Tổng quan về Machine Learning

Trần Lê Minh Đức

Nội dung

Machine Learning

Là một lĩnh vực của trí tuệ nhân tạo liên qua đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống học tự động từ dữ liệu để giải quyết các vấn đề cụ thể. Ví dụ các máy có thể học cách phân loại thư điện tử có phải thư rác hay không và tự động sắp xếp vào các thư mục tương ứng.

Machine Learning có liên quan đến thống kê vì cả hai lĩnh vực đều nghiêncứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán

Machine Learning có hiện nay được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, máy phân tích thị trường chứng khoán, nhận dạng tiếng nói và chữ viết...

Các khái niệm cơ bản

1. Data Variables.
2. Data Processing.
3. Feauture Engineering

1. *Data Variables*

- Là một đại lượng hoặc đặc trưng mà bạn thu thập từ dữ liệu để sử dụng trong các phân tích, mô hình học máy (machine learning), hoặc thống kê. Trong ngữ cảnh học máy, các biến dữ liệu thường được chia thành hai loại chính: biến độc lập (independent variables) và biến phụ thuộc (dependent variables).

- Các loại Data Variables:

+ Biến Độc Lập (Independent Variables):

* Khái niệm: Là các biến được sử dụng để dự đoán hoặc giải thích sự thay đổi của biến phụ thuộc. Biến độc lập không bị ảnh hưởng bởi các biến khác trong mô hình.

* Ví dụ: Trong mô hình dự đoán giá nhà, biến độc lập có thể là diện tích nhà, số phòng ngủ,v.v.

+ Biến Phụ Thuộc (Dependent Variables):

* Khái niệm: Là biến mà bạn muốn dự đoán hoặc giải thích. Biến này phụ thuộc vào các giá trị của các biến độc lập.

* Ví dụ:Trong ví dụ trên, giá nhà sẽ là biến phụ thuộc vì nó phụ thuộc vào diện tích, số phòng ngủ, và các yếu tố khác.

2. Data Processing

- Là một quá trình quan trọng trong phân tích dữ liệu và học máy, giúp biến dữ liệu thô thành thông tin có giá trị để mô hình hóa và phân tích. Quá trình này có thể bao gồm nhiều bước từ việc thu thập, làm sạch, chuyển đổi, cho đến chuẩn hóa dữ liệu. Mục đích là tạo ra dữ liệu có chất lượng tốt và dễ sử dụng trong các mô hình học máy.

- Các bước chính bao gồm:

- Thu thập dữ liệu: Thu thập dữ liệu từ các nguồn khác nhau.
- Làm sạch dữ liệu: Xử lý dữ liệu thiếu, sai hoặc không hợp lệ.
- Chuyển đổi dữ liệu: Chuẩn hóa, mã hóa hoặc chuẩn hóa dữ liệu.
- Tách dữ liệu: Chia dữ liệu thành tập huấn luyện và kiểm tra.
- Rút trích và chọn đặc trưng: Tạo ra hoặc lựa chọn đặc trưng quan trọng.
- Giảm chiều dữ liệu: Dùng các kỹ thuật như PCA để giảm số lượng đặc trưng không cần thiết.

3. Feauture Engineer

- Là một trong những bước quan trọng trong quy trình xây dựng mô hình học máy. Mục tiêu của feature engineering là tạo ra hoặc chuyển đổi các đặc trưng (features) từ dữ liệu thô thành những đặc trưng có khả năng giúp mô hình học máy hiểu rõ hơn về vấn đề và cải thiện độ chính xác.

- Các bước chính :

Chuyển đổi dữ liệu: Áp dụng các phép toán hoặc hàm để tạo ra các đặc trưng mới (ví dụ: log transformation, bình phương).
Tạo đặc trưng mới: Kết hợp các đặc trưng hiện có để tạo ra thông tin có giá trị (ví dụ: kết hợp tuổi và thu nhập để tạo đặc trưng mới).
Xử lý dữ liệu thiếu: Điền giá trị thiếu bằng cách sử dụng các chiến lược như trung bình, giá trị gần nhất, hoặc mô hình dự đoán.
Chọn đặc trưng: Lựa chọn các đặc trưng quan trọng nhất, loại bỏ các đặc trưng không hữu ích hoặc dư thừa.

