

Nanodegree Engenheiro de Machine Learning

Proposta de projeto final

Marcelo Luiz de Amorim Cabral

06 de Maio de 2018

Proposta

A proposta de projeto final que apresento a seguir seguirá um dos modelos sugeridos no item 02. *Descrição - Proposta de projeto de conclusão*, e dos quatro modelos, a minha opção foi pelo proposto no hiperlink *Investimentos e trading* [1].

A minha escolha se deve ao fato de que eu opero na bolsa de valores e os modelos de predição existentes carecem de maior eficiência ou de um treinamento mais apurado das técnicas de análise empregadas, tais como, curvas de bollinger ou a média móvel de η dias (MACD) e o critério de escolha da técnica de análise costuma ser a preferência do investidor quanto ao modelo ao qual se habituou ou quanto ao que eventualmente o trouxe mais sorte.

Este projeto visa basicamente testar um modelo de inferência baseado em regressão linear, pois esta inferência já é utilizada para análise técnica de ações e a regressão KNN que aparentemente se ajusta aos requisitos de análise técnica. Em ambos os casos, os modelos serão comparados às análises técnicas MACD e Bollinger, visto que ambas possuem grande aceitação e a MACD, em particular, tem forte similaridade com a regressão linear.

Vale destacar que a MACD é amplamente utilizada por sua simplicidade e um sistema de aprendizagem de máquina que se aproxime dela é algo que permite avaliar o resultado esperado.

As técnicas de análise para a predição de ações tem seus graus de eficiência e estão sujeitas a erros mediante situações adversas, que nos casos mais graves são conhecidos entre os economistas como “Eventos de Deus” tais como o Joesley day na B3 Ibovespa, a quebra do banco Lehman Brothers e o consequente estouro da bolha imobiliária americana na economia mundial.



Figura 01 – MACD [2]



Figura 02 – Bandas de Bollinger [3]

A aprendizagem de maquina será testada por cenários controlados, uma vez que o conjunto de ações escolhido se refere a um único segmento da economia e fatores externos à análise técnica tendem a impactar no desempenho do segmento e a recuperação dos valores destas ações tendem a seguir um padrão conjunto.

Histórico do assunto

A análise técnica de ações não é algo novo. As diversas técnicas empregadas seguem diferentes padrões não relacionados, por vezes meramente intuitivo, baseado em experimentação pessoal e cenários finitos de épocas distintas em bolsas distintas. Inicialmente, qualquer trader é apresentado a maioria das técnicas e tende a seguir por aquela que ele mais se habitua.

Atualmente, temos visto um gradativo aumento de robôs trader, que tendem a empregar as técnicas que seus autores estão habituados a utilizar em suas análises manuais. Mas quase ninguém é capaz de afirmar quão mais eficiente é sua técnica de análise é em relação as demais disponíveis.

Descrição do problema

Após uma análise do cenário proposto em *MLND Capstone Project Description - Investment and Trading* alguns pontos relevantes merecem destaque, sendo eles:

01 – As API propostas na página não externam ações listadas na B3 IBOVESPA, mas sim as listadas nas bolsas Dow Jones e NASDAQ, ambas americanas, tais como:

```
In [3]: import pandas_datareader.data as web
import datetime as dt
start = dt.datetime(2018, 4, 20)
end = dt.datetime(2018, 4, 27)
f = web.DataReader('GGB', 'morningstar', start, end)
f.head()
```

Out[3]:

		Close	High	Low	Open	Volume
Symbol	Date					
GGB	2018-04-20	4.91	5.010	4.865	5.00	6143762
	2018-04-23	4.81	4.880	4.790	4.84	5179080
	2018-04-24	4.80	4.915	4.730	4.83	5382071
	2018-04-25	4.75	4.810	4.620	4.70	8588249
	2018-04-26	4.85	4.890	4.740	4.78	3209755

Figura 03 - Morningstar

```
In [24]: import pandas_datareader.data as web
import datetime as dt
start = dt.datetime(2018, 4, 20)
end = dt.datetime(2018, 4, 27)
f = web.DataReader('GGB', 'iex', start, end)
f.head()
```

Out[24]:

		open	high	low	close	volume
	date					
	2018-04-20	5.00	5.010	4.865	4.91	6143762
	2018-04-23	4.84	4.880	4.790	4.81	5179080
	2018-04-24	4.83	4.915	4.730	4.80	5382071
	2018-04-25	4.70	4.810	4.620	4.75	8588249
	2018-04-26	4.78	4.890	4.740	4.85	3209755

