Prediction of Twitter's impact on Bitcoin's value
Machine Learning for Natural Language Processing 2020

Charles CANEILLES
MS Data Science - ENSAE IP Paris
charles.caneilles@ensae.fr

Abstract

Bitcoin is one of the main phenomena in recent times together with other cryptocurrencies due to the redefinition of the money term and its price fluctuations. Moreover, scientists are increasingly recognizing Twitter's predictive power for a wide range of events, and particularly for financial markets. The goal of this project is to identify a link between the social network Twitter and the price of Bitcoin.

First we will show that there is a correlation between the tweets about Bitcoin and its price.

Then, using Deep Learning algorithms, we will try to predict whether a tweet will have a positive, negative or neutral effect on the price of this crypto-currency.

NB : I'm actually working alone on this project because of an organizational problem with my colleague Skander MARSIT, don't hesitate to contact me if you want more explanations about this.

1 Problem Framing

The first idea of this project was to focus on the classical market (stock market values of companies), the crypto-currency market, and also the NFT market.

This topic being too broad I had to narrow down the scope of the project and thus decided to focus on Bitcoin, the most famous crypto-currency, which will still leave me a considerable amount of data to work with.

The first part of the project is to confirm my intuition that the price of Bitcoin is closely related to everything you read about it on Twitter.

Secondly, it is normal to think that not all tweets affect the price. Indeed, it is even obvious that most tweets have no effect. Then, the ultimate goal of this project is to find out whether a given tweet will ultimately have a positive, negative or neutral impact on the price of Bitcoin.

To address these two issues we will use two datasets from kaggle : Bitcoin related Tweets and Bitcoin price history.

2 Experiments Protocol

2.1 Analysis of our dataset

Before any prediction or analysis work, we had to become familiar with the dataset. Through several visuals we were able to go deeper into the data. These analyses (vocabulary, Zipf's Law, etc.) are presented in part 3.Data visualisation of our tweets of our notebook.

2.2 Highlighting the Twitter-Bitcoin price correlation

Using VADER (Valence Aware Dictionary for Sentiment Reasoning) I'm able to calculate the sentiment "compound" and using this compound I'm giving a score to each tweets.

This score is taking into account :

- The number of followers of the account posting the tweet,

- The number of likes of the tweet,

- The fact if it's a retweet or not.

This score is given by this formula :

$$score = compound \times N_{followers} \frac{N_{likes} + 1}{(N_{followers} + 1)\,(1_{isRT} + 1)}$$

After calculating scores for each tweets we sum hour by hour all the scores obtained for each tweet and we're plotting these scores on the Bitcoin price line.

2.3 Predicting tweet's impact on Bitcoin price

2.3.1 Cleaning the data

First we had to clean our data (tweets), for this we first removed the URLs, hastags and then used the Regex, the TweetTokenizer and WordNetLemmatizer (see cleaning function in the notebook).

2.3.2 Get all of our features

In order to launch a classification algorithm, we first had to complete our dataset. Indeed, thanks to the TextBlob library (a python library that offers a simple API to access its methods and perform basic NLP tasks, such as sentiment analysis), we were able to calculate the subjectivity and the polarity of each tweet.

Then we had to calculate the theoretical sentiment analysis thanks to the Bitcoin price (on a 7 days basis) (see theoricalSentiment and observePeriod functions of the notebook) and then calculate the impact (positive, negative or neutral) of each tweet (getSentiment function of the notebook)

2.3.3 Classification models

As I am more comfortable with the TensorFlow framework, I decided to work with it.

To predict the impact of a tweet on the price of Bitcoin or not I first had to tokenize and do a padding method on our own tweets (tokenize-pad-sequence function in the notebook)

Then I wanted to create two models, a first one quite naive and a second one including a dropout layer and optimized with a stochastic gradient descent. (see appendixes 1 and 2)

3 Results

3.1 First result about our intuition

As expected we have noticed that tweets have an impact on the price of Bitcoin. As seen in the appendices, the time when the score is highest indicates the time when the price of Bitcoin will increase. (see appendixes 3)

Now we will try to find out which tweets will, or not have an impact on the Bitcoin price.