- Mục tiêu của feature engineering là giúp mô hình học máy nhận diện được các mẫu trong dữ liệu một cách chính xác hơn, từ đó cải thiện hiệu suất dự đoán.

Biến đổi và tạo đặc trưng (Feature Engineering)

Feature Engineering là quá trình:

- Chọn lọc, biến đổi hoặc
- Tạo ra đặc trưng (feature) mới từ dữ liệu thô → Nhằm giúp mô hình hiểu dữ liệu tốt hơn và dự đoán chính xác hơn.

Các kỹ thuật phổ biến :

- Tạo đặc trưng mới (Feature Creation)
- Biến đổi đặc trưng (Feature Transformation)
- Scaling và chuẩn hóa đặc trưng
- Giảm số chiều (Dimensionality Reduction)
- Encoding đặc trưng phân loại (Categorical Encoding)
- Tự động hóa Feature Engineering

Tiền xử lí dữ liệu

- Là một bước quan trọng trong quy trình xây dựng mô hình học máy. Đây là quá trình làm sạch, chuẩn bị và chuyển đổi dữ liệu thô thành một định dạng có thể sử dụng để huấn luyện mô hình học máy. Dữ liệu thực tế thường không hoàn hảo và có thể chứa rất nhiều vấn đề như dữ liệu thiếu, dữ liệu không chính xác, hoặc dữ liệu không đồng nhất. Tiền xử lý dữ liệu giúp làm sạch và chuẩn hóa dữ liệu để mô hình có thể học được các mẫu hiệu quả hơn.

- Các bước chính :

- Xử lý dữ liệu thiếu (Missing Data)
- Chuyển đổi dữ liệu (Data Transformation)
- Loại bỏ dữ liệu nhiễu (Noise)
- Biến đổi và tạo đặc trưng (Feature Engineering)
- Mã hóa dữ liệu phân loại (Categorical Data Encoding)
- Xử lý dữ liệu không đồng nhất (Handling Inconsistent Data)

Xử lý dữ liệu thiếu (Missing Data):

- Dữ liệu thiếu là vấn đề phổ biến trong nhiều tập dữ liệu. Việc không xử lý đúng cách có thể làm giảm hiệu quả của mô hình. Các phương pháp xử lý dữ liệu thiếu bao gồm:

- Điền giá trị trung bình, trung vị hoặc mode: Đối với các đặc trưng số học, bạn có thể điền giá trị thiếu bằng trung bình (mean) hoặc trung vị (median). Đối với các đặc trưng phân loại, bạn có thể điền giá trị thiếu bằng mode (giá trị xuất hiện nhiều nhất).
- Loại bỏ các hàng hoặc cột có dữ liệu thiếu: Nếu số lượng dữ liệu thiếu quá nhiều, một cách khác là loại bỏ các hàng (row) hoặc cột (column) có dữ liệu thiếu.
- Sử dụng mô hình để dự đoán giá trị thiếu: Một phương pháp tiên tiến hơn là sử dụng các mô hình học máy để dự đoán các giá trị thiếu dựa trên các đặc trưng khác có sẵn trong dữ liệu.

Chuẩn hóa dữ liệu (Normalization/Standardization):

- Dữ liệu thường có các đơn vị hoặc phạm vi khác nhau. Việc chuẩn hóa giúp đưa dữ liệu về một phạm vi thống nhất:

- Normalization: Đưa tất cả các giá trị về một phạm vi từ 0 đến 1. Thường áp dụng cho các thuật toán yêu cầu tính toán khoảng cách, như k-NN (K-Nearest Neighbors).
- Standardization: Điều chỉnh dữ liệu sao cho có phân phối chuẩn với trung bình bằng 0 và độ lệch chuẩn bằng 1. Thường được sử dụng khi dữ liệu có phân phối không chuẩn, chẳng hạn trong các mô hình hồi quy tuyến tính hoặc SVM.

Loại bỏ dữ liệu nhiễu (Noise) :

Dữ liệu nhiễu là các điểm không tuân theo quy luật chung của dữ liệu hoặc có thể là do lỗi trong quá trình thu thập, nhập liệu hoặc truyền tải dữ liệu.

