

Trading Strategy Report: Sentiment-Based Approach to Dogecoin Trading

Introduction:

This report summarizes a trading strategy developed to capitalize on Dogecoin price movements by leveraging the sentiment of tweets related to Dogecoin. The strategy is based on the idea that the sentiment of tweets can have a significant impact on the price of Dogecoin. The trading strategy uses a custom-built algorithm to analyze tweet sentiment and generate buy and sell signals accordingly.

Design:

The trading strategy is designed around two primary factors: tweet sentiment and Exponential Moving Average (EMA). Tweet sentiment is determined using the ChatGPT API, which classifies the sentiment of each tweet as either positive or negative. The strategy also employs the concept of sentiment density, represented by the 'sameSentiment' variable, which measures the number of consecutive tweets with the same sentiment within a 24-hour window.

The tweets are sourced from selected crypto influencers who are known for posting content quickly and staying closely connected to market developments. By focusing on these influencers, the strategy aims to capture real-time sentiment changes and make informed trading decisions based on the latest market insights.



The strategy uses four main conditions to generate buy and sell signals:

1. Long Entry:

- No existing long position
- Tweet sentiment is negative
- Closing price is below the EMA

2. Long Exit:

- Existing long position
- Tweet sentiment is positive
- Closing price is above the EMA

- d. Same sentiment occurs consecutively for at least two tweets

3. Short Entry:

- a. No existing short position
- b. Tweet sentiment is positive
- c. Closing price is above the EMA
- d. Same sentiment occurs consecutively for at least two tweets

4. Short Exit:

- a. Existing short position
- b. Tweet sentiment is negative
- c. Closing price is equal to or below the EMA

I have chosen not to set stop-loss (SL) and take-profit (TP) levels for this trading strategy because I analyzed the past year's data and observed that an increase in the density of tweets with the same sentiment often corresponds to relatively high or low points in the market. This observation likely reflects the collective mindset of traders at those moments. By not setting SL and TP levels, the strategy aims to capture the maximum amount of profit possible during these significant market movements.

Implementation:

