**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN II**

-----

|  |
| --- |
|  |
|  |  |

**BÁO CÁO CUỐI KÌ**

**MÔN: IoT và Ứng dụng**

**Đề tài :**

**TÌM HIỂU VÀ TRIỂN KHAI MÔ HÌNH PIPELINE VỚI THUẬT TOÁN DECISION TREE**

|  |  |
| --- | --- |
| **Giảng viên :** | **ThS. Đàm Minh Lịnh** |
| **Nhóm:**  **Sinh viên thực hiện:** | **3** |
| * **Đặng Khắc Toản** | **N20DCCN067** |
| * **Nguyễn Phúc Luân** | **N20DCCN036** |
|  |  |

**TPHCM, tháng 1 năm 2024**

Mục lục

[I. Tổng quan 4](#_Toc155336430)

[1. Lý do chọn đề tài 4](#_Toc155336431)

[2. Mục tiêu 4](#_Toc155336432)

[II. Cơ sở lý thuyết 4](#_Toc155336433)

[1. Pipeline trong machine learning. 4](#_Toc155336434)

[2. Data validation 6](#_Toc155336435)

[3. Data preprocessing 7](#_Toc155336436)

[4. Thuật toán Decision Tree. 9](#_Toc155336437)

[5. Evaluation 12](#_Toc155336438)

[III. Triển khai 13](#_Toc155336439)

[1. Công nghệ áp dụng 13](#_Toc155336440)

[2. Thu thập dữ liệu (dataset) 13](#_Toc155336441)

[3. Giải thích code 13](#_Toc155336442)

[4. Kết quả và đánh giá 18](#_Toc155336443)

[IV. Kết luận 19](#_Toc155336444)

[Tài liệu tham khảo 20](#_Toc155336445)

# Tổng quan

## Lý do chọn đề tài

Hệ thống học máy thường bao gồm nhiều thành phần nhỏ, chẳng hạn như xử lý dữ liệu, đào tạo, đánh giá mô hình, dự đoán dữ liệu mới,… Nếu không xây dựng được một pipeline hoàn chỉnh với từng thành phần được phân tách rõ ràng, rất nhiều vấn đề có thể phát sinh. Việc tách dần hệ thống và kết hợp lại thành pipeline dẫn đến việc dễ dàng phát hiện lỗi khi thực hiện mô hình training. Đồ án tập trung vào nghiên cứu và triển khai một pipeline cho việc dự đoán những loại người nào có khả năng sống sót và những hành khách nào sống sót trong thảm kịch đắm tàu Titanic.

## Mục tiêu

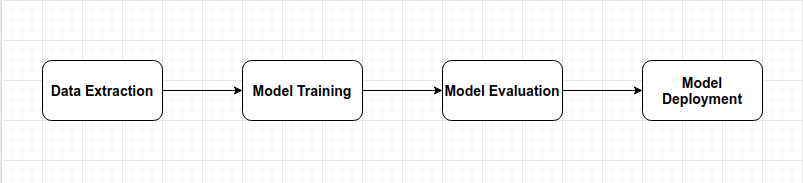
* Tìm hiểu về pipeline trong machine learning.
* Tìm hiểu các khái niệm: data validation, data preprocessing, feature engineering.
* Tìm hiểu về đánh giá mô hình: accuracy, precision, recall.
* Thiết kế pipeline và triển khai mô hình học máy.

# Cơ sở lý thuyết

## Pipeline trong machine learning.

* 1. **Khái niệm**

Pipeline là một cách để mã hóa và tự động hóa các công đoạn và quy trình làm việc cần thiết để tạo ra một mô hình học máy. Pipeline bao gồm nhiều bước tuần tự thực hiện mọi thứ từ trích xuất dữ liệu (data extraction) và tiền xử lý dữ liệu (preprocessing data) cho đến huấn luyện và triển khai mô hình.