Các cách loại bỏ :

- Phát hiện bằng thống kê : Z-score, IQR (Interquartile Range),...
- Dùng mô hình học máy để phát hiện dị biệt : Isolation Fores (Phát hiện điểm dị biệt trong dữ liệu lớn), DBSCAN(Thuật toán phân cụm có thể phát hiện điểm nhiễu (không thuộc cụm nào)), Local Outlier Factor (LOF)(Tính điểm bất thường dựa trên mật độ lân cận),....
- Làm mịn dữ liệu (Data Smoothing)

Mã hóa dữ liệu phân loại (Categorical Data Encoding)

là một phần bắt buộc trong tiền xử lý dữ liệu nếu bạn làm với machine learning hoặc bất kỳ pipeline xử lý nào mà yêu cầu dữ liệu đầu vào là số. Mô hình không hiểu văn bản, mà chỉ hiểu số, nên ta cần chuyển đổi các giá trị phân loại thành dạng số một cách hợp lý.

Tại sao cần mã hóa dữ liệu phân loại?

- Mô hình không thể xử lý dữ liệu dạng string (như "loại máy = A").
- Mỗi phương pháp mã hóa sẽ phù hợp với loại mô hình và dữ liệu cụ thể.
- Việc chọn sai cách mã hóa có thể làm méo mó mối quan hệ, gây overfitting hoặc underfitting.

Các kĩ thuật phổ biến như:

- One-Hot Encoding
- Label Encoding
- Frequency / Count Encoding
- Ordinal Encoding
- Target Encoding (Mean Encoding)

Xử lý dữ liệu không đồng nhất (Handling Inconsistent Data)

Dữ liệu không đồng nhất là gì?

Là khi cùng một thông tin nhưng được ghi nhận theo nhiều cách khác nhau, làm cho phân tích hoặc mô hình học máy hiểu sai hoặc lỗi.

Kỹ thuật xử lý dữ liệu không đồng nhất

- Chuẩn hóa định dạng
- Sửa lỗi chính tả hoặc mapping
- Chuyển đổi đơn vị
- Chuẩn hóa kiểu dữ liệu
- Xử lý dữ liệu trùng hoặc dư thừa

Hậu quả nếu không xử lý:

- Model học máy sai → dự đoán tệ.
- Dashboard hiển thị lệch → quyết định kinh doanh sai.
- CI/CD pipeline fail do dữ liệu lỗi → ảnh hưởng vận hành.

Model Training – Các bước huấn luyện mô hình

1. Thu Thập Dữ Liệu
   - Mục tiêu: Tìm kiếm và chuẩn bị dữ liệu phù hợp với bài toán.
   - Hành động chi tiết:
     - Định nghĩa rõ ràng vấn đề và thông tin cần thiết.
     - Sử dụng các công cụ như API, khảo sát, hoặc web scraping để thu thập dữ liệu.
     - Kiểm tra tính hợp lệ và độ đầy đủ của dữ liệu ban đầu.
   - Lưu ý: Cần đảm bảo dữ liệu không vi phạm pháp luật (ví dụ: GDPR).
2. Tiền Xử Lý Dữ Liệu
   - Mục tiêu: Đảm bảo dữ liệu sẵn sàng cho mô hình.
   - Hành động chi tiết:
     - Làm sạch dữ liệu: Xóa trùng lặp, sai lệch, hoặc giá trị ngoại lệ.
     - Xử lý dữ liệu thiếu: Thay thế bằng giá trị trung bình hoặc sử dụng các kỹ thuật suy diễn.
     - Mã hóa dữ liệu phân loại: Sử dụng One-Hot Encoding hoặc Label Encoding.
     - Tái định dạng: Biến đổi dữ liệu thành dạng phù hợp, ví dụ: chuẩn hóa hoặc tiêu chuẩn hóa.
   - Lưu ý: Chú ý tính đa dạng và cân bằng giữa các lớp dữ liệu.

3. Chia Tách Dữ Liệu

- Mục tiêu:Chia dữ liệu cho các giai đoạn khác nhau.
- Hành động chi tiết:
  - Chia tập huấn luyện (70-80%), tập kiểm tra (20-30%), và tập xác nhận (nếu cần).
  - Giữ nguyên phân phối của dữ liệu giữa các tập (stratified split).
  - Sử dụng thư viện như `train_test_split` trong scikit-learn.