3.2 Classifications results

The results of the two classifications will be shown in appendixes :

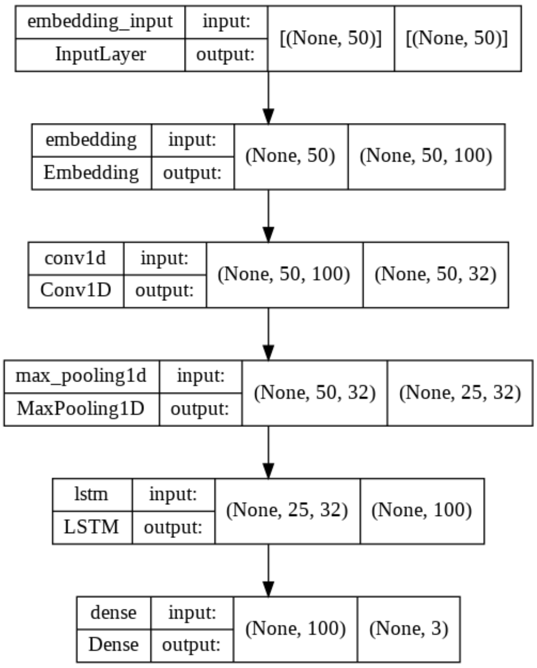
- Appendixes 4 to 6 : First model classification
- Appendixes 7 to 9 : Second model classification

4 Discussion/Conclusion

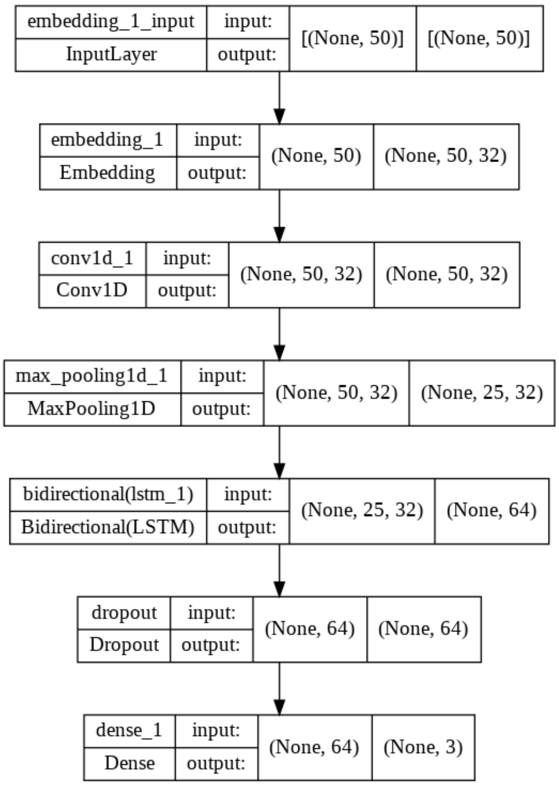
To conclude, we have first seen that tweets about Bitcoin naturally have an effect on its price.

Then the goal of our work was to determine whether a tweet will have an impact (positive or negative) or not (neutral) on the price of Bitcoin. We were therefore able to build two Deep Learning (neural network) models that classify a tweet into one of the following classes: positive, negative, or neutral.

