1. **Chủ đề**

Một công ty tài chính đang tìm cách để tối ưu hóa lợi nhuận trên thị trường ngoại hối (Forex). Họ nhận thấy rằng thị trường ngoại hối rất khó đoán, cần sự tốc độ trong giao dịch để có thể tối ưu hóa các cơ hội trên thị trường và cần sự tự động trong các giao dịch nhằm giảm thiểu rủi ro các giao dịch viên (broker) có thể tạo ra trong quá trình giao dịch. Do đó, công ty đang cần tìm 1 giải pháp sử dụng công nghệ để có thể đáp ứng được 3 tiêu chí trên.

1. **Business Understanding**

* Xác định m­­ục tiêu của dự án
* Xây dựng 1 hệ thống có khả năng sử dụng thuật toán để giao dịch tự động dựa theo mô hình dữ liệu xây dựng phù hợp với các dữ liệu về giá cả trong ngày (mở phiên, cao, thấp, đóng phiên, khối lượng).
* Ngoài ra, các dữ liệu về tài khoản của khách hàng (số tiền mặt hiện có, các hợp đồng giao dịch đang sở hữu…) cũng sẽ được sử dụng).
* Hệ thống sẽ dựa theo các mô hình thống kê (statistical model) hoặc mô hình máy học (machine learning model) từ đó đưa ra các hành động phù hợp (mua/bán, quản lý).
* Có phải bài toán đặt ra là phải đưa ra các quyết định mua bán dựa trên dự báo của mô hình với dữ liệu đầu vào là giá cả quá khứ của các hợp đồng trao đổi ngoại hối (Forex)?
* Vấn đề cần giải quyết:
* Xây dựng hoặc tìm kiếm API/Database về dữ liệu giá cả trong ngày của các hợp đồng trao đổi (ví dụ EUR/USD).
* Mô hình sẽ dự báo giá tương lai của các mã Forex (như EUR/USD) rồi sau đó sẽ đưa ra chi báo mua bán dựa vào việc mua đắt bán rẻ hoặc bán đắt và mua lại rẻ (ví dụ Buy: True/False, Sell: True/False) (trong trường hợp này coi như có khả năng bán khống/short do đó không cần sở hữu hợp đồng trước khi bán).
* **Mục đích sẽ là xây dựng 1 model có khả năng đáp ứng việc chỉ báo giao dịch bằng cách dự báo (predictive model) giá cả tương lai của mã Forex.**

1. **Analytic Approach**

* Đầu vào sẽ là bảng giá theo thời gian (time-series data) bao gồm có các cột:
* Ngày
* Giá cao nhất ngày
* Giá thấp nhất ngày
* Giá mở cửa
* Giá đóng cửa
* Giá đóng cửa đã điều chỉnh
* Khối lượng
* Đối với time series data sẽ có các đặc tính như dưới đây:
* Thời gian: Dữ liệu chuỗi thời gian được ghi lại theo thứ tự thời gian, với các quan sát được thực hiện trong các khoảng thời gian liên tiếp. Thời gian có thể được đo bằng giờ, ngày, tháng, năm hoặc các đơn vị thời gian khác.
* Tính chu kỳ: Dữ liệu chuỗi thời gian thường có tính chu kỳ, tức là có sự lặp lại các mẫu hoặc mô hình theo một chu kỳ nhất định. Tuy nhiên, tính chu kỳ sẽ có sự lặp lại không phụ thuộc vào mùa. Ví dụ như doanh số bán hàng sẽ tang vào các dịp lễ như Giáng sinh hoặc Tết.
* Tính mùa vụ: Một dạng đặc biệt của tính chu kỳ là tính mùa vụ. Dữ liệu chuỗi thời gian có tính mùa vụ thể hiện sự biến đổi theo mùa trong một chu kỳ năm. Ví dụ, doanh số bán hàng có thể tăng cao vào mùa giáng sinh và giảm vào mùa hè.
* Tính xu hướng: Dữ liệu chuỗi thời gian có thể có xu hướng tăng dần hoặc giảm dần theo thời gian. Xu hướng có thể là tăng dần (tăng trưởng), giảm dần (suy thoái), hoặc không có xu hướng rõ ràng. Ví dụ tiêu biểu có thể là xu hướng tang hoặc giảm của thị trường chứng khoán.
* Tính phi tuyến: Dữ liệu chuỗi thời gian có thể có tính phi tuyến, tức là không tuân theo một mô hình tuyến tính đơn giản. Các yếu tố phi tuyến có thể là sự biến đổi không đều, sự biến đổi không đồng nhất theo thời gian hoặc các yếu tố không xác định khác. Một ví dụ của tính phi tuyến có thể là biến đổi theo mối quan hệ hàm mũ trong phân tích đường cong suy giảm của khai thác dầu khí.
* Tính nhiễu: Dữ liệu chuỗi thời gian thường chứa các yếu tố nhiễu, tức là các biến động ngẫu nhiên không có mô hình hoặc không thể dự đoán trước. Nhiễu có thể gây ra sự biến động ngẫu nhiên trong dữ liệu và làm khó khăn trong việc dự đoán.