4. Lựa Chọn Mô Hình

- Mục tiêu: Chọn mô hình phù hợp với đặc thù bài toán.
- Hành động chi tiết:
  - Đánh giá các mô hình tiềm năng dựa trên yêu cầu (ví dụ: độ chính xác, tốc độ).
  - Tham khảo các mô hình phổ biến như hồi quy logistic, SVM, hoặc mạng nơ-ron sâu.
  - Xác định các siêu tham số cần tối ưu.

5. Huấn Luyện Mô Hình

- Mục tiêu: Đào tạo mô hình để học từ tập huấn luyện.
- Hành động chi tiết:
  - Khởi tạo mô hình với các tham số ban đầu.
  - Sử dụng thuật toán tối ưu hóa, ví dụ: Gradient Descent.
  - Theo dõi loss function qua từng epoch để đảm bảo quá trình hội tụ.

6. Đánh Giá Mô Hình
- Mục tiêu: Kiểm tra hiệu suất của mô hình trên tập kiểm tra.
- Hành động chi tiết:
    - Chọn các chỉ số phù hợp với bài toán:
        - Hồi quy: RMSE, MAE.
        - Phân loại: Precision, Recall, F1-score.
    - Vẽ biểu đồ như Confusion Matrix để hiểu rõ hơn về lỗi.
- Lưu ý: Tránh đánh giá mô hình chỉ dựa vào một chỉ số duy nhất.

7. Tối Ưu Hóa Mô Hình
- Mục tiêu: Cải thiện hiệu suất và độ tin cậy.
- Hành động chi tiết:
    - Áp dụng kỹ thuật Regularization (L1, L2) để tránh overfitting.
    - Tăng cường dữ liệu (Data Augmentation) nếu dữ liệu ít.
    - Dùng Cross-Validation để đánh giá mô hình với các siêu tham số khác nhau.

8. Triển Khai Mô Hình
- Mục tiêu: Đưa mô hình vào môi trường thực tế.
- Hành động chi tiết:
  - Đóng gói mô hình dưới dạng API hoặc container (ví dụ: Docker).
  - Kết nối với cơ sở dữ liệu để thu thập đầu vào thực tế.
  - Thử nghiệm trên một lượng nhỏ dữ liệu thực tế trước khi triển khai toàn diện.

9. Theo Dõi Hiệu Năng
- Mục tiêu: Duy trì và cải thiện hiệu suất lâu dài.
- Hành động chi tiết:
  - Giám sát độ chính xác qua thời gian.
  - Điều chỉnh mô hình khi dữ liệu hoặc điều kiện thay đổi.
  - Lập lịch huấn luyện lại định kỳ nếu cần.

Model Evaluation: Các chỉ số đánh giá mô hình (Accuracy, Precision, Recall, F1 Score).

1. Accuracy (Độ Chính Xác)

- Mô tả: Tỷ lệ dự đoán đúng trên tổng số dữ liệu.
- Cách tính:

$$Accuracy = \frac{Số\ dự\ đoán\ đúng}{Tổng\ số\ mẫu}$$

- Ưu điểm:
  - Đơn giản, dễ hiểu.
  - Hữu ích khi dữ liệu cân bằng (số lượng mẫu các lớp không chênh lệch).
- Nhược điểm:
  - Không hiệu quả trong trường hợp dữ liệu không cân bằng.

2. Precision (Độ Chính Xác của Lớp Dương)

- Mô tả: Tỷ lệ mẫu dương được dự đoán đúng trên tổng số mẫu được dự đoán là dương.
- Cách tính:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- Ưu điểm:
  - Quan trọng trong bài toán giảm thiểu cảnh báo sai (False Positive).
- Nhược điểm:
  - Có thể bỏ qua các mẫu dương bị dự đoán sai (False Negative).

3. Recall (Độ Bao Phủ)

- Mô tả: Tỷ lệ mẫu dương được dự đoán đúng trên tổng số mẫu thực tế là dương.
- Cách tính:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- Ưu điểm:
  - Hiệu quả trong bài toán giảm thiểu bỏ sót (False Negative).
- Nhược điểm:
  - Có thể dẫn đến nhiều cảnh báo sai (False Positive).

4. F1 Score (Điểm F1)

- Mô tả: Trung bình hài hòa của Precision và Recall.
- Cách tính:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- Ưu điểm:
  - Cân bằng giữa Precision và Recall.
- Nhược điểm:
  - Có thể không rõ ràng khi cần ưu tiên một chỉ số cụ thể hơn.