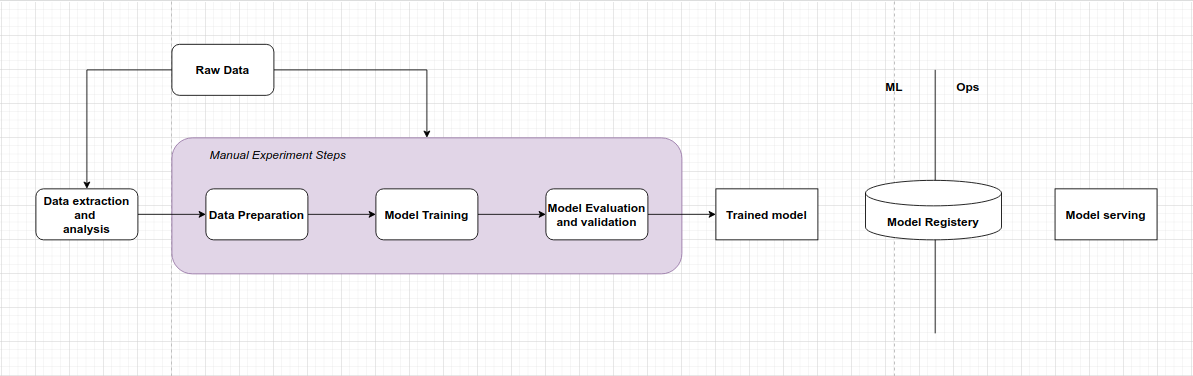


Hình 1. Các bước trong quá trình xây dựng và triển khai mô hình

Đối với các hệ thống sử dụng mô hình ML, thì quy trình pipeline là phần trung tâm của sản phẩm. Nó đóng gói toàn bộ các phương pháp xử lý dữ liệu tốt nhất để tạo ra một mô hình học máy phù hợp nhất cho một bộ dữ liệu cụ thể. Ngoài ra pipeline còn cho phép mô hình thực thi trên quy mô lớn. Một thiết kế pipeline end-to-end sẽ cho phép hệ thống của bạn cập nhật một cách thường xuyên các mô hình học máy một cách nhanh chóng.

* 1. **Quy trình machine learning thủ công**

Ở thời điểm ban đầu của các mô hình ML hướng tới giải quyết một bài toán cụ thể mà doanh nghiệp đối diện, chẳng hạn bài toán gợi ý câu search hay bài toán phát hiện gian lận khách hàng. Team data science thông thường bắt đầu với một qui trình thủ công mà các bước trong qui trình xây dựng mô hình như: thu thập dữ liệu, làm sạch dữ liệu, huấn luyện mô hình và đánh giá mô hình sẽ được viết ngắn gọn trong một notebook. Notebook này được vận hành cục bộ để tạo ra phiên bản mô hình thô sơ chưa thế triển khai ngay. Sau đó chúng được chuyển giao sang cho Team engineer để chuyển hoá thành API và áp dụng vào sản phẩm.



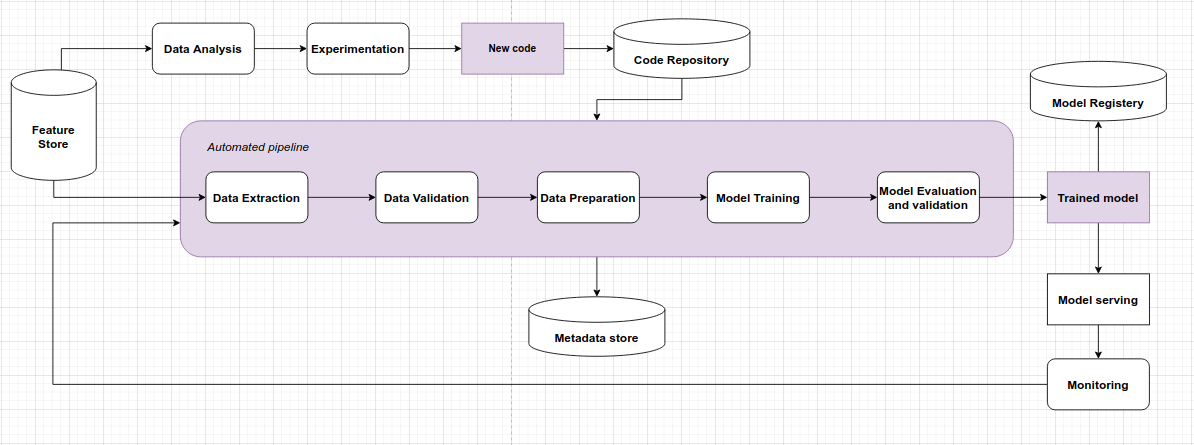
Hình 2: Workflow của mô hình machine learning thủ công