* Phương pháp phân tích được sử dụng: ARIMA (Autoregressive Integrated Moving Average). ARIMA được chia ra làm 3 phần: AR (Auto regression), I (Integrated), MA (Moving Average). Mô hình ARIMA giả thiết rằng dữ liệu chuỗi thời gian có mean, standard deviation sẽ không thay đổi theo thời gian (stationary) của các dữ liệu quá khứ để có thể dự báo dữ liệu tương lai, trong trường hợp này sẽ là sử dụng phương pháp tích hợp (Integrated) để loại bỏ nhiễu, và sẽ dự báo dựa trên việc tự hồi quy (Auto regression) và trung bình trượt (Moving average).
  + Thông thường thì time-series data của Forex hay chứng khoán khó có thể là non-stationary, nhưng trước khi sử dụng ARIMA model thì chúng ta có thể sử dụng Augmented Dickey-Fuller test để có thể kiểm tra xem chuỗi data chọn có phải là stationary hay không, từ đó áp dụng ARIMA model sẽ đảm bảo sự chính xác cho dự đoán hơn.
* Kết quả trả về sẽ là:
  + Giá tương lai của hợp đồng Forex.
  + Kết luận nên mua hay bán dựa theo giá cả tương lai dự báo (ví dụ như giá tương lai dự báo tang so với hiện tại thì sẽ signal mua, và ngược lại)

1. **Data Requirement**

* Đầu vào sẽ là bảng giá theo thời gian (time-series data) bao gồm có các cột:
* Ngày
* Giá cao nhất ngày
* Giá thấp nhất ngày
* Giá mở cửa
* Giá đóng cửa
* Khối lượng (có thể có hoặc không, tùy vào dữ liệu lấy của sàn giao dịch nào)
* Ví dụ như bảng DataFrame của EUR/USD như hình dưới:

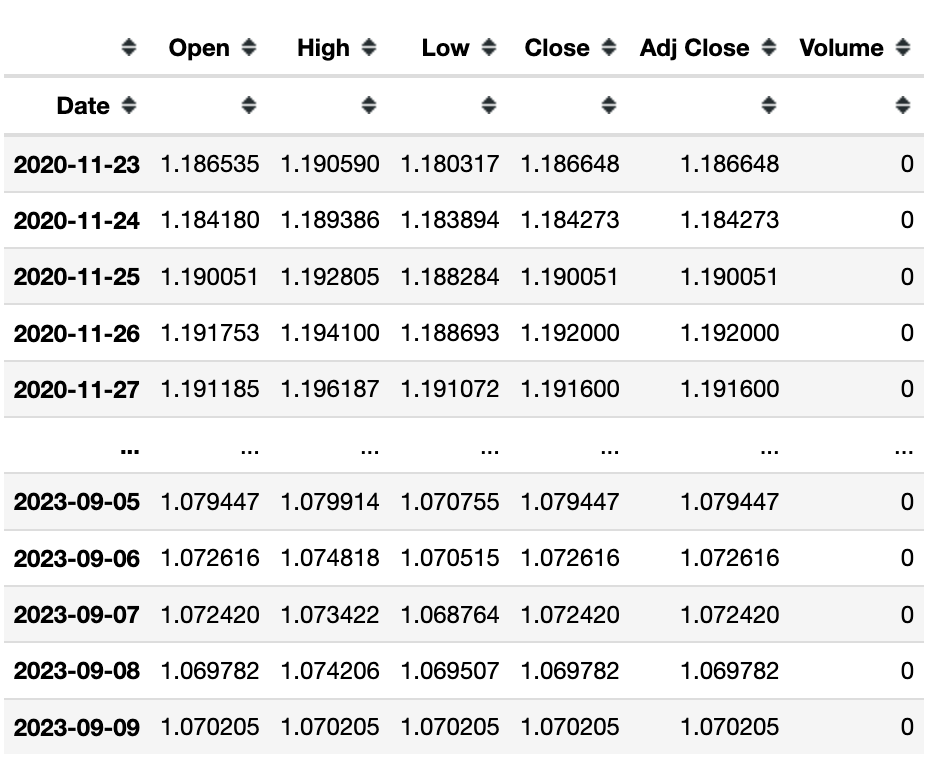


Figure 1 - Ví dụ của DataFrame EUR/USD – dữ liệu được cung cấp từ Yahoo! Finance

* Vì lý do khối lượng phụ thuộc vào các sàn giao dịch khác nhau, do đó dữ liệu volume thực tế sẽ không được sử dụng. Nếu sử dụng them dữ liệu volume thì sẽ phải sử dụng model ARIMAX (ARIMA with Exogenus Variable) để bổ sung volume là 1 factor ảnh hưởng đến giá cả.

