Chương 1: Dataset

Giá cả chứng khoán luôn liên tục biến động, không ngừng thay đổi do chịu ảnh hưởng của nhiều yếu tố cả vi mô lần vĩ mô, như chính trị, chiến tranh, kinh tế, tình hình tài chính công ty,,, Đồng nghĩa với việc có rất nhiều dữ liệu để dự đoán được giá chứng khoán. Trong đề tài này chúng em sẽ sử dụng kiến thức học máy đã học để dự đoán giá cổ phiếu.

Bộ dữ liệu bao gồm 30 mã cổ phiếu chứa thông tin về giá cả (open, high, low, close) và khối lượng giao dịch chứng khoán từ ngày 04/01/2021 đến ngày 29/10/2021 được lấy từ package [vnquant](https://github.com/phamdinhkhanh/vnquant.git).

Bài toán gồm các dataset:

* price\_train.csv: Giá cả và khối lượng giao dịch chứng khoán, dùng để training mô hình
* price\_test.csv: Tập dữ liệu test dùng để đánh giá kết quả mô hình trên Kaggle.
* business\_train.csv: Báo cáo kinh doanh.

finance\_train.csv: Báo cáo tài chính.

* sample\_submission.csv: Tệp mẫu kết quả nộp lên Kaggle (30% dữ liệu Public và 70% dữ liệu Private trên Leaderboard Score)

Trong đó mỗi dataset có các thuộc tính sau:

****price\_train****:

* date: thời gian giao dịch
* open: giá mở cửa (giá thực hiện tại lần khớp lệnh đầu tiên trong ngày giao dịch chứng khoán)
* high: giá cao nhất trong ngày
* low: giá thấp nhất trong ngày
* close: giá đóng cửa (giá thị trường của các cổ phiếu vào thời điểm đóng cửa một phiên giao dịch) (target\_variable)
* volume: khối lượng cổ phiếu giao dịch trong ngày (target variable)
* symbol: mã cổ phiếu

****business\_train****:

* index: chỉ số kết quả hoạt động kinh doanh
* các cột còn lại dạng YYYY-MM: tháng báo cáo tài chính, thường là 3, 6, 9, 12 hàng năm.

****finance\_train****:

* index: chỉ số tài chính
* các cột còn lại dạng YYYY-MM: tháng báo cáo tài chính, thường là 3, 6, 9, 12 hàng năm.

****price\_test****: Đây là dữ liệu chứa các ngày và mã chứng khoán tương ứng mà chúng ta cần phải dự báo giá close. Lưu ý sau khi dự báo trên template của price\_test thì phải chuyển sang template sample\_submission để nộp bài. Các trường bao gồm:

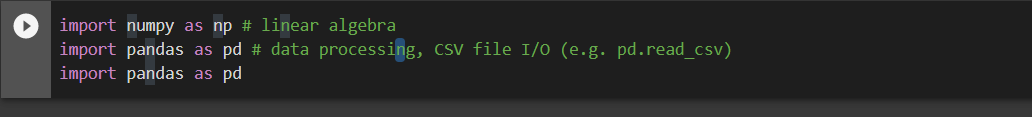
* date: ngày dự báo yyyy-mm-dd
* symbol: mã chứng khoán
* close: giá đóng cửa, đây là trường mục tiêu cần dự báo.

****sample\_submission****: Template được biến đổi từ price\_test, dùng để submit kết quả.

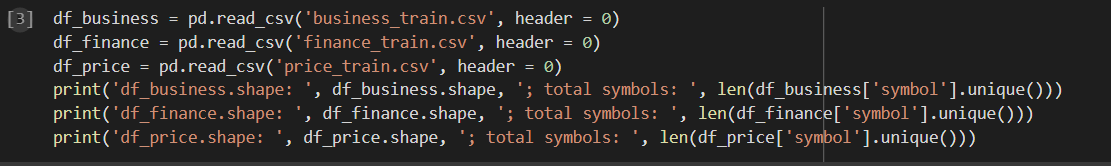
* Id: Mã id của một quan sát dự báo, là kết hợp giữa date và symbol ngăn cách bởi dấu ":".
* Predicted: giá trị dự báo của giá close tương ứng với Id.