Phần màu xám Manual Experement Steps là những bước trong qui trình thủ công được thực hiện bởi Team data science. Đầu ra của bước này là trained model, để triển khai được trained model lên môi trường production dưới dạng API (chính là các Model serving trên sơ đồ) thì trained model cần được giao cho team Engineer.

Workflow trong qui trình thủ công thường mang tính đột xuất và bắt đầu bị phá vỡ khi Team data science tăng tốc chu kỳ mô hình. Các quy trình thủ công rất khó lặp lại vì các xử lý không được đóng gói. Do đó những khối mã lệnh được viết trên block code của notebook sẽ không còn phù hợp để tăng tốc chu kỳ của mô hình.

* 1. **Quy trình machine learning tự động**

Khi Team data science chuyển sang giai đoạn phát triển với qui mô lớn và cần cập nhật nhiều mô hình thường xuyên trên môi trường production, thì phương pháp tiếp cận theo pipeline đóng vai trò cực kì quan trọng. Trong workflow theo cách tiếp cận tự động hoá pipeline, bạn không phát triển và duy trì mỗi một mô hình là một sản phẩm mà thay vào đó, bạn phát triển một pipeline và pipeline chính là sản phẩm.



Hình 3: Workflow của mô hình machine learning trong pipeline tự động.

Phần màu xám Automated Pipeline thể hiện các thành phần trong qui trình pipeline tự động. Một qui trình pipeline tự động bao gồm các thành phần được sắp đặt theo một bản thiết kế, trong đó qui định cách chúng được kết hợp với nhau để xây dựng và cập nhật toàn bộ mô hình.

Một hệ thống áp dụng pipeline tự động cung cấp khả năng thực thi, lặp lại dễ dàng và nhanh chóng. Nó cũng cho phép bạn xác định các đầu vào và đầu ra cần thiết được sử dụng trong mô hình.

* 1. **Ưu điểm khi sử dụng pipeline**
* Đóng gói theo quy trình: Quy trình xây dựng mô hình của một hệ thống được đóng gói lại trong một pipeline và có khả năng tái sử dụng khi cần thiết.
* Khả năng tự động hoá: Mọi bước trong quy trình của mô hình được tự động hoá mà không cần phải can thiệp vào code.
* Triển khai nhanh chu kỳ vòng lặp: Một chu kì từ thu thập dữ liệu tới triển khai mô hình có thể được triển khai ngay khi cập nhật phiên bản code mới của pipeline.
* Tự động hoá quá trình kiểm thử và đo lường hiệu suất mô hình: Mô hình sau khi được huấn luyện sẽ được triển khai ngay và kết quả đánh giá hiệu suất mô hình được thể hiện ở Monitoring. Team data science có thể dễ dàng thực hiện đối chiếu, so sánh, đánh giá hiệu năng mô hình.
* Kiểm soát version của pipeline: Những phiên bản mới của pipeline sẽ được tagging và caching trong một Code Repository.