1. **Data Collection**
   1. *Thu thập dữ liệu*

* Dữ liệu thu thập sẽ từ Yahoo! Finance với library yfinance theo định dạng Pandas Dataframe. Ví dụ như dòng code dưới:

A screen shot of a computer

Description automatically generated

Figure 2 - Python code để lấy dữ liệu dạng DataFrame từ Yahoo! Finance

* Vì lý do mô hình ARIMA sẽ hoạt động tốt nhất khi lấy các dữ liệu gần nhất để dự báo, và càng lấy dữ liệu ở xa sẽ dễ gây ra noise, nên sẽ chỉ lấy dữ liệu trong 2 năm gần nhất, theo ngày.
* Đối với trường hợp sau này sử dụng high-frequency data trên thời gian thực để tang tốc độ trading (thay vì ngày thì là giờ hoặc phút, có thể xem xét đầu tư them các API thích hợp để phục vụ cho việc lấy dữ liệu như Xgnite, Oanda, hoặc FireAnt có trả phí. Các trường hợp sử dụng API có thể sẽ phải thực hiện bổ sung them các bước Data Preparation để convert data qua Pandas DataFrame và thực hiện các bước modelling tiếp theo (vì khi sử dụng Python sẽ hiệu quả nhất khi data ở dạng DataFrame)
  1. *Thống kê dữ liệu:*
* Sau khi thu thập dữ liệu từ Yahoo! Finance về, điều đầu tiên sẽ cần phải mô tả data sơ qua theo cách thống kê dữ liệu bằng các chỉ số thống kê như dưới đây:

A screenshot of a computer

Description automatically generated

Figure - Thống kê mô tả của ví dụ EUR/USD

* Trong phần Descriptive Statistic này, chúng ta có thể detect được xem data có những giá trị bị sai số cao (extreme value) hay không. Có 1 vài trường hợp giá trị không tồn tại như NaN được thể hiện bằng các extreme value (-99999 chẳng hạn, như trong dữ liệu Địa vật lý giếng khoan của ngành Dầu khí). Trong trường hợp này, vì mean gần với các giá trị min và max (ở 1 khoảng thích hợp – ví dụ như Adj Close là mean = 1.11, min = 0.96, max = 1.23) do đó có thể kết luận là không có data nào bị extreme value.
* Có thể xem xét vẽ them histogram để kiểm tra data của chúng ta, như ví dụ dưới đây:

A screen shot of a computer

Description automatically generated

Figure - Code để vẽ Histogram của cột Giá đóng cửa (Close) bằng Seaborn

A graph with blue bars

Description automatically generated

Figure - Histogram của giá đóng cửa (Close)

* Sau đó kiểm tra xem data có bị các giá trị không tồn tại (NaN) hay không

A screenshot of a computer

Description automatically generated

Figure - Kiểm tra data có chứa giá trị không tồn tại (NaN)

* Trong trường hợp dữ liệu bị tồn tại NaN, có thể xem xét đến việc handle empty data bằng cách loại bỏ cả dòng đó hoặc thay thế bằng mean. Dòng code sau đây là 1 ví dụ (tuy nhiên không chạy do data của chúng ta không có NaN như trên ở ví dụ này)

A screen shot of a computer code

Description automatically generated

Figure - Code ví dụ về 2 cách xử lý NaN value

* 1. *Đánh giá dữ liệu*
* Dữ liệu của EUR/USD phù hợp với yêu cầu của đề bài (không tồn tại giá trị extreme values cũng như NaN).
* Vì các dữ liệu chỉ dừng lại ở 2 năm, do đó để có thể cải thiện model của chúng ta thì sẽ cần phải thu thập them các dữ liệu ở quá khứ và sử dụng ARIMA model và Train-Test split để có thể validify model tốt hơn.

1. **Bổ sung them 1 vài ý tưởng liên quan**

* Vì lý do đặc tính của ARIMA model sẽ chỉ sử dụng 1 biến time-series để phân tích và dự đoán, do đó sẽ chỉ có thể dự đoán được 1 trong 4 cột Open, Close, High, Low như trên. Việc này có thể sẽ khiến cho dự báo của chúng ta sẽ mang tính chất bị chưa đưa đầy đủ.
* Thông thường đối với các time-series data như lĩnh vực tài chính, có thể sẽ sử dụng them các phân tích kỹ thuật. (Technical indicator/analysis) để có thể tạo them các feature mang tính chất analytical như Bollinger Band, Relative Strength Index để có thể bổ sung them insight cho feature.
* Cuối cùng là sẽ sử dụng các model mang tính chất tích hợp nhiều loại dữ liệu time-series vào để xây dựng predictive model nhằm thực hiện việc chỉ báo mua bán hiệu quả hơn, ví dụ như Long-Short Term Memory Model (LSTM) hoặc Random Forest.