

ТЗ:

Общие термины:

Сайт - одностраничный сайт coingauge.io

Цель сайта - отражение общего "настроения сети" относительно Bitcoin на основе данных Twitter

NS - network sentiment - "настроение сети"

Описание работы:

* Парсер на бэкенде собирает через API данные твиттера, передает их через очереди RabbitMQ на оценку нейросети

* Нейросеть проводит оценку NS для каждого твита и рассчитывает общий NS, затем отдает данные в формате json

* Сайт получает данные через веб-сокеты, обновляется в реальном времени и визуализирует содержимое на фронтенде

Функциональные блоки:

* Лента твиттера с присвоенными каждому твиту значениями NS

- лента визуализируется со скоростью, установленной конфигами (на данный момент 2 твита в секунду), должна отрисовываться по возможности плавно и равномерно

* Графики BTC/USD и NS (реализовано с помощью recharts)

- расположены внизу, и далее называются "нижними"

- отображаются на гибридной шкале, отражающей как цену, так и NS (градуирование с двух сторон шкалы)

* "Тахометр" - стрелка, визуализирующая на шкале от 0 до 1 средний NS, на шкалу нанесены цифровые значения и соответствующие им смайлы, шаг = 0.25 sentiment, располагается в верхней части страницы

* Цена - числовое отображение текущей цены BTC по курсу биржи (+изменения в процентах, рассчитываемые на фронтенде), располагается вблизи тахометра/на нем

Блоки переключателей:

* Выбора интервала для графиков (5 min, 1 h, 24 h, 1 week, 1 month)

- под нижними графиками

* Выбора интервала для тахометра (RT = 10 sec, 30 sec, 1 min)

- под тахометром

Блок social-sharing:

* 3 активные кнопки (telegram, fb, twitter)

- внизу страницы

Архитектура front-end:

* При заходе на сайт клиент коннектится через sock.js к серверу tornado, далее все регулируется коллбэками sock.js

* sock.опорен представляет данные для начальной отрисовки страницы:

Все данные приходят в json файлах одного из 6 типов

- type 1 - история цен BTC/USD для нижнего графика

```
{'type':'price_history', 'period': '1h'(or '3h' or '12h' or '1d' or ...), 'data':[]}
```

- type 2 - история NS для нижнего графика

```
{'type':'twitts_history', 'period': '1h' (or '3h' or '12h' or '1d' or ...), 'data': []}
```

- type 3 - конфиги по временным промежуткам для тахометра и количеством твитов в секунду, выдаваемых в блоке (по факту - переключатели)

```
{'type':'configs', 'data':[{  
'gauge_intervals': [  
{'period': '10s', 'seconds': '10'},  
{'period': '30s', 'seconds': '30'},  
{'period': '1m', 'seconds': '60'},  
],  
'twitts_per_second': 2,  
}]
```

- type 4 - информация о количестве обработанных нейросетью (не фронтендом) твитов (posts analyzed - выводится на страницу) и среднем NS за час и месяц

```
{'type':'overall', 'data':{  
'twitts_counter':1900021,  
'overall_sentiment':[{'1h':0.87, '1m':0.66}]  
}}
```

#overall_sentiment сейчас не показывается, и не слишком нужен

- type 5 - твиты в текстовом виде (json + поле со значением NS)

```
{'type': 'twitts_sentiment', 'sentiment': 0.9 ...}
```

- type 6 - значение цены BTC из API биржи (комплексное, включающее наиболее высокие/низкие за 24 часа, цены спроса/предложения и т.д.)

```
{"queue_type": "tickers", "highest_bid": "10682.00000083", "percent_change": "0.01217547",  
"timestamp": 1519918100, "lowest_ask": "10698.01659358", "is_frozen": "0", "high24hr":  
"10829.99999999", "low24hr": "10201.12439999", "last": "10698.01659373", "quote_volume":  
"2427.33801221", "base_volume": "25578611.42877830"}, "type": "cache_load", "user":  
"9d80d687-7442-40a6-b6e1-f1898c0a5276"}
```

пока используется только для визуализации цены на тахометре, только цена "last"

sock.onmessage предоставляет данные для обновления страницы в реалтайм

- получает json файлы тех же 6 типов, что и sock.onopen

* При получении нового твита (json type 5), твиты складываются в одно и то же место на фронтенде (создаваемый специально массив X, хранящий данные некоторое время, и затем удаляющий старые по принципу first in - first out). При этом, общий счетчик твитов увеличивается на 1 (инкрементально меняет значение posts analyzed, тогда как при получении json type 4 значение posts analyzed перезаписывается)

* Часть твитов из массива X выводится в блоке справа - ленте Твиттера (самые эмоциональные, т.е наиболее далекие от среднего значения 0.5), количество и скорость выводимых твитов регулируется конфигами (json type 3)

* Фронтенд рассчитывает медианное значение NS твитов в массиве X за период взятый из

конфига (json type 3), затем тахометр это значение визуализирует в виде движений стрелки на шкале.

расчет идет именно по общим собранным твитам, а не тем, что попадают в ленту