Figura 04 - IEX

```
In [4]: import pandas_datareader.data as web
import datetime as dt
start = dt.datetime(2018, 4, 20)
end = dt.datetime(2018, 4, 27)
f = web.DataReader('GGB', 'robinhood', start, end)
f.head()

Out[4]:
```

symbol	begins_at	close_price	high_price	interpolated	low_price	open_price	session	volume
GGB	2017-05-08	2.8213	2.8908	False	2.7810	2.8412	reg	4804204
	2017-05-09	2.9305	2.9504	False	2.8412	2.9511	reg	7044823
	2017-05-10	3.0100	3.0597	False	3.0001	3.0299	reg	9590679
	2017-05-11	3.1094	3.1293	False	2.9604	3.0001	reg	6170612
	2017-05-12	3.0100	3.1591	False	3.0001	3.1293	reg	6106002

Figura 05 - Robinhood

02 – Algumas API foram descontinuadas, tais como:

```
In [22]: import pandas_datareader.data as web
import datetime as dt
start = dt.datetime(2018, 4, 20)
end = dt.datetime(2018, 4, 27)
f = web.DataReader('GGB', 'yahoo', start, end)
f.head()

NameError: Traceback (most recent call last)
<ipython-input-20-cf8b6a1c7d70 in <module>()
      1 start = dt.datetime(2018, 4, 20)
      2 end = dt.datetime(2018, 4, 27)
----> 3 f = web.DataReader('GGB', 'yahoo', start, end)
      4 f.head()

C:\Users\Marcelo\Anaconda2\lib\site-packages\pandas_datareader\data.py in __init__(self, name, data_source, start, end, retry_count, pause, session, access_key)
    201     """
    202     if data_source == 'yahoo':
-> 203         raise NotImplementedError("YF API is deprecated. Use format('Yahoo Daily')")
    204         return YahooData(self, name, start, end, retry_count, pause, session, access_key, chunksize=10)
    205         return YahooData(self, name, start, end, retry_count, pause, session, access_key, chunksize=10)

NameError: Traceback (most recent call last)
Yahoo Daily has been immediately deprecated due to large breaks in the API without the introduction of a stable replacement. Pull Requests to re-enable these data sources are welcome.

See https://github.com/pydata/pandas-datareader/issues
```

Figura 06 - Yahoo finance

```
In [31]: import pandas_datareader.data as web
import datetime as dt
start = dt.datetime(2018, 4, 20)
end = dt.datetime(2018, 4, 27)
f = web.DataReader('GGB', 'google', start, end)
f.head()

NameError: Traceback (most recent call last)
<ipython-input-20-cf8b6a1c7d70 in <module>()
      1 start = dt.datetime(2018, 4, 20)
      2 end = dt.datetime(2018, 4, 27)
----> 3 f = web.DataReader('GGB', 'google', start, end)
      4 f.head()

C:\Users\Marcelo\Anaconda2\lib\site-packages\pandas_datareader\data.py in __init__(self, name, data_source, start, end, retry_count, pause, session, access_key)
    201     """
    202     if data_source == 'yahoo':
-> 203         raise NotImplementedError("YF API is deprecated. Use format('Yahoo Daily')")
    204         return YahooData(self, name, start, end, retry_count, pause, session, access_key, chunksize=10)
    205         return YahooData(self, name, start, end, retry_count, pause, session, access_key, chunksize=10)

NameError: Traceback (most recent call last)
Yahoo Daily has been immediately deprecated due to large breaks in the API without the introduction of a stable replacement. Pull Requests to re-enable these data sources are welcome.

See https://github.com/pydata/pandas-datareader/issues
```

