



Trabalho prático de Machine Learning - Buying and selling operations on the B3 stock exchange

Centro Federal de Educação Tecnológica de Minas Gerais

Grupo: Arthur E. Machado, Guilherme G. Giacomini, Uriel B. Braga, Rubio Torres

Data: 04/04/2020

Introdução

O mercado de ações é um ramo da economia muito importante que gera interesse de investidores com cifras bilionárias em todo mundo. Trata-se de um mercado de compra e venda de porções de empresas de capital aberto, a fim de captar recursos e investimentos para essas empresas. A bolsa de valores nada mais é do que o espaço de acesso ao mercado acionário, que irá fazer uma ligação entre investidores e empresas por meio de ações, também conhecidas como ativos ou papéis.

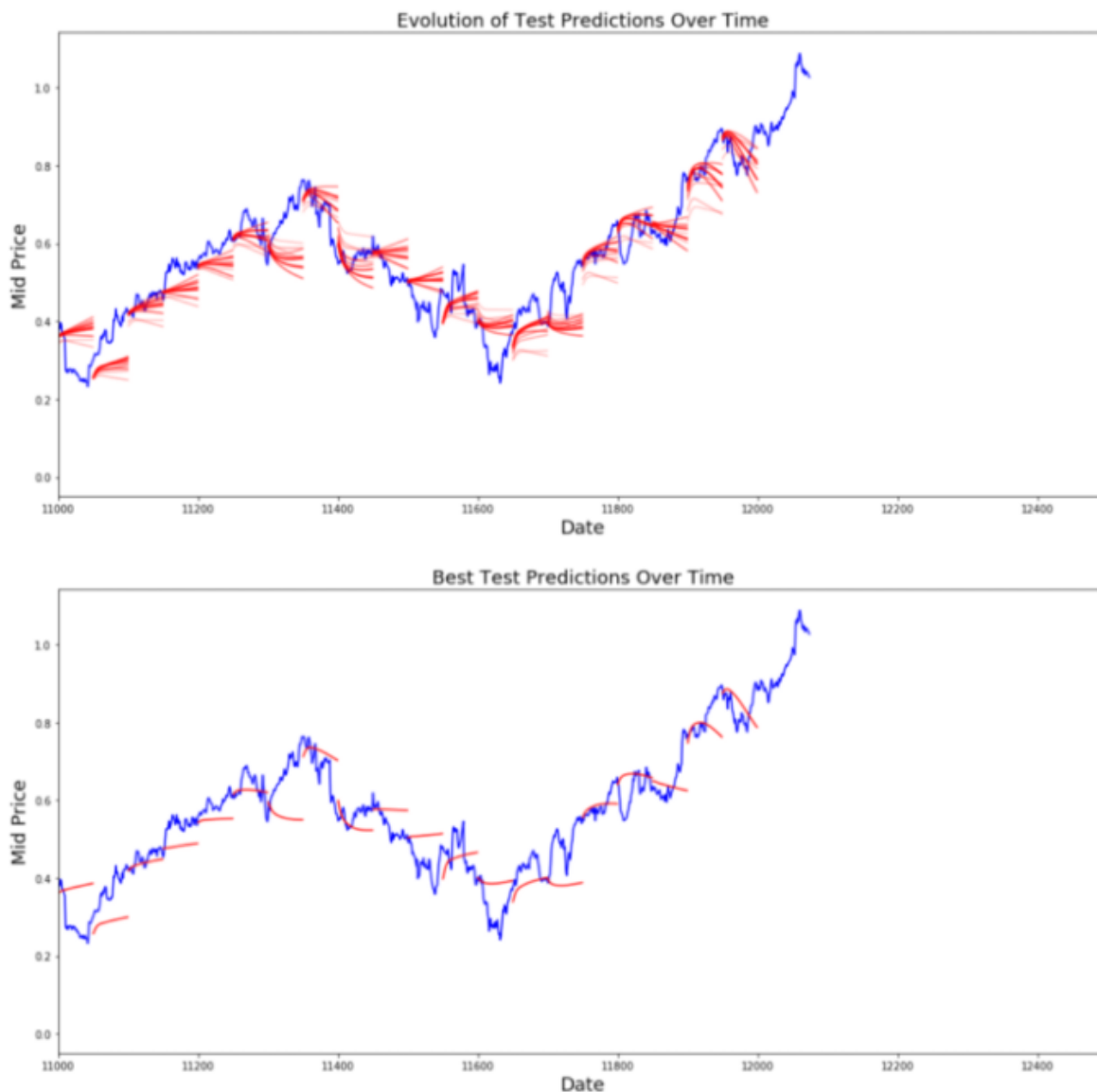
No Brasil, a bolsa de valores (B3) movimenta mais de 10 bilhões de reais por dia (em 2021) e justamente por se tratar de um mercado muito rentável, os investidores sempre buscam tomar as melhores decisões, com os menores riscos, maiores lucros e maiores retornos provenientes de aporte financeiro. Portanto, utilizar Machine Learning, Inteligência Artificial e outros recursos computacionais pode ser muito interessante para auxiliar tais investidores a tomarem as boas decisões e diminuir o aporte financeiro em ativos de alto risco ou baixo retorno.

