

Desafio técnico - SmarttBot

Artur Gontijo Bessas Fonseca

December 22, 2020

1 Introdução

Neste documento será apresentado o algoritmo desenvolvido para resolver o desafio de implementar alguns indicadores técnicos de mercado utilizando os dados do Bitcoin, para o período informado via linha de comando.

2 Download e Execução

Para ter acesso ao programa, pode-se baixar e descompactar o arquivo .zip enviado por email, ou clonar o repositório que encontra-se em:

<https://github.com/arturbessas/IndicadoresDesafioSB.git>

Para executar o programa, deve-se colar no diretório do projeto o arquivo de input disponível em https://www.kaggle.com/mczielinski/bitcoin-historical-data/data#coinbaseUSD_1-min_data_2014-12-01_to_2019-01-09.csv e renomeá-lo para *input.csv*.

Feito isso, basta acessar o terminal no diretório e executar o seguinte comando:

```
python main.py data_de_inicio data_de_fim
```

Sendo "data_de_inicio" e "data_de_fim" referentes ao período para o qual deseja-se que os indicadores sejam calculados. Ambos devem ser inseridos no formato **dd-mm-aaaa**.

3 Implementação

3.1 Leitura e tratamento dos dados

Na primeira execução do programa, os dados do csv de input serão lidos e armazenados em um *DataFrame*. A seguir, é feito um ajuste dos dados, onde são removidos os primeiros candles que não possuem preço de fechamento. Após o primeiro candle que possui esse dado, todos os próximos candles que não possuem esse valor recebem o valor do último candle que possui um preço de fechamento válido. Feito esse tratamento, os dados ajustados são escritos em um arquivo chamado *adjusted.csv*, que será utilizado como arquivo de input nas próximas execuções do programa, enquanto ele estiver presente no diretório.

3.2 Indicadores

Para armazenar e manipular os dados usados e resultantes do cálculo dos indicadores de maneira mais organizada, a opção foi por utilizar uma classe cujo objeto armazena o DataFrame de input, as datas de início e fim, além do DataFrame contendo os valores calculados para os indicadores.

Para fins de cálculo dos indicadores, foram implementados métodos dentro da classe mencionada, que calculam os valores e os armazena no DataFrame contendo a solução.

3.2.1 Média Móvel Exponencial (MME)

Para calcular os valores de MME, foi implementado um método que recebe como parâmetro o número de períodos da média a ser calculada. Foi escolhida a MME de 20 períodos para resolução do desafio, sem nenhum motivo específico.

O primeiro passo é calcular a Média Móvel Simples de 20 períodos para um período anterior ao intervalo desejado, a fim de alimentar o cálculo da primeira MME. Feito isso, as MMEs seguintes são calculadas seguindo a fórmula:

$$MME_{atual} = (Fechamento_{atual} - MME_{anterior}) \cdot K + MME_{anterior} \quad (1)$$

Onde $K = 2/(1 + N)$ e $N =$ número de períodos.

3.2.2 Índice de Força Relativa (IFR)

Também foi implementado um método que recebe o número de períodos para cálculo do indicador como parâmetro. Foi escolhido calcular o IFR de 14 períodos, usado com maior frequência por investidores.

Começa-se populando dois arrays com dados dos candles: um contendo valores de fechamento de candles de alta, outro de candles de baixa. Eles são populados até que atinjam um número de itens igual ao número de períodos utilizado. A partir daí os arrays são atualizados, seguindo a regra FIFO, removendo-se o dado do candle mais antigo e adicionando-se o candle atual. Quando os arrays atingem um número de itens igual ao número de períodos utilizados, o IFR referente ao candle atual pode ser calculado da seguinte maneira:

$$IFR = (100 - (\frac{100}{1 + \frac{U}{D}})) \quad (2)$$

Sendo U a média dos valores do array de alta, e D a média dos valores no array de baixa.

3.2.3 Bandas de Bollinger (BBs)

Assim como os outros indicadores implementados, as BBs recebem como parâmetro um número de períodos, usado para calcular a média móvel e o desvio padrão.

Começa-se populando um array com dados de fechamento dos candles. Quando o array atinge um tamanho igual ao número de períodos utilizados, passa-se a atualizar o array

removendo o dado mais antigo e inserindo o dado do candle atual.

Então para cada candle após o array atingir esse tamanho, são calculadas as bandas superior e inferior, definidas por:

$$Banda_{sup} = MMS + 2 \cdot dp \quad (3)$$

$$Banda_{inf} = MMS - 2 \cdot dp \quad (4)$$

Sendo MMS a média móvel simples referente ao período trabalhado, e dp = desvio padrão do array acima mencionado.

3.3 Plotagem

Para plotar os dados, foi definida uma classe responsável por configurar e executar os plots. Foi definido um método separado para plotar cada indicador, e os gráficos gerados são salvos como imagens no diretório do projeto.