projects as well as Google resources in order to provide valuable insights into the current cryptocurrency prices and their popularity. The plots showing cryptocurrency trends could be viewed in Jupyter Notebooks accessible via one of Google's web interface.

Reviewing the sentiment and popularity of particular cryptocurrencies can lead to determining whether the public opinion influences the cryptocurrency prices. Thanks to our solution, the model predicting reddit posts' sentiment can show whether or not the chosen cryptocurrency is popular at this moment.

The user can also see the predictions of chosen crypticurrency prices for the near future. Prediction of the cryptocurrency prices is based on a fixed window of time and it takes into account the predicted reddit sentiment within such a window of time.

— possibility of changing the business model thanks to saved raw data in master data set, which increases the **elasticity** of the solution
— thanks to Pub/Sub, combined with Spark Streaming, the solution will adapt to increasing inflow of data streams from streaming data APIs, so in case of peaks, there will be no bottleneck, which will increase the performance of the end solution for the business user
— current cryptocurrency prices can be quickly presented to the end user without a considerable delay – such result is obtained thanks to the speed layer (having components Google Pub/Sub Lite, Spark Streaming, Google Bigtable)
— **scalability non-functional requirement** – the solution could be scaled horizontally, involving more worker nodes, besides, used BigTable NoSQL database assures reliable and consistent batch views storage
— **security non-functional requirement** – thanks to GCP security policies (e.g. IAM, encryption, physical security), the security of user's data is kept
— **fast recovery non-functional requirement** – google permits recreation of thearchitecture from the template, besides, the data is backuped in Google Storage to enable fast recovery scenarios

8. Potential impact of the solution

The project's solution might have impact on various areas of everyone's lives, ranging from economical, ethical and societal aspects:

— Possibility of verifying the influence of public opinion on financial instruments might cause possible cascade of changes (if the opinion is already bad, it will get worse and the inverse could also be possible). Such changes would be rather artificially caused by reviewing the current reddit sentiment with our solution
— Proving the high variance and unpredictability in cryptocurrency prices to describe the risks of cryptocurrency transactions, so that risk-takers could invest in cryptocurrency (Solana and dogecoin revealed to be the most changing cryptocurrencies)
— Combining recent prices with reddit sentiment induces some explainability to how future cryptocurrency prices will change. In case of relying on this model, each investment would impact the real price of the cryptocurrency and its popularity would increase, so the decreasing tendency from 2022 could stop thanks to this solution
— The solution provides an easy way of monitoring the cryptocurrency current prices, and could be extended to investing modules, so the automation of investment could be increased, which would save peoples' time.
— Depending on the accessibility of the solution (restricted usage, premium accounts), the opportunities of investments for common people would increase of decrease, changing the economical gap between the wealthy and the poor.

9. Contribution of individual team members

Each team member contributed to the solutions used in the project according to the following table 9.1.

Table 9.1. List of team members with their completed tasks.

Team member	Contribution
Patryk Wrona	project manager reddit data acquisition and processing Hive, BigTable test documentations
Mateusz Wójcik	data preprocessing, especially NLP Spark Structured Streaming Pub/Sub, Pub/Sub Lite, BigTable GCP environment setup Demonstration video
Dominika Umiastowska	analytical module machine learning models and metrics sentiment and crypto acquisition and processing
Laura Bąkała	exploratory data analysis NiFi flow analytical module

Bibliography

[Cha20] Yash Chaudhary. *Stock-Market Sentiment Dataset*. 2020. DOI: 10.34740/KAGGLE/DSV/1217821. URL: https://www.kaggle.com/dsv/1217821.

[Bus22] Crypto Busters. *Project Repository on Github*. 2022. URL: https://github.com/WronaPatryk/BigDataAnalytics/.

[Sur22a] SurgeAI. *Crypto Sentiment Analysis Dataset*. 2022. URL: https://github.com/surge-ai/crypto-sentiment.

[Sur22b] SurgeAI. *Stock Sentiment Analysis Dataset*. 2022. URL: https://github.com/surge-ai/stock-sentiment.

[Coia] Coinbase. *Coinbase Cryptocurrency real-time API*. URL: https://docs.cloud.coinbase.com/exchange/docs/websocket-overview.

[Coib] Coinbase. *Coinbase Cryptocurrency real-time API available crypto identificator list*. URL: https://api.exchange.coinbase.com/products.

[Coic] CoinGecko. *Cryptocurrency API for real-time and historical data*. URL: https://www.coingecko.com/en/api.

[stu] stuck_in_the_matrix. *Reddit Historical Submissions Archives*. URL: https://files.pushshift.io/reddit/submissions/.