## Data validation

Data Validation là bước kiểm định dữ liệu. Bước này ít được nhắc tới trong các tài liệu nhưng lại rất quan trọng. Tại bước này, dữ liệu cần được kiểm tra xem có tương tự như dữ liệu hiện có trong cơ sở dữ liệu hay không. Lưu ý rằng trong các bài toán thực tế, dữ liệu mới thường xuyên được sinh ra, và mô hình cần thường xuyên được cập nhật dựa trên dữ liệu mới này để “bắt kịp xu thế”. Việc kiểm tra dữ liệu có tương tự không có thể được xác định dựa trên các thông số thống kê. Chẳng hạn với bài toán dự đoán một người dùng có click vào một quảng cáo hay không, nếu tỉ lệ click/không click hiện tại chỉ là 1% mà trong dữ liệu mới, tỉ lệ này là 50% thì nhiều khả năng dữ liệu mới chứa nhiều thông tin spam. Nếu để mô hình huấn luyện trên dữ liệu này, rất nhiều khả năng mô hình của bạn sẽ có kết quả tệ.

## Data preprocessing

Data preprocessing trong machine learning là quá trình chuẩn bị và xử lý dữ liệu trước khi đưa vào mô hình máy học. Nó bao gồm một loạt các bước để làm sạch, chuyển đổi và chuẩn hóa dữ liệu để làm cho nó phù hợp và hữu ích hơn cho quá trình huấn luyện của mô hình. Các bước thông thường trong data preprocessing bao gồm:

* Xử lý dữ liệu thiếu: Điền vào các giá trị thiếu trong dữ liệu, có thể thông qua các phương pháp như điền giá trị trung bình, median hoặc sử dụng các kỹ thuật máy học để dự đoán giá trị.
* Chuẩn hóa dữ liệu: Đưa dữ liệu về cùng một khoảng giá trị hoặc phân phối, giúp cho các thuật toán máy học hoạt động hiệu quả hơn. Ví dụ, chia tỷ lệ các biến số về cùng một đơn vị hoặc sử dụng phép chuẩn hóa z-score.
* Encoding dữ liệu: Chuyển đổi dữ liệu từ dạng không số sang số để mô hình có thể hiểu được. Ví dụ, chuyển đổi biến categorical thành dạng one-hot encoding hoặc sử dụng kỹ thuật Label Encoding.
* Feature scaling: Đảm bảo rằng tất cả các biến có cùng đơn vị đo lường, tránh sự thiên lệch của dữ liệu gây ảnh hưởng đến hiệu suất của mô hình.
* Lựa chọn đặc trưng: Loại bỏ các đặc trưng không cần thiết hoặc tạo ra các biến mới từ các đặc trưng hiện có để cải thiện hiệu suất của mô hình.
* Phân chia dữ liệu: Chia dữ liệu thành các tập huấn luyện, tập validation và tập kiểm tra để đánh giá hiệu suất của mô hình.

Quá trình data preprocessing đóng vai trò quan trọng trong machine learning bởi vì dữ liệu đầu vào tốt sẽ giúp mô hình học được các mẫu và thông tin quan trọng một cách hiệu quả, từ đó cải thiện khả năng dự đoán và hiệu suất của mô hình.

1. **Feature Engineering**
   1. **Khái niệm**

Hiện nay các phương pháp học máy xuất hiện ngày càng nhiều và trở nên mạnh mẽ hơn. Các mô hình học máy như mạng neural network, Random Forest, Decision Tree, SVM, kNN,… đều là những mô hình có tính tường minh thấp, độ chính xác cao, độ phức tạp và tính linh hoạt cao. Các mô hình học máy đa dạng sẽ làm phong phú thêm sự lựa chọn của các modeler. Tuy nhiên bên cạnh việc áp dụng các phương pháp mạnh, modeler cần phải chuẩn hóa dữ liệu tốt, bởi dữ liệu là nguyên liệu để mô hình dựa trên đó xây dựng một phương pháp học. Nếu mô hình học trên một bộ dữ liệu không tốt, kết quả dự báo sẽ không tốt. Nếu mô hình học trên một bộ dữ liệu tốt, kết quả mô hình tự khắc sẽ được cải thiện. Chính vì thế vai trò của chuẩn hóa dữ liệu quan trọng đến mức Andrew Nguyen đã từng nói ‘xây dựng mô hình machine learning không gì khác là thực hiện feature engineering’. Và thực tế cũng cho thấy trong các cuộc thi phân tích dữ liệu, các leaderboard đều áp dụng tốt các kĩ thuật tạo đặc trưng bên cạnh việc áp dụng những phương pháp mạnh. Những mô hình đơn giản nhưng được xây dựng trên biến chất lượng thường mang lại hiệu quả hơn những mô hình phức tạp như mạng nơ ron hoặc các mô hình kết hợp nhưng được xây dựng trên biến chưa được sử dụng các kĩ thuật tạo đặc trưng.