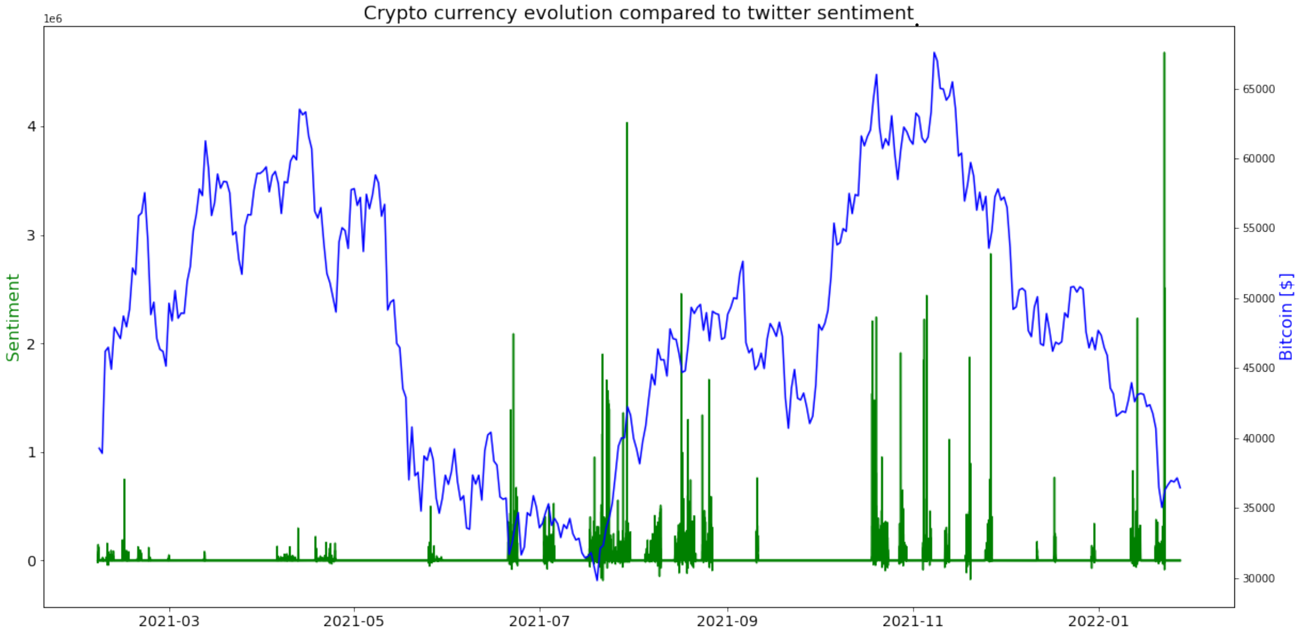
References


embedding_input | InputLayer | input: | output: | [(None, 50)] | [(None, 50)]
embedding | Embedding | input: | output: | (None, 50) | (None, 50, 100)
conv1d | Conv1D | input: | output: | (None, 50, 100) | (None, 50, 32)
max_pooling1d | MaxPooling1D | input: | output: | (None, 50, 32) | (None, 25, 32)
lstm | LSTM | input: | output: | (None, 25, 32) | (None, 100)
dense | Dense | input: | output: | (None, 100) | (None, 3)

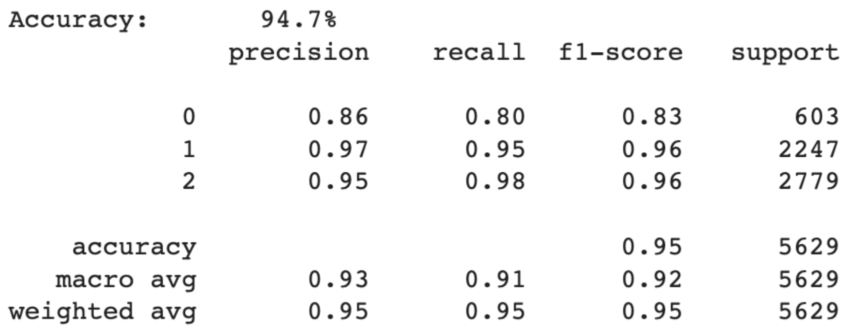
Appendix 1 :Model 1


embedding_1_input | InputLayer | input: | output: | [(None, 50)] | [(None, 50)]
embedding_1 | Embedding | input: | output: | (None, 50) | (None, 50, 32)
conv1d_1 | Conv1D | input: | output: | (None, 50, 32) | (None, 50, 32)
max_pooling1d_1 | MaxPooling1D | input: | output: | (None, 50, 32) | (None, 25, 32)
bidirectional(lstm_1) | Bidirectional(LSTM) | input: | output: | (None, 25, 32) | (None, 64)
dropout | Dropout | input: | output: | (None, 64) | (None, 64)
dense_1 | Dense | input: | output: | (None, 64) | (None, 3)

