Previsão do Valor do Fechamento Ajustado Para Ações

Nanodegree Engenheiro de Machine Learning

Projeto final

Thiago Henrique Martins de Souza

16 de abril de 2019

I. Definição do problema

Visão geral

Empresas de investimento, hedge funds e até indivíduos têm usado modelos financeiros para entender melhor o comportamento do mercado e fazer investimentos e negócios lucrativos. Uma riqueza de informações está disponível na forma de preços de ações históricos e dados de desempenho da empresa, adequados para algoritmos de aprendizado de máquina a serem processados.

Declaração do problema

Realizar a previsão do valor do fechamento ajustado para ações de um determinado período, utilizando dados históricos como entrada. Um script simples deve ser capaz de prever o valor de ações e compará-lo com os valores reais e mostrar isso através de um gráfico e cálculo da média total.

Métricas

Para avaliar o desempenho do projeto, podemos verificar o valor da média de erro absoluta entre valor real e valor previsto e será gerado um gráfico com os valores reais e os valores previstos. A intenção foi acertar o maior número de vezes o valor da ação e/ou tentar ficar o mais próximo possível do valor real da ação, diminuindo a média de absoluta de erro. Foi utilizada como métrica do preditor a média MAE (Mean Absolute Error) que deverá ser menor que 0,25.

II. Análise do problema

Exploração dos dados

Os dados foram extraídos da fonte Yahoo! Finanças, ações da Petrobras PETR4.SA no período de 01/01/2013 até 31/01/2017 e os dados para teste serão de 01/01/2018 até 31/01/2018. Arquivos .csv utilizados estão anexo ao projeto enviado.

Visualização exploratória

Os dados são importados e convertidos para Numpy Array.

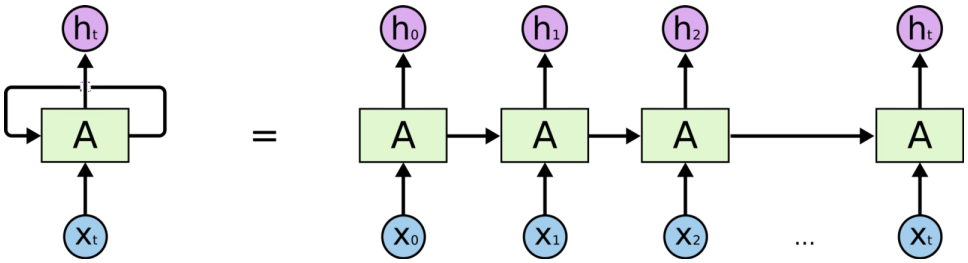
	Date	Open	High	Low	Close	Adj Close	Volume
0	2013-01-02	19.990000	20.209999	19.690001	19.690001	18.086271	30182600.0
1	2013-01-03	19.809999	20.400000	19.700001	20.400000	18.738441	30552600.0
2	2013-01-04	20.330000	20.620001	20.170000	20.430000	18.766001	36141000.0
3	2013-01-07	20.480000	20.670000	19.950001	20.080000	18.444506	28069600.0
4	2013-01-08	20.110001	20.230000	19.459999	19.500000	17.911745	29091300.0

O valor previsto será a coluna “Adj Close”.

Algoritmos e técnicas

Duas técnicas foram utilizadas, Redes Neurais Recorrentes (RNN) e Long Short Term Memory (LSTM). As redes recorrentes tomam como entrada não apenas o exemplo de entrada atual que veem, mas também o que perceberam anteriormente no tempo. A decisão de uma rede recorrente alcançada na etapa de tempo t-1 afeta a decisão que chegará um momento mais tarde na etapa de tempo t. Assim, as redes recorrentes têm duas fontes de entrada, o presente e o passado recente, que se combinam para determinar como respondem a novos dados, da mesma forma que fazemos na vida. Os LSTMs ajudam a preservar o erro que pode ser retropropagado através do tempo e das camadas. Ao manter um erro mais constante, eles permitem que redes recorrentes continuem aprendendo ao longo de muitos passos de tempo, abrindo assim um canal