Figura -07 Google finance

03 – Algumas API possuem requisitos específicos para acesso e estes dependem de informações sensíveis, tais como:

```
In [25]: import pandas_datareader.data as web
import datetime as dt
start = dt.datetime(2018, 4, 20)
end = dt.datetime(2018, 4, 27)
f = web.DataReader('GGB', 'tiingo', start, end)
f.head()

ValueError: Traceback (most recent call last)
<ipython-input-25-f2679d7d2b42 in <module>()
      1 start = dt.datetime(2018, 4, 20)
      2 end = dt.datetime(2018, 4, 27)
----> 3 f = web.DataReader('GGB', 'tiingo', start, end)
      4 f.head()

C:\Users\Marcelo\Anaconda2\lib\site-packages\pandas_datareader\data.py in DataReader(name, data_source, start, end, retry_count, pause, session, access_key)
    398     """
    399     if data_source == 'tiingo':
-> 400         raise ValueError("TIINGO API key must be provided either 'through the api_key variable or through the 'environmental variable TIINGO_API_KEY.'")
    401         return TiingoData(self, name, start, end, retry_count, pause, session, access_key, chunksize=10)
    402         return TiingoData(self, name, start, end, retry_count, pause, session, access_key, chunksize=10)

C:\Users\Marcelo\Anaconda2\lib\site-packages\pandas_datareader\tiingo.py in __init__(self, symbols, start, end, retry_count, pause, session, frag, api_key)
    61     """
    62     if not api_key or not isinstance(api_key, str):
-> 63         raise ValueError("The tiingo API key must be provided either 'through the api_key variable or through the 'environmental variable TIINGO_API_KEY.'")
    64         return TiingoData(self, symbols, start, end, retry_count, pause, session, frag, api_key, chunksize=10)
    65         return TiingoData(self, symbols, start, end, retry_count, pause, session, frag, api_key, chunksize=10)

ValueError: The tiingo API key must be provided either through the api_key variable or through the environmental variable TIINGO_API_KEY.
```

Figura 08 – Tiingo

```

In [8]: import pandas_datareader.data as web
import datetime as dt
start = dt.datetime(2018, 4, 20)
end = dt.datetime(2018, 4, 27)
f = web.DataReader('GOB', 'enigma', start, end)
f.head()

ValueError: Traceback (most recent call last):
<ipython-input-8-2b645b43d0c> in <module>()
      3 start = dt.datetime(2018, 4, 20)
      4 end = dt.datetime(2018, 4, 27)
----> 5 f = web.DataReader('GOB', 'enigma', start, end)
      6 f.head()

C:\Users\Marcelo\Anaconda2\lib\site-packages\pandas_datareader\data.py in DataReader(name, data_source, start,
end, retry_count, pause, session, access_key)
    347
    348     elif data_source == "enigma":
--> 349         return EnigmaReader(dataset_id=name, api_key=access_key).read()
    350
    351     elif data_source == "fred":

C:\Users\Marcelo\Anaconda2\lib\site-packages\pandas_datareader\enigma.py in __init__(self, dataset_id, api_key,
retry_count, pause, session)
     41     self._api_key = os.getenv('ENIGMA_API_KEY')
     42     if self._api_key is None:
--> 43         raise ValueError("Please provide an Enigma API key or set "
     44                        "the ENIGMA_API_KEY environment variable\n"
     45                        "If you do not have an API key, you can get "

ValueError: Please provide an Enigma API key or set the ENIGMA_API_KEY environment variable
If you do not have an API key, you can get one here: http://public.enigma.com/signup

```

Figura 09 – Enigma

04 – Algumas API não estão implementadas em PANDAS e, por isso requerem uma codificação específica, e tal qual o item 03, dependem de informações sensíveis, tais como:

```

In [21]: import pandas_datareader.data as web
from datetime import datetime
start = datetime(2018, 4, 20)
end = datetime(2018, 4, 27)
f = web.DataReader('SID', 'blipapi', start, end)
f.head()

NotImplementedError: Traceback (most recent call last):
<ipython-input-21-4ec733af01fe> in <module>()
      3 start = datetime(2018, 4, 20)
      4 end = datetime(2018, 4, 27)
----> 5 f = web.DataReader('SID', 'blipapi', start, end)
      6 f.head()

C:\Users\Marcelo\Anaconda2\lib\site-packages\pandas_datareader\data.py in DataReader(name, data_source, start, end, retry_
count, pause, session, access_key)
    401     else:
    402         msg = "data_source=%r is not implemented" % data_source
--> 403         raise NotImplementedError(msg)
    404
    405

NotImplementedError: data_source='blipapi' is not implemented

```

Figura 10 – Bloomberg

05 – Algumas API estrangeiras possuem informações sobre ações listadas na B3 Ibovespa, porém o acesso só é livre para ações listadas na Dow Jones/NASDAQ, para as demais, como é o caso da B3 Ibovespa, ele é pago, como é o caso do Quandl.