Appendix 2 : Model 2

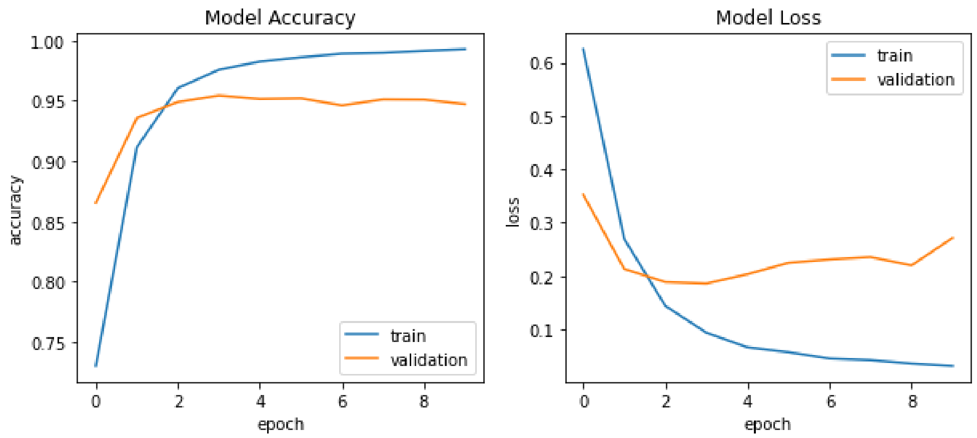


Appendix 3 : Impact of tweets on Bitcoin price

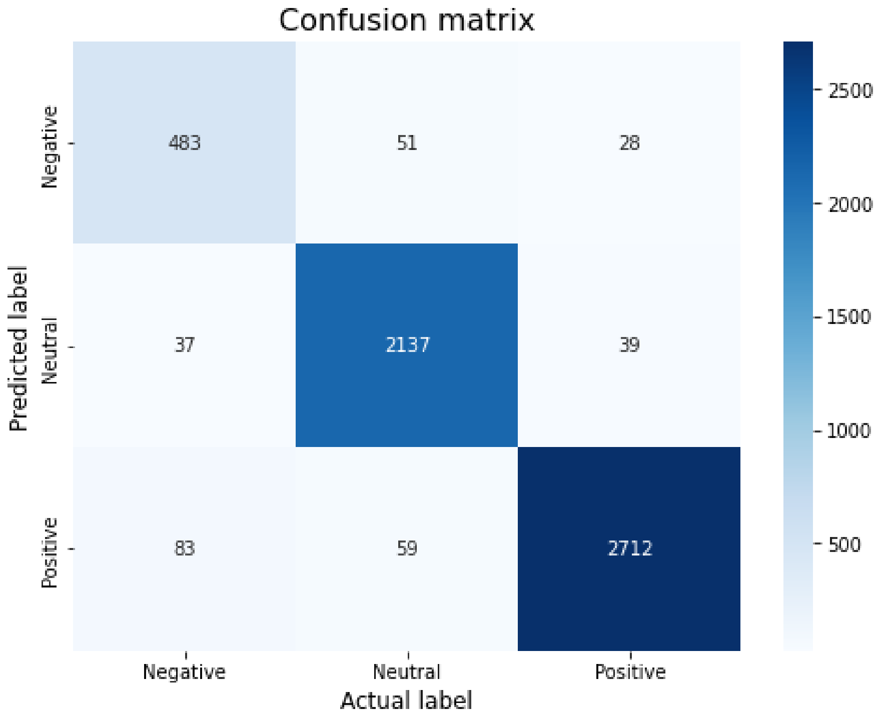


```
Accuracy:       94.7%
               precision     recall  f1-score    support

           0        0.86       0.80      0.83        603
           1        0.97       0.95      0.96       2247
           2        0.95       0.98      0.96       2779

    accuracy                             0.95       5629
   macro avg        0.93       0.91      0.92       5629
weighted avg        0.95       0.95      0.95       5629
```