Chương 2: Tiền xử lý dữ liệu

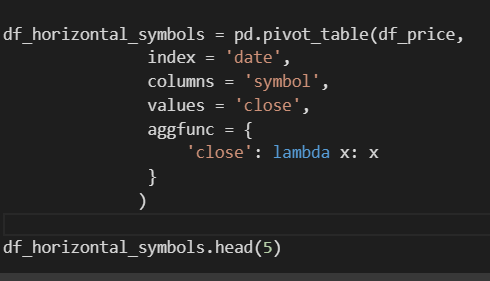
Import thư viện



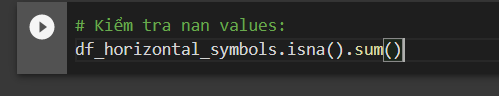
Đọc dữ liệu



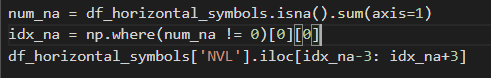
Để thuận tiện xử lý dữ liệu, chúng ta cần sắp xếp những mã chứng khoán này về dạng bảng có các cột là các mã chứng khoán và dòng là chuỗi thời gian.

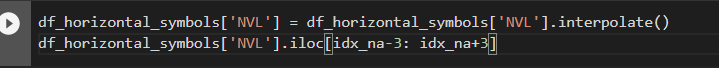


Kiểm tra nan values



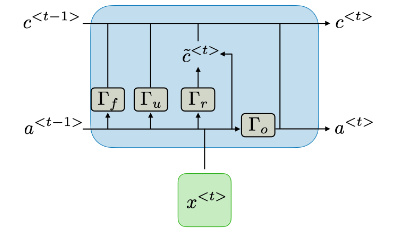
Do có cột NVL có một phần tử là NaN nên sử lý dữ liệu bằng phương pháp feed forward và sau đó sẽ fill dữ liệu bằng nội suy tuyến tính





Chương 3: Mô hình học máy

LTSM



Mạng trí nhớ ngắn hạn định hướng dài hạn còn được viết tắt là LSTM làm một kiến trúc đặc biệt của RNN. Bước đầu tiên trong LSTM sẽ quyết định xem thông tin nào chúng ta sẽ cho phép đi qua ô trạng thái (cell state). Nó được kiểm soát bởi hàm sigmoid trong một tầng gọi là tầng quên (forget gate layer).

Đầu tiên nó nhận đầu vào là 2 giá trị ht−1 và xt và trả về một giá trị nằm trong khoảng 0 và 1 cho mỗi giá trị của ô trạng thái Ct−1. Nếu giá trị bằng 1 thể hiện ‘giữ toàn bộ thông tin’ và bằng 0 thể hiện ‘bỏ qua toàn bộ chúng’.



Bước tiếp theo chúng ta sẽ quyết định loại thông tin nào sẽ được lưu trữ trong ô trạng thái. Bước này bao gồm 2 phần. Phần đầu tiên là một tầng ẩn của hàm sigmoid được gọi là tầng cổng vào (input gate layer) quyết định giá trị bao nhiêu sẽ được cập nhật. Tiếp theo, tầng ẩn hàm tanh sẽ tạo ra một véc tơ của một giá trị trạng thái mới C~t mà có thể được thêm vào trạng thái. Tiếp theo kết hợp kết quả của 2 tầng này để tạo thành một cập nhật cho trạng thái.