O problema

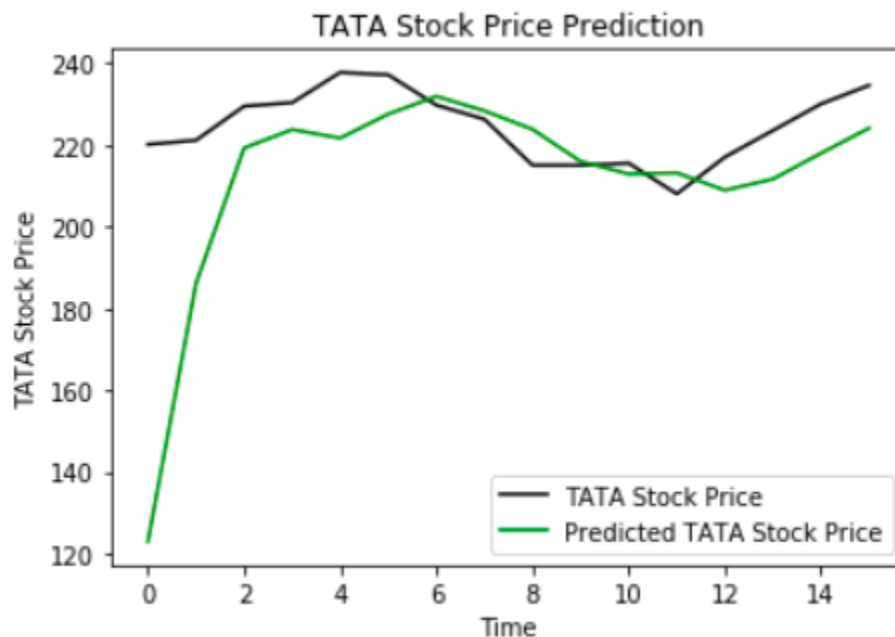
Portanto o problema a ser resolvido é: utilizar técnicas de machine learning para prever índices e valores econômicos da bolsa de valores brasileira (B3) e com isso, fazer previsões razoáveis a cerca de compras e vendas de ativos para maximizar o lucro, minimizar as perdas. Para isso, o modelo precisará de algumas características, dados e *features* de determinado conjunto de ativos.

Trabalhos relacionados

Essa implementação foi inspirada em uma pesquisa e análise em algumas soluções já existentes. A primeira delas se chama [Using LSTMs For Stock Market Predictions \(Tensorflow\)](#) e utiliza as Long Short-Term Memory (LSTM) para realizar uma abordagem colocando em foco o histórico da série de dados para prever os elementos futuros da sequência corretamente. Esse trabalho realiza um pré-processamento nos dados utilizando a técnica MinMaxScalar, que deixa todos os valores numa região entre 0 e 1, mas com os mesmos divididos em janelas. A avaliação do modelo foi feita por meio do Mean Squared Error (MSE) ou "erro médio quadrado" e durante a passagem das épocas de treinamento, essa medida foi caindo consideravelmente. O autor ressaltou que essa solução foi conseguida prever o resultado da bolsa de valores corretamente em boa parte do tempo, apesar de ser imperfeita.



Um outro trabalho analisado foi o Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices que concebeu um modelo bem parecido (LSTM) para prever a bolsa de valores utilizando o Keras - uma biblioteca de rede neural de código aberto escrita em Python. O autor também utilizou o MinMaxScalar do Sckitlearn para fazer o Feature Scaling e além disso fez um ajuste de dividir os dados em 60 Timesteps para que a entrada da rede LSTM seja em vetor em 3D. Os resultados desse trabalho no gráfico abaixo mostram que o valor da predição se assemelhou muito com o valor real, principalmente nos instantes de tempo finais, o que mostra como a rede LSTM é poderosa para analisar séries temporais e dados sequenciais.



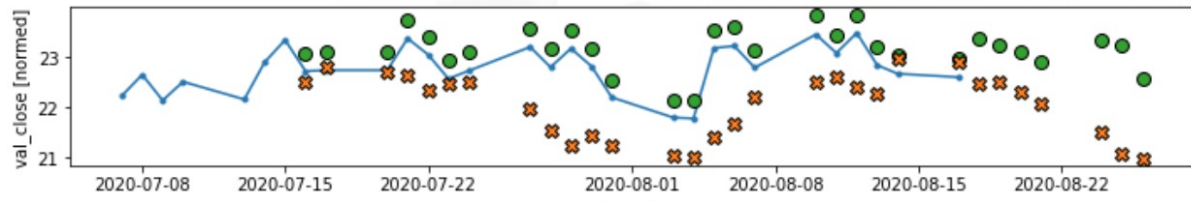
Pré-processamento

O pré-processamento utilizado neste modelo será a regularização por normalização que é o quanto uma medida se afasta da média em termos de Desvios Padrão. Quando o valor é positivo isto indica que o dado está acima da média e quando o mesmo é negativo significa que o dado está abaixo da média.

A normalização deve ser calculada para cada elemento a partir da fórmula abaixo, em que M é a média dos valores e σ é o desvio padrão.

$$x_{Normalizado_i} = \frac{x_i - M}{\sigma}$$

Além disso, os dados foram submetidos à divisão em janelas por meio do método *WindowGenerator* que gera essas fatias do conjunto de dados. Com isso, cria-se uma janela deslizante para que o treinamento e validação dos dados utilize intervalos de 7 dias, conforme mostra o gráfico abaixo:



Além disso, ainda como ferramenta de pré-processamento dessas janelas, utilizou-se o método `tf.keras.preprocessing.timeseries_dataset_from_array` para converter um *Dataframe* do Pandas para um objeto `TensorFlow.data.Dataset` do *TensorFlow*, conforme sugerido no [tutorial](#) dessa biblioteca. Por fim, vale ressaltar que o objeto *WindowGenerator* contém dados de treinamento, validação e teste.

```
data=data,
targets=None,
sequence_length=self.total_window_size,
sequence_stride=1,
shuffle=True,
batch_size=32,)
```

parâmetro passados para o método `timeseries_dataset_from_array`