06 – As API nacionais são poucas e são pagas, tais como o Crystal Data Feed – API (Financial Service On-Demand) da cedrotech [4].

Ou seja, um programa que obtenha a cotação de uma ação listada na B3 Ibovespa, em tempo real ou não, dependerá da inserção de informações sensíveis no código e isto impede que este projeto siga pela análise por API, tal qual inicialmente proposto, para ações listadas na B3 Ibovespa. No entanto, isto não impede a análise das ADR (American Depositary Receipt) destas ações, quando se tratar de ações listadas na Dow Jones.

Portanto, uma análise de ações B3 Ibovespa passa necessariamente pela leitura de dados exportados de sites especializados que possuem portais dedicados ao mercado

brasileiro, tais como o Infomoney e o Investing e, a partir das planilhas importadas destes, poder prever uma tendência de preços diários, para análises swingtrade ou superiores.

```

In [15]: import pandas as pd

def read_data():
    df = pd.read_csv("data/Historico_GGBR4.csv")
    print(df.tail())

if __name__ == "__main__":
    read_data()

```

	Data	Historico	Fech.	Var.Dia (%)	Abertura	Minimo	Medio	\
18	06/04/2018	16.13	16.13	-1.53	16.26	15.96	16.18	
19	05/04/2018	16.38	16.38	6.71	15.89	15.73	16.10	
20	04/04/2018	15.35	15.35	-2.42	15.26	15.18	15.35	
21	NaN	NaN	NaN	NaN	NaN	NaN	
22	NaN	NaN	NaN	NaN	NaN	NaN	

	Maximo	Volume	Negocios
18	16.49	149247604.0	13384
19	16.40	206060024.0	16382
20	15.51	114960276.0	14723
21	NaN	NaN	NaN	
22	NaN	NaN	NaN	

Figura 11 – Leitura de GGBR4 obtida pelo arquivo *Historico_GGBR4* fornecido por Infomoney

Conjuntos de dados e entradas

Os dados para esta análise são basicamente os obtidos em sites especializados, em formato *.csv, conforme a ação pretendida, cujo período de análise coincide com o tipo de análise técnica de investimento de interesse.

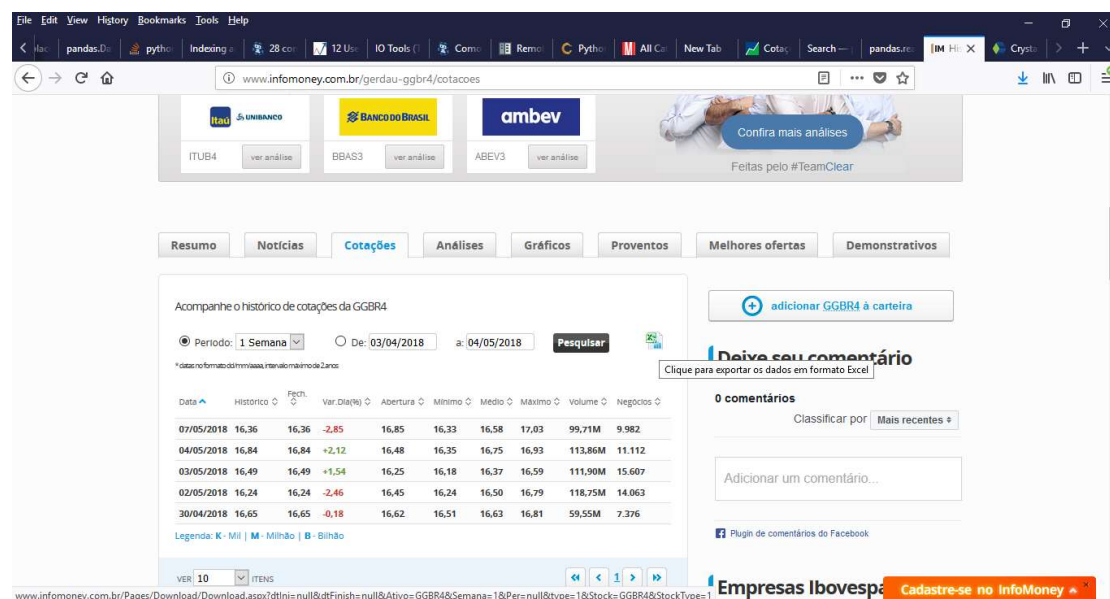


Figura 12 – Obtenção do arquivo *Historico_GGBR4* fornecido por Infomoney. [5]

A escolha por este portal se deve a disposição de informações do arquivo *.csv fornecido e as poucas modificações que requerem para obter um modelo plenamente ajustável ao pandas.