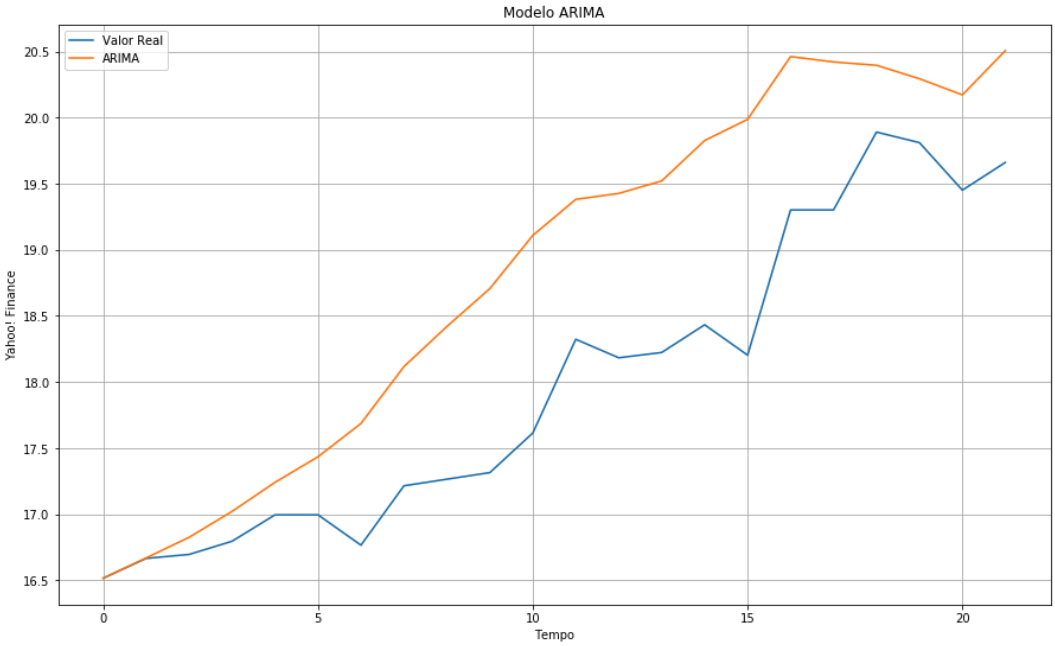
para vincular causas e efeitos remotamente. Esse tipo de modelo de rede neural é adequado para realizar previsões de série temporal, são redes utilizadas para reconhecer padrões quando os resultados do passado influenciam no resultado atual.



Modelo de referência

Serviram de referência os dados reais do período entre 01/01/2018 até 31/01/2018. As previsões foram realizadas com base nos dados do passado e comparadas com os dados de Janeiro de 2018.

Usarei um modelo ARIMA para prever os dados de Janeiro de 2018 e assim fazer comparação do desempenho do modelo de previsão de ações para o período.



III. Metodologia

Pré-processamento dos dados

Os dados são importados dos arquivos .csv disponíveis no projeto, e depois são convertidos em Numpy Array. Em seguida é selecionada apenas a coluna “Adj Close” que será nosso alvo para previsões. Utilizaremos os dados históricos para prever os valores. A normalização dos dados é feita utilizando MinMaxScaler() do sklearn. Bases de treinamento e testes são devidamente separadas.

Implementação

Utilizando Redes Neurais Recorrentes (RNN) e Long Short Term Memory (LSTM). É criado um regressor Sequential() do keras, e são adicionadas 4 camadas LSTM, a primeira com 100 unidades e as seguintes com 50 unidades de memória. Depois o regressor é ativado utilizando uma camada densa com ativação linear. Após o treinamento, os dados são preparados para a realização das previsões.

Utilizarei um intervalo de tempo de 90 dias para realizar a previsão do dia atual, ou seja, para cada valor previsto será considerado os 90 dias anteriores para a realização da previsão.

Refinamento

Foram criados 2 modelos com diferença na quantidade de unidades de memória utilizados. O primeiro modelo com 4 camadas, 50 unidades de memória na primeira camada e 30 nas camadas restantes e o segundo modelo com 4 camadas, 100 unidades de memória na primeira camada e 50 nas camadas restantes.

IV. Resultados

Avaliação e validação do modelo

A média absoluta do erro para os valores previstos para o período são:

MAE (Mean Absolute Error)

```
from sklearn.metrics import mean_absolute_error
# MSE Valor Real sobre as Previsões (não otimizada)
mean_absolute_error(preco_real_teste, previsoes1)
```

```
0.4948271728265933
```

```
# MSE Valor Real sobre as Previsões (otimizada)
mean_absolute_error(preco_real_teste, previsoes)
```

```
0.2298753828235973
```

```
# MSE Valor Real sobre o modelo ARIMA
mean_absolute_error(predictions_ARIMA, previsoes)
```

```
0.8945400589439759
```

