Melhores ações para 2020: veja boas opções para investir

Análise de uma ação negociadas no mercado brasileiro com data sciente, escolhidas com base

Referências para a escolha das ações:

https://blog.toroinvestimentos.com.br/melhores-acoes-2020

Empresas de energia elétrica: Taesa - TAEE11

Double-click (or enter) to edit

- Vale ressaltar que a intenção é aplicação de dataScience.

Importaremos uma figura para ilustrar o assunto: de Análise Técnica:

```
from google.colab import files
from IPython.display import Image

upload = files.upload()
```

Escolher arquivos Nenhum arquivo selecionado Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving analise_tecnica.jpg to analise_tecnica.jpg
```

```
Image('analise_tecnica.jpg', width = 1000)
```

```
! pip install pandas_datareader
```

```
Requirement already satisfied: pandas_datareader in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: lxml in /usr/local/lib/python3.6/dist-packages (from p
Requirement already satisfied: wrapt in /usr/local/lib/python3.6/dist-packages (from 
Requirement already satisfied: pandas>=0.19.2 in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python3.6/dist
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dis
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (fr
```

```
! pip install plotly
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.6/dist-packages (4.4.
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from pl
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.6/dist-packa
```

```
# importar as bibliotecas necessárias:
```

```
# importar as bibliotecas necessárias:
import pandas as pd
from pandas_datareader import data as web
import plotly.graph_objects as go

# criar um DataFrame vazio:
df = pd.DataFrame()


# escolha da ação website:
acao = 'TAEE11.SA'

# importação de dados para o DataFrame:
df = web.DataReader(acao, data_source='yahoo', start='01-01-2000')

# ver as 5 primeiras entradas
df.head()
```

	High	Low	Open	Close	Volume	Adj Close
Date						
2019-03-07	25.000000	24.360001	24.719999	24.490000	1430100.0	22.865997
2019-03-11	25.790001	25.100000	25.100000	25.540001	1217800.0	23.846367
2019-03-12	26.059999	25.350000	25.580000	25.980000	1652800.0	24.257185
2019-03-13	26.280001	25.700001	25.879999	26.180000	1514800.0	24.443928
2019-03-14	26.360001	25.830000	26.230000	25.990000	1348100.0	24.266523

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 215 entries, 2019-03-07 to 2020-02-18
Data columns (total 6 columns):
High         215 non-null float64
Low          215 non-null float64
Open         215 non-null float64
Close        215 non-null float64
Volume       215 non-null float64
Adj Close    215 non-null float64
dtypes: float64(6)
memory usage: 11.8 KB
```

```
df.shape
```

```
(215, 6)
```

```
df.isnull().sum()
```

```
High         0
Low          0
Open         0
Close        0
Volume       0
Adj Close    0
dtype: int64
```

```
df.count()
```

```
High         215
Low          215
Open         215
Close        215
Volume       215
Adj Close    215
dtype: int64
```

```
df.describe()
```

	High	Low	Open	Close	Volume	Adj Close
count	215.000000	215.000000	215.000000	215.000000	2.150000e+02	215.000000
mean	28.310047	27.790791	28.094326	28.066884	1.266501e+06	27.226975
std	1.879815	1.891453	1.891663	1.884336	5.721058e+05	2.399039
min	24.629999	23.559999	23.799999	24.100000	3.670000e+05	22.501858
25%	27.185000	26.715000	26.900000	26.860001	9.460000e+05	25.582061
50%	28.309999	27.850000	28.110001	28.090000	1.173700e+06	27.274223
75%	29.090000	28.634999	28.889999	28.870000	1.467200e+06	28.246074
max	32.349998	31.870001	32.299999	32.160000	6.266100e+06	32.160000

```
df[df['High'] == df['High'].min()]
```

	High	Low	Open	Close	Volume	Adj Close
Date						
2019-03-27	24.629999	23.85	24.59	24.1	1240200.0	22.501858

```
df[df['High'] == df['High'].max()]
```

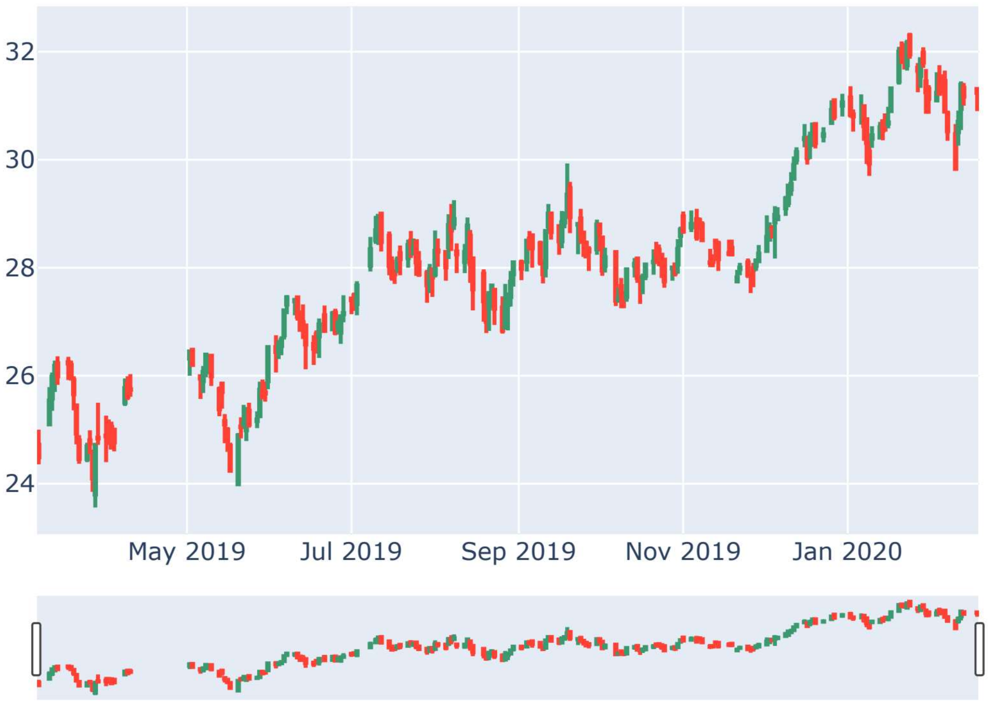
	High	Low	Open	Close	Volume	Adj Close
Date						
2020-01-24	32.349998	31.870001	32.299999	31.950001	627200.0	31.950001

```
# plotar o gráfico de candlestick
```

```
trace1 = {
    'x': df.index,
    'open': df.Open,
    'close': df.Close,
    'high': df.High,
    'low': df.Low,
    'type': 'candlestick',
    'name': acao,
    'showlegend': False
}

data = [trace1]
layout = go.Layout()

fig = go.Figure(data=data, layout=layout)
fig.show()
```



#https://readthedocs.org/projects/pandas-datareader/downloads/pdf/stable/

#https://www.youtube.com/watch?v=uXh-EsWLpyw