* 1. **Các phương pháp**

Về kĩ thuật tạo đặc trưng chúng ta có 3 phương pháp chính:

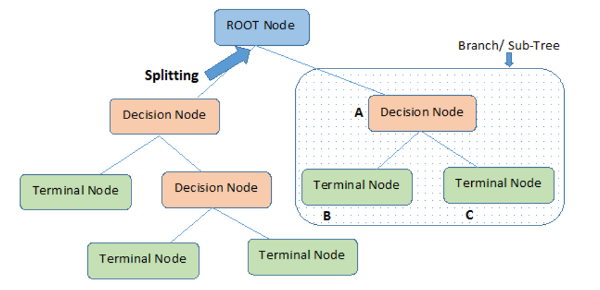
* Trích lọc feature: Không phải toàn bộ thông tin được cung cấp từ một biến dự báo hoàn toàn mang lại giá trị trong việc phân loại. Do đó chúng ta cần phải trích lọc những thông tin chính từ biến đó. Chẳng hạn như trong các mô hình chuỗi thời gian chúng ta thường sử dụng kĩ thuật phân rã thời gian để trích lọc ra các đặc trưng như Ngày thành Năm, Tháng, Quí,…. Các đặc trưng mới sẽ giúp phát hiện các đặc tính chu kì và mùa vụ, những đặc tính mà thường xuất hiện trong các chuỗi thời gian. Kĩ thuật trích lọc đặc trưng thông thường được áp dụng trên một số dạng biến như:
  + Trích lọc đặc trưng trong xử lý ảnh và xử lý ngôn ngữ tự nhiên: Các mạng nơ ron sẽ trích lọc ra những đặc trưng chính và học từ những đặc trưng này để thực hiện tác vụ phân loại.
  + Dữ liệu về vị trí địa lý: Từ vị trí địa lý có thể suy ra vùng miền, thành thị, nông thôn, mức thu nhập trung bình, các yếu tố về nhân khẩu,….
  + Dữ liệu thời gian: Phân rã thời gian thành các thành phần thời gian
* Biến đổi feature: Biến đổi dữ liệu gốc thành những dữ liệu phù hợp với mô hình nghiên cứu. Những biến này thường có tương quan cao hơn đối với biến mục tiêu và do đó giúp cải thiện độ chính xác của mô hình. Các phương pháp này bao gồm:
  + Chuẩn hóa và thay đổi phân phối của dữ liệu thông qua các kĩ thuật feature scaling như Minmax scaling, Mean normalization, Unit length scaling, Standardization.
  + Tạo biến tương tác: Thông qua việc thêm vào mô hình các biến bậc cao và biến tương tác để tạo ra một mô hình mới và kiểm tra hệ số các biến mới có ý nghĩa thống kê hay không
  + Xử lý dữ liệu missing: Có nhiều lý do khiến ta phải xử lý missing data. Một trong những lý do đó là dữ liệu missing cũng mang những thông tin giá trị, do đó nếu thay thế được các missing bằng những giá trị gần đúng sẽ mang lại nhiều thông tin hơn cho mô hình. Bên cạnh đó nhiều mô hình không làm việc được với dữ liệu missing dẫn tới lỗi training. Do đó ta cần giải quyết các biến missing. Đối với biến numeric, các phương pháp đơn giản nhất là thay thế bằng mean, median,…. Một số kĩ thuật cao cấp hơn sử dụng phân phối ngẫu nhiên để fill các giá trị missing dựa trên phân phối của các giá trị đã biết hoặc sử dụng phương pháp simulate missing value dựa trên trung bình của các quan sát láng giềng. Đối với dữ liệu category, missing value có thể được giữ nguyên như một class độc lập hoặc gom vào các nhóm khác có đặc tính phân phối trên biến mục tiêu gần giống.
* Lựa chọn feature: Phương pháp này được áp dụng trong những trường hợp có rất nhiều dữ liệu mà chúng ta cần lựa chọn ra dữ liệu có ảnh hưởng lớn nhất đến sức mạnh phân loại của mô hình. Các phương pháp có thể áp dụng đó là ranking các biến theo mức độ quan trọng bằng các mô hình như Random Forest, Linear Regression, Neural Network, SVD,…; Sử dụng chỉ số IV trong scorecard; Sử dụng các chỉ số khác như AIC hoặc Pearson Correlation, phương sai. Chúng ta có thể phân chia các phương pháp trên thành 3 nhóm:
  + Cách tiếp cận theo phương pháp thống kê: Sử dụng tương quan Pearson Correlation, AIC, phương sai, IV.
  + Lựa chọn đặc trưng bằng sử dụng mô hình: Random Forest, Linear Regression, Neural Network, SVD.
  + Lựa chọn thông qua lưới (grid search): Coi số lượng biến như một thông số của mô hình. Thử nghiệm các kịch bản với những số lượng biến khác nhau.