No entanto, para fins de coerência ao modelo proposto pelo portal [1], para obter um modelo mais ajustável a uma interface GUI (objetivo posterior e já descrito), será mantido o modelo de API também e este seguirá, sempre que possível, as ADR (American Depositary Receipt) das ações brasileiras listadas na bolsa de Nova Iorque (Dow Jones). Esta forma de trabalho é possível tão somente porque o comportamento das ADR seguem o mesmo padrão das ações listadas na B3 Ibovespa e isto ocorre em grande parte ao fato de que os investidores são os mesmos.

Descrição da solução

O projeto consiste na criação de uma primeira etapa de um código amplo que permita predizer o valor de uma ação baseado em seu histórico anterior, conforme o período escolhido, cujo resultado possa ser comparado com a análise técnica convencional.

O termo primeira etapa se refere ao fato de que os dados ainda possuem inserção manual e visa somente validar o modelo e, posteriormente o modelo terá uma interface GUI.

Modelo de referência (benchmark)

O modelo de referência passa basicamente pela comparação dos resultados entre o obtido por meio de ferramentas próprias para análise de gráficos, tais como:



Figura 13 – Bandas de Bollinger de USIM5. [6]

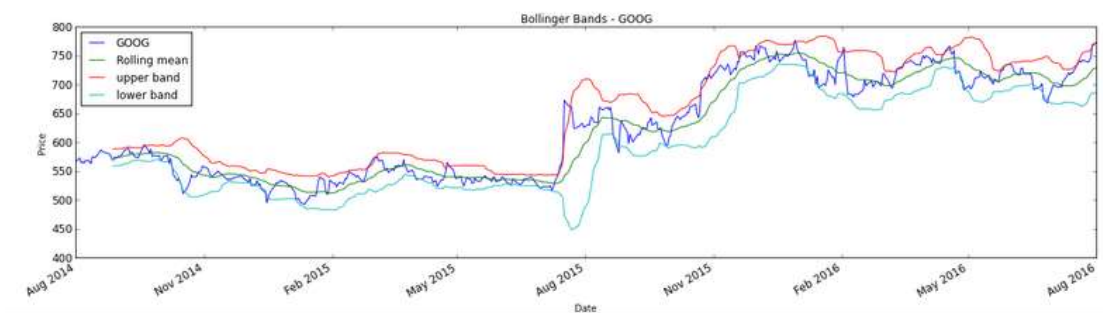


Figura 14 – Bandas de Bollinger de GOOG.

Métricas de avaliação

A métrica de avaliação é bastante simples pois consiste em comparar o valor encontrado pelas regressões lineares propostas para um determinado dia, comparar com o valor previsto pelas análises técnicas convencionais e com o valor real da ação para a data prevista, onde:

Etapa 01: Comparar o valor da ação no dia em relação ao predito pela regressão linear;

Etapa 02: Comparar o valor da ação no dia em relação ao predito pela regressão KNN;

Etapa 03: Comparar o valor da ação no dia em relação ao predito pela análise MACD;

Etapa 04: Comparar o valor da ação no dia em relação ao predito pelas curvas de bollinger;

Etapa 05: Comparar o valor das predições por regressão e pelas análises técnicas de uma ação no dia e obter o erro médio das predições.

O objetivo é obter a eficiência do modelo.

Resultados obtidos pela modelagem:

```
IEX_stock_GGB_chart_1m
Preco inicial: 4.56
Preco final: 4.61
Ganho: 1.09649122807 %
```

```
IEX_stock_SID_chart_1m
Preco inicial: 2.44
Preco final: 2.43
Ganho: -0.409836065574 %
```

```
IEX_stock_VALE_chart_1m
Preco inicial: 12.5
Preco final: 13.89
Ganho: 11.12 %
```

```
Média:
IEX_stock_GGB_chart_1m      0.000749
IEX_stock_SID_chart_1m      0.000101
IEX_stock_VALE_chart_1m     0.005757
dtype: float64
```

```
Daytrade - desvio padrão:
IEX_stock_GGB_chart_1m      0.019284
IEX_stock_SID_chart_1m      0.026130
IEX_stock_VALE_chart_1m     0.020268
dtype: float64
```

Design do projeto

A partir dos dados obtidos no formato *.CSV, estando todos no mesmo período de análise e na mesma formatação, o que pressupõe mesma origem, os dados são

apresentados sobrepostos, se referem a um mesmo segmento de mercado, neste caso o mercado de aço, pois este oscila por igual, tem abrangência mundial, e os três principais entes brasileiros deste mercado possuem ADR na Dow Jones.