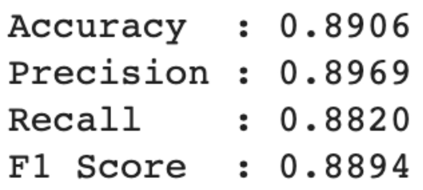
Appendix 4 : Performances of first model



Appendix 5 : First model accuracy and loss evolution through epochs

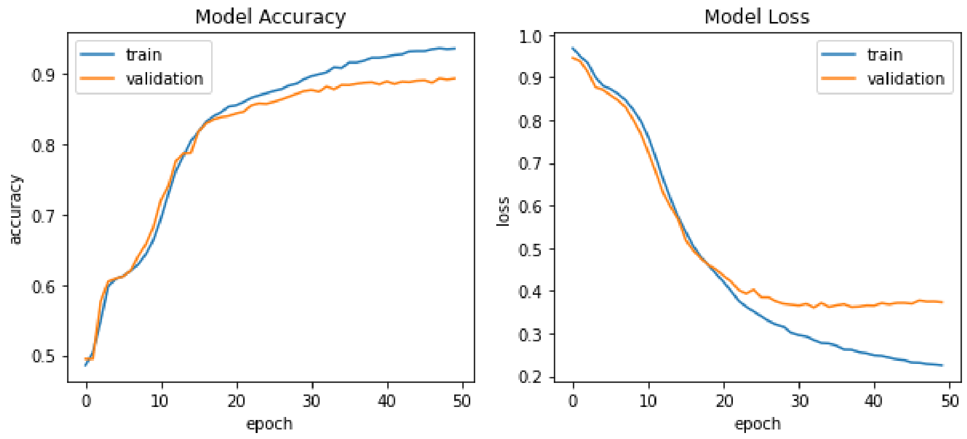


Appendix 6 : First model confusion matrix

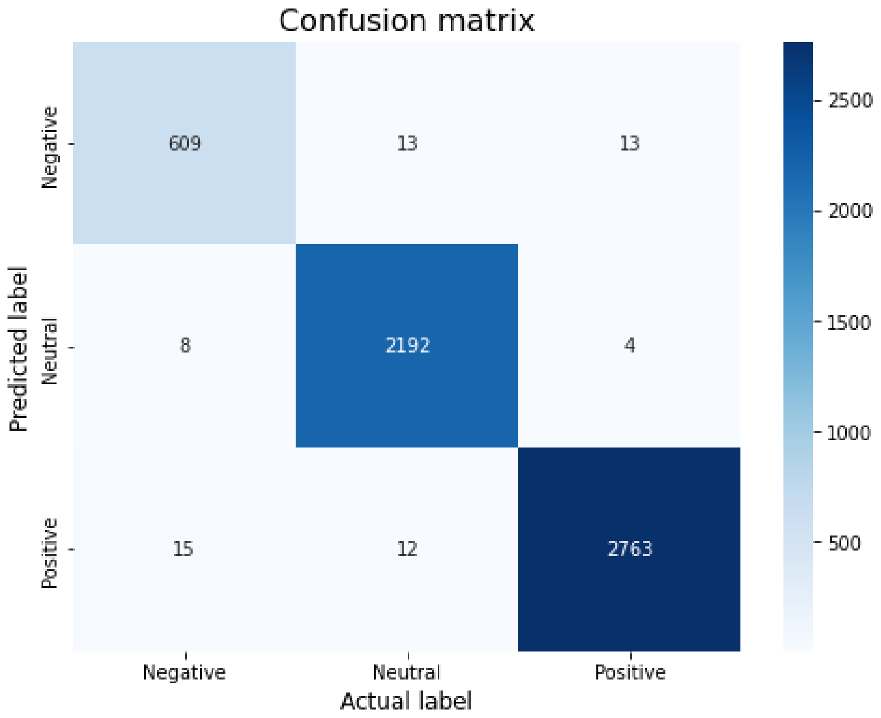


```
Accuracy  : 0.8906
Precision : 0.8969
Recall    : 0.8820
F1 Score  : 0.8894
```

Appendix 7 : Performances of second model



Appendix 8 : Second model accuracy and loss
evolution through epochs



Appendix 9 : Second model confusion matrix