## Thuật toán Decision Tree.

* 1. **Tổng quan về cây quyết định**

Cây quyết định là một trong những công cụ mạnh mẽ nhất của thuật toán học có giám sát được sử dụng cho cả nhiệm vụ phân loại và hồi quy. Nó xây dựng một cấu trúc cây giống như sơ đồ trong đó mỗi nút bên trong biểu thị một thử nghiệm trên một thuộc tính, mỗi nhánh biểu thị một kết quả của thử nghiệm và mỗi nút lá (nút đầu cuối) giữ một nhãn lớp. Nó được xây dựng bằng cách chia đệ quy dữ liệu huấn luyện thành các tập con dựa trên giá trị của các thuộc tính cho đến khi đáp ứng tiêu chí dừng, chẳng hạn như độ sâu tối đa của cây hoặc số lượng mẫu tối thiểu cần thiết để phân tách một nút. Trong quá trình đào tạo, thuật toán Cây quyết định chọn thuộc tính tốt nhất để phân chia dữ liệu dựa trên một số liệu như tạp chất entropy hoặc Gini, đo mức độ tạp chất hoặc tính ngẫu nhiên trong các tập hợp con. Mục tiêu là tìm ra thuộc tính tối đa hóa việc thu được thông tin hoặc giảm tạp chất sau khi phân chia.

* 1. **Các thuật ngữ trong cây quyết định**
* Root Node (Nút gốc): Nút ban đầu ở đầu cây quyết định, nơi toàn bộ tập hợp hoặc tập dữ liệu bắt đầu phân chia dựa trên các tính năng hoặc điều kiện khác nhau.
* Decision Nodes (Nút quyết định): Các nút tạo ra từ việc phân tách các nút gốc được gọi là nút quyết định. Các nút này đại diện cho các quyết định hoặc điều kiện trung gian trong cây.
* Leaf Nodes (Nút lá): Các nút không thể phân tách thêm, thường biểu thị phân loại hoặc kết quả cuối cùng. Các nút lá còn được gọi là nút cuối.
* Sub-Tree (Cây con): Tương tự như phần con của biểu đồ được gọi là biểu đồ con, phần con của cây quyết định được gọi là cây con. Nó đại diện cho một phần cụ thể của cây quyết định.
* Pruning (Cắt tỉa): Quá trình loại bỏ hoặc cắt giảm các nút cụ thể trong cây quyết định để ngăn chặn việc trang bị quá mức và đơn giản hóa mô hình.
* Branch / Sub-Tree (Nhánh / Cây con): Một phần con của toàn bộ cây quyết định được gọi là nhánh hoặc cây con. Nó đại diện cho một con đường cụ thể của các quyết định và kết quả trong cây.
* Parent and Child Node (Nút cha và con): Trong cây quyết định, một nút được chia thành các nút con được gọi là nút cha và các nút con xuất hiện từ nó được gọi là nút con. Nút cha thể hiện một quyết định hoặc điều kiện, trong khi các nút con thể hiện các kết quả tiềm năng hoặc các quyết định tiếp theo dựa trên điều kiện đó.