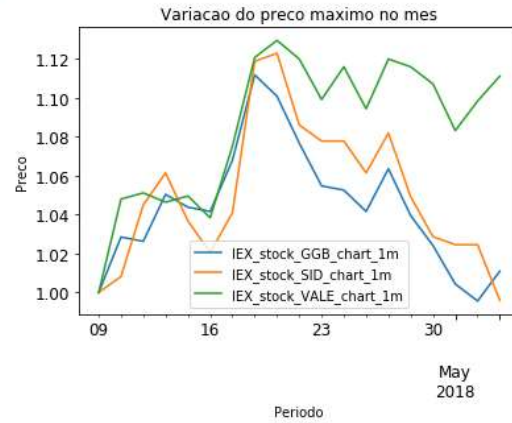


Figura 15 – Sobreposição das ADR de aço no mês de Abril/18.

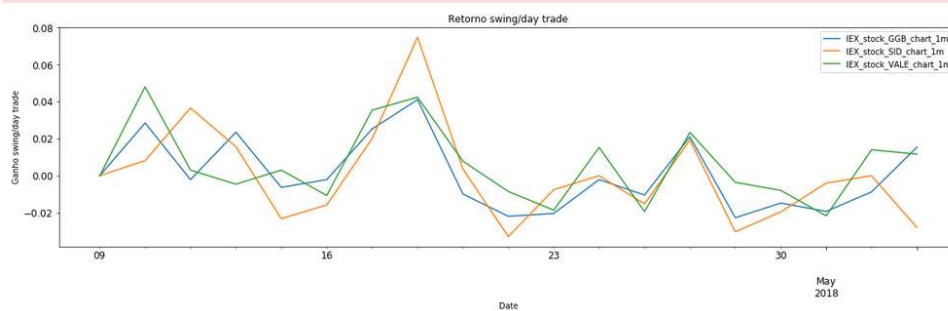


Figura 16 – Retorno de operações com as ADR de aço no mês de Abril/18.

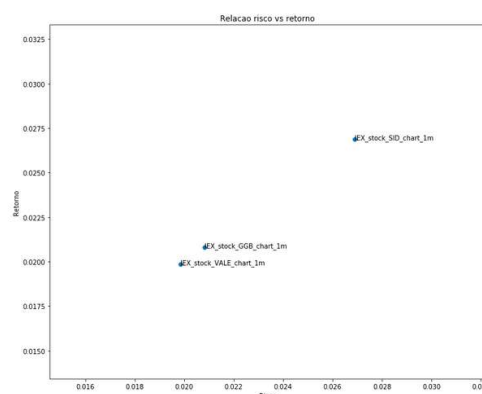


Figura 17 – Relação Risco/Retorno de operações com as ADR de aço no mês de Abril/18.

Bibliografia

- [1] <https://docs.google.com/document/d/1ycGeb1QYKATG6jvz74SAMqxrlek9Ed4RYrzWNhWS-0Q/pub>
- [2] <https://br.advfn.com/educacional/analise-tecnica/macd>
- [3] <http://meutrade.com.br/bandas-de-bollinger/>
- [4] http://promo.cedrotech.com/crystal-data-feed-solucoes-de-market-data?utm_source=InfoMoneyDiaria&utm_medium=referral&utm_campaign=APIsCrystal
- [5] <http://www.infomoney.com.br/gerdau-ggbr4/cotacoes>
- [6] <https://iextrading.com/apps/stocks/#/>
- [7] https://pandas-datareader.readthedocs.io/en/latest/remote_data.html#remote-data-google
- [8] https://github.com/pydata/pandas-datareader/blob/master/docs/source/remote_data.rst
- [9] <https://paulovasconcellos.com.br/28-comandos-%C3%BAteis-de-pandas-que-talvez-voc%C3%AA-n%C3%A3o-conhe%C3%A7a-6ab64beefa93>