The strategy is implemented using the Python programming language, with the help of the 'backtesting' library for backtesting purposes. The code defines a custom strategy class, 'MyStrategy', which extends the base 'Strategy' class provided by the 'backtesting' library. The custom strategy class contains two main methods: 'init' and 'next'. The 'init' method initializes the strategy, while the 'next' method handles the logic for generating buy and sell signals based on the conditions mentioned above.

```
import backtesting
from backtesting import Backtest, Strategy
from backtesting.lib import crossover

class MyStrategy(Strategy):

    def init(self):
        super().init()

    def next(self):
        # Check if we should long
        if (self.data.sentiment[-1] == 'negative') and (self.data.Close[-1] <
self.data.EMA10[-1]):
            self.buy()

        # Check if we should exit a long position
        if self.position.is_long and (self.data.sentiment[-1] == 'positive') and
(self.data.EMA10[-1] < self.data.Close[-1]):
            if self.data.sameS_within_1day[-1] >= 2:
                self.position.close()

        # Check if we should short
        if ((self.data.sentiment[-1] == 'positive') and (self.data.Close[-1] >
self.data.EMA10[-1]) and (self.data.sameS_within_1day[-1] >= 2)):
```

```

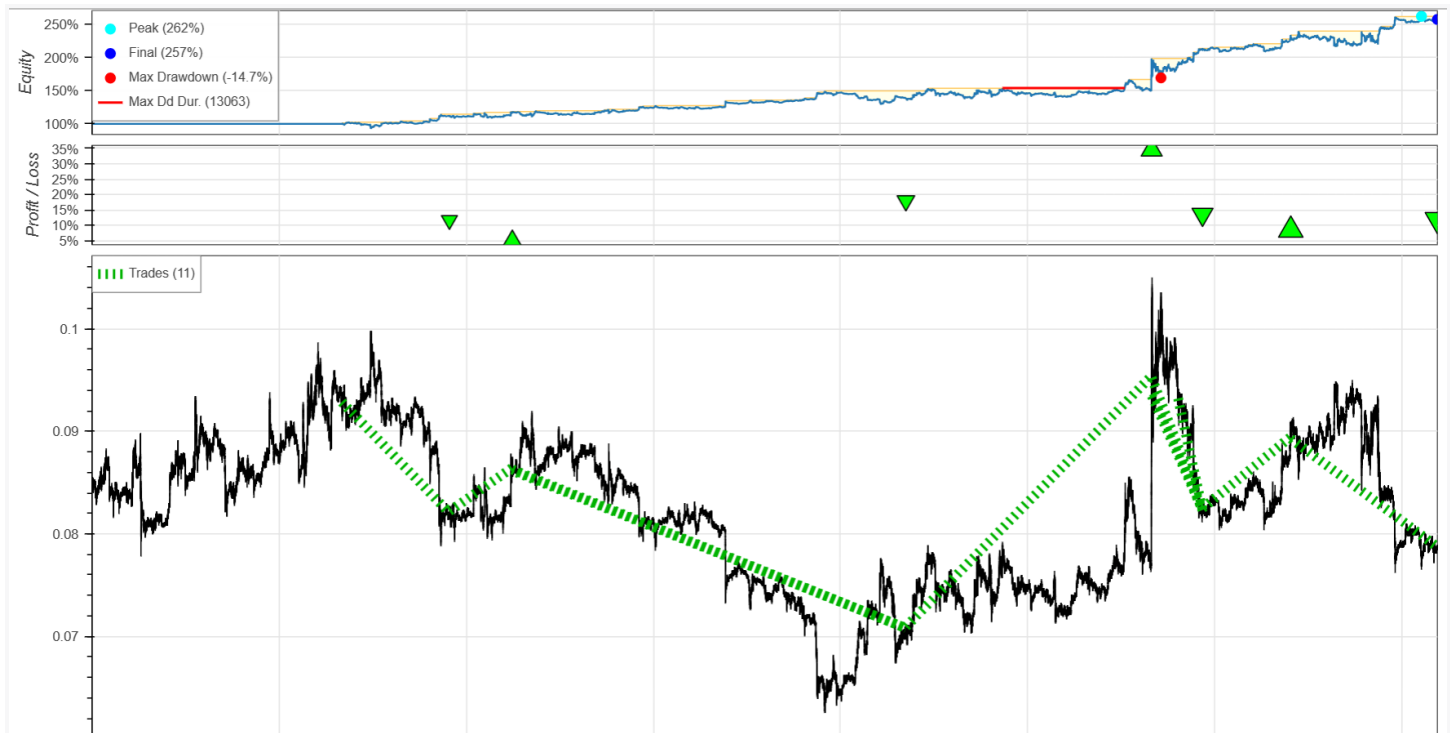
        self.sell()

        # Check if we should exit a short position
        if self.position.is_short and (self.data.sentiment[-1] == 'negative') and
(self.data.Close[-1] <= self.data.EMA10[-1]):
            self.position.close()

# Define the backtest
bt = Backtest(df, MyStrategy, cash=10000, commission=0.002)

```

Performance:



The trading strategy is backtested using historical data for Dogecoin prices from January 1st, 2023, onwards. The initial cash balance for the backtest is set at \$10,000, with a commission of 0.2% applied to each trade.

Can be summarized by the following key metrics:

1. Equity Final: \$25,749.23 - The final value of the equity at the end of the backtesting period.
2. Return: 157.49% - The total percentage return achieved by the trading strategy.
3. Win Rate: 100% - The percentage of winning trades out of the total trades executed.
4. Number of Trades: 11 - The total number of trades executed by the strategy during the backtesting period.
5. Best Trade: 34.49% - The highest percentage gain achieved in a single trade.
6. Average Trade: 13.99% - The average percentage gain per trade.

Challenges and Limitations:

The trading strategy has a few limitations:

1. The strategy heavily relies on the accuracy of the ChatGPT API for sentiment classification which is not stable.
2. The trading strategy might not perform well for short-term trades or during periods of market stability. More news and tweets tend to be posted when the market reaches new highs or lows, and only in such cases does the tweet sentiment provide valuable insights for trading decisions.

3. This strategy depends on the assumption that crypto influencer accounts consistently post the latest information at a rapid pace. If these influencers were to change their editors or become less diligent in providing timely updates, the reliability of their accounts as information sources could be compromised. Therefore, it is crucial to continuously monitor and evaluate the performance of these influencers to ensure that they remain reliable sources of up-to-date information for the strategy.

Areas for Improvement:

The trading strategy can be improved by:

1. Incorporating machine learning and using past data to train a NLP model specifically for crypto related tweets, increase the accuracy of sentiment classification.
2. Adding more technical indicators or market factors to the strategy for better risk management.
3. Optimizing the strategy parameters, such as the EMA period or the sameSentiment threshold, to enhance its performance.