* 1. **Cách hình thành cây quyết định**

Quá trình hình thành cây quyết định bao gồm việc phân vùng đệ quy dữ liệu dựa trên giá trị của các thuộc tính khác nhau. Thuật toán chọn thuộc tính tốt nhất để phân chia dữ liệu tại mỗi nút bên trong, dựa trên các tiêu chí nhất định như mức tăng thông tin hoặc tạp chất Gini. Quá trình phân tách này tiếp tục cho đến khi đáp ứng một tiêu chí dừng, chẳng hạn như đạt đến độ sâu tối đa hoặc có số lượng phiên bản tối thiểu trong một nút lá.

* 1. **Information Gain**

Khi chúng ta sử dụng một nút trong cây quyết định để phân chia các trường hợp huấn luyện thành các tập con nhỏ hơn thì entropy sẽ thay đổi. Độ lợi thông tin là thước đo sự thay đổi entropy này.

* Giả sử S là một tập hợp các thể hiện,
* A là một thuộc tính
* Sv là tập con của S
* v đại diện cho một giá trị riêng lẻ mà thuộc tính A có thể nhận và Giá trị (A) là tập hợp tất cả
* các giá trị có thể có của A thì



Entropy: là thước đo độ không chắc chắn của một biến ngẫu nhiên, nó đặc trưng cho mức độ tạp chất của một tập hợp các ví dụ tùy ý. Entropy càng cao thì nội dung thông tin càng nhiều.

Giả sử S là tập hợp các thể hiện, A là thuộc tính, Sv là tập con của S với A = v và Giá trị (A) là tập hợp tất cả các giá trị có thể có của A, khi đó:



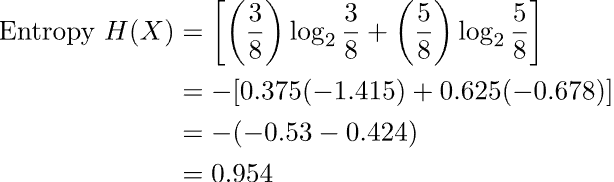
Ví dụ:

For the set X = {a,a,a,b,b,b,b,b}

Total instances: 8

Instances of b: 5

Instances of a: 3



Xây dựng cây quyết định bằng cách sử dụng thông tin thu được Các yếu tố cần thiết:

* Bắt đầu với tất cả các trường hợp huấn luyện được liên kết với nút gốc
* Sử dụng thông tin thu được để chọn thuộc tính nào để gắn nhãn cho mỗi nút
* Lưu ý: Không có đường dẫn từ gốc tới lá nào được chứa cùng một thuộc tính rời rạc hai lần
* Xây dựng đệ quy từng cây con trên tập hợp con của các trường hợp huấn luyện sẽ được phân loại theo đường dẫn đó trong cây. Nếu tất cả các trường hợp huấn luyện tích cực hoặc tiêu cực vẫn còn, gắn nhãn nút đó là "có" hoặc "không" tương ứng
* Nếu không còn thuộc tính nào, gắn nhãn với đa số phiếu bầu của các trường hợp đào tạo còn lại tại nút đó. Nếu không còn trường hợp nào, gắn nhãn với đa số phiếu bầu của các trường hợp đào tạo của cha mẹ.
  1. **Gini Index**
* Chỉ số Gini là thước đo tần suất một phần tử được chọn ngẫu nhiên bị xác định sai.
* Điều đó có nghĩa là thuộc tính có chỉ số Gini thấp hơn sẽ được ưu tiên.
* Sklearn hỗ trợ tiêu chí "Gini" cho Chỉ số Gini và theo mặc định, nó lấy giá trị "gini".