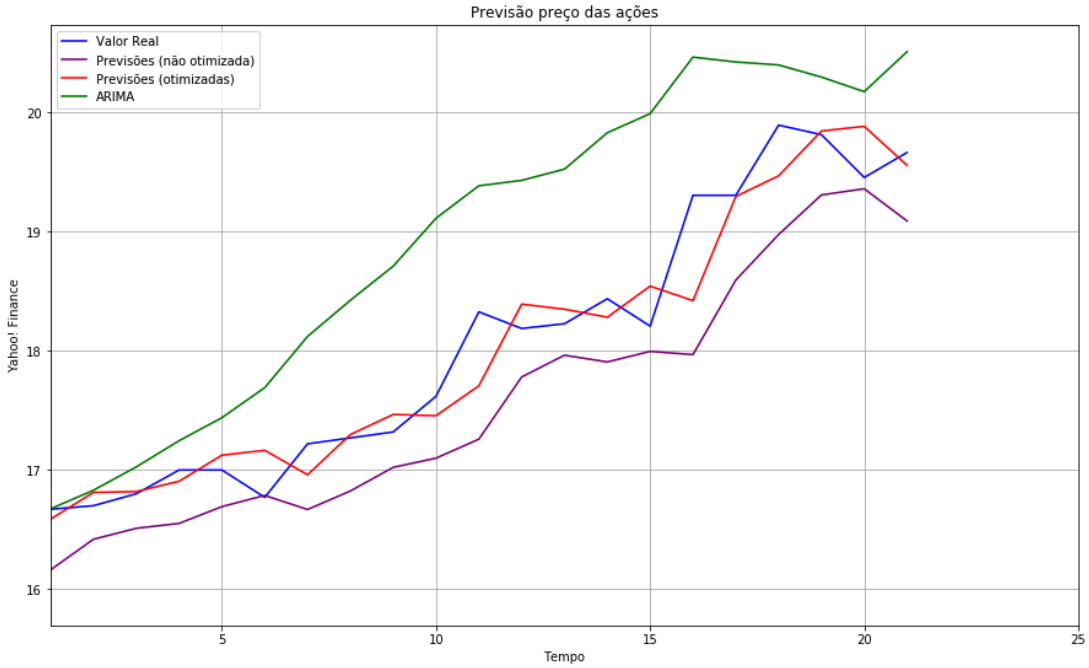
- Modelo com 4 camadas, 50 unidades de memória na primeira camada e 30 nas camadas restantes: MAE 0.4948271728265933

- Modelo otimizado com 4 camadas, 100 unidades de memória na primeira camada e 50 nas camadas restantes: MAE 0.2298753828235973

- Modelo ARIMA: MAE 0.8945400589439759

Melhor resultado obtido é do modelo otimizado.

Analisando o gráfico podemos notar que o modelo preditor se saiu melhor ao modelo ARIMA e ao modelo não otimizado, ou seja, teve um melhor desempenho.



Notamos também que houve uma subida nos valores das ações no período, bem como os valores previstos.

As previsões ficaram bem próximas do valor real. Comparação a métrica MAE para o modelo final verificamos o seguinte resultado:

```
from sklearn.metrics import mean_absolute_error
# MSE Valor Real sobre as Previsões
mean_absolute_error(preco_real_teste, previsoes)
```

```
0.2298753828235973
```

A solução apresentada funcionou bem para as ações e o período selecionado e só foi testada com esses dados. Os dados utilizados foram analisados em um curto período de tempo e a ação utilizada não teve muita variação, ou seja, existia uma tendência de subida.

Essa solução pode não se sair bem quando há grandes mudanças de comportamento, como uma queda repentina do valor das ações. Para que a solução esteja pronta seria necessário treiná-la e testá-la com outras ações, em outros períodos e com comportamentos diferentes.

Justificativa

O projeto pode ser justificado com base nas previsões realizadas e uma quantidade satisfatória de acertos. Comparando os resultados obtidos, o modelo se saiu bem ao prever os valores das ações. Verificando os valores utilizados para a previsão nota-se que os modelos se comportaram conforme o esperado.

V. Conclusão

Visualização de forma livre

O projeto foi desenvolvido para ser capaz de realizar previsões de ações da bolsa, utilizando dados históricos. Esses dados foram tratados e processados antes do treinamento do modelo de previsão, foram normalizados e os valores foram previstos. Como referência foi criado um modelo ARIMA simples e depois podemos verificar a eficácia através da média geral dos valores e do gráfico das previsões X valores reais X ARIMA.

Reflexão

O desenvolvimento do modelo não foi uma tarefa fácil, primeiramente os dados de entrada foram extraídos do Yahoo! Finance e depois foram pré-processados de maneira correta para que as previsões fossem realizadas. Foi escolhido o modelo de Redes Neurais Recorrentes para a previsão dos valores das ações, após uma pesquisa o modelo foi solucionado pois ele poderia obter bons resultados com séries temporais. Um grande desafio foi a criação do modelo de referência ARIMA. Precisei pesquisar e errar muito até chegar em um modelo satisfatório para ser

usado como referência. O modelo é bastante simples e muitos processos que envolvem o pré-processamento dos dados são manuais e não dinâmicos. Mesmo assim o projeto atingiu seu objetivo de forma satisfatória, entregando a previsão dos valores das ações muito próximo dos valores reais.

Aperfeiçoamento

O aperfeiçoamento do projeto pode ser um aumento da quantidade de camadas utilizadas pela rede neural. Podemos aumentar também a quantidade de units de memória e a quantidade de épocas processadas. Porém, essa alteração teria um custo computacional maior.

Referências bibliográficas

Keras LSTM: https://keras.io/layers/recurrent/
Python for Finance: https://www.learndatasci.com/tutorials/python-finance-part-yahoo-finance-api-pandas-matplotlib/
Skymind A.I. week: https://skymind.ai/wiki/lstm
Vooo – Insights: https://www.vooo.pro/insights/guia-completo-para-criar-time-series-com-codigo-em-python/