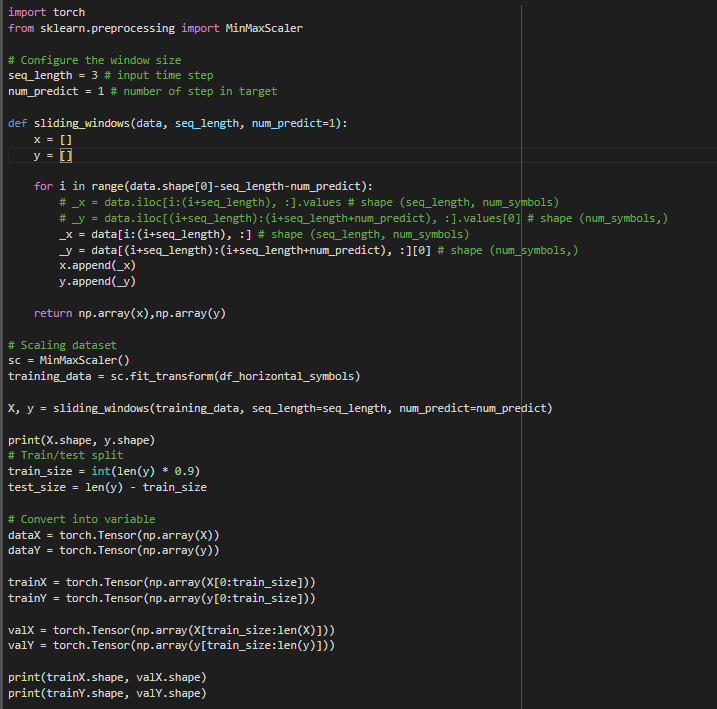
Đây là thời điểm để cập nhật một ô trạng thái cũ, Ct−1 sang một trạng thái mới Ct. Những bước trước đó đã quyết định làm cái gì, và tại bước này chỉ cần thực hiện nó.

Chúng ta nhân trạng thái cũ với ft tương ứng với việc quên những thứ quyết định được phép quên sớm. Phần tử đề cử it∗C~t là một giá trị mới được tính toán tương ứng với bao nhiêu được cập nhật vào mỗi giá trị trạng thái.



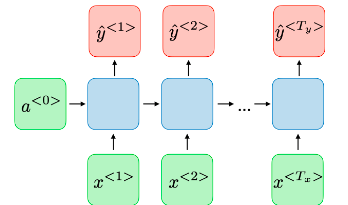
Cuối cùng cần quyết định xem đầu ra sẽ trả về bao nhiêu. Kết quả ở đầu ra sẽ dựa trên ô trạng thái, nhưng sẽ là một phiên bản được lọc. Đầu tiên, chúng ta chạy qua một tầng sigmoid nơi quyết định phần nào của ô trạng thái sẽ ở đầu ra. Sau đó, ô trạng thái được đưa qua hàm tanh (để chuyển giá trị về khoảng -1 và 1) và nhân nó với đầu ra của một cổng sigmoid, do đó chỉ trả ra phần mà chúng ta quyết định.

Xây dựng model

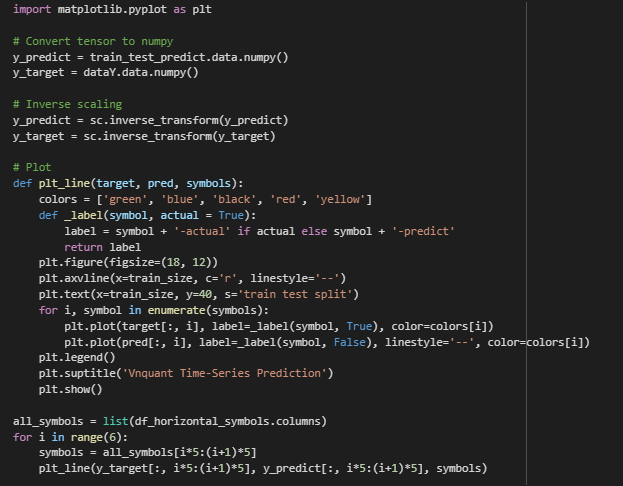


Many-to-Many

Dựa trên mô hình LSTM chúng ta có thể xem xét bài toán là một tác vụ dự báo `many-to-many`. Trong đó mỗi một giá trị đầu vào là một véc tơ window của một mã chứng khoán thứ i. Các đầu ra là giá trị dự báo của phiên tiếp theo của chính mã chứng khoán đó.



Xây dựng model



Link tham khảo:

[Khoa học dữ liệu (phamdinhkhanh.github.io)](https://phamdinhkhanh.github.io/2019/04/22/Ly_thuyet_ve_mang_LSTM.html" \l "3-m%E1%BA%A1ng-tr%C3%AD-nh%E1%BB%9B-ng%E1%BA%AFn-h%E1%BA%A1n-%C4%91%E1%BB%8Bnh-h%C6%B0%E1%BB%9Bng-d%C3%A0i-h%E1%BA%A1n-lstm---long-short-term-memory)