Chỉ số Gini là thước đo mức độ bất bình đẳng hoặc tạp chất của phân phối, thường được sử dụng trong cây quyết định và các thuật toán học máy khác. Nó nằm trong khoảng từ 0 đến 1, trong đó 0 đại diện cho sự bình đẳng hoàn hảo (tất cả các giá trị đều giống nhau) và 1 đại diện cho sự bất bình đẳng hoàn hảo (tất cả các giá trị đều khác nhau).

Một số tính năng và đặc điểm bổ sung của Chỉ số Gini là:

* Nó được tính bằng cách tính tổng các xác suất bình phương của từng kết quả trong một phân phối và trừ kết quả từ 1.
* Chỉ số Gini thấp hơn biểu thị sự phân phối đồng nhất hoặc thuần túy hơn, trong khi Chỉ số Gini cao hơn biểu thị sự phân phối không đồng nhất hoặc không tinh khiết hơn.
* Trong cây quyết định, Chỉ số Gini được sử dụng để đánh giá chất lượng phân chia bằng cách đo lường sự khác biệt giữa tạp chất của nút cha và tạp chất có trọng số của các nút con.
* So với các thước đo tạp chất khác như entropy, Chỉ số Gini được tính toán nhanh hơn và nhạy cảm hơn với những thay đổi về xác suất của lớp.
* Một nhược điểm của Chỉ số Gini là nó có xu hướng thiên về việc phân chia để tạo ra các nút con có kích thước bằng nhau, ngay cả khi chúng không tối ưu về độ chính xác phân loại.
* Trong thực tế, việc lựa chọn giữa việc sử dụng Chỉ số Gini hoặc các thước đo tạp chất khác phụ thuộc vào vấn đề và tập dữ liệu cụ thể, đồng thời thường yêu cầu thử nghiệm và điều chỉnh.

## Evaluation

* 1. **Accuracy**

Accuracy là tỷ lệ số lần dự đoán đúng trên tổng số lượt dự đoán. Trong dòng code này, bạn in ra giá trị trung bình của cvs làm đại diện cho accuracy và độ biến động là hai lần độ lệch chuẩn của cvs. Nó giúp đánh giá mức độ ổn định của mô hình.

* 1. **Precision**

Precision đo lường tỷ lệ các dự đoán dương tính (positive predictions) đúng trên tổng số các dự đoán dương tính. Nói cách khác, nó đo lường khả năng của mô hình khi dự đoán positive.

* 1. **Recall**

Recall (sensitivity) là tỷ lệ các dự đoán dương tính đúng trên tổng số các thực tế dương tính. Nó đo lường khả năng của mô hình để bắt capture tất cả các trường hợp positive.

# Triển khai

## Công nghệ áp dụng

Google Collab, python.

## Thu thập dữ liệu (dataset)

Dữ liệu gồm 2 file: train.csv (891 record), test.csv (418 record)

Tỉ lệ phân chia dữ liệu cho việc train và test là khoảng 70% và 30%.

Dataset gồm 12 cột thuộc tính: “PassengerId”, “Survived ”, “Pclass”, “Name”, “Sex”, “Age”, “SibSp”, “Parch”, “Ticket”, “Fare”, “Cabin”, “Embarked”.

## Giải thích code

|  |
| --- |
|  |
| Hình 3. Import thư viện cần thiết |

|  |
| --- |
|  |
| Hình 4. Load dữ liệu |

|  |
| --- |
|  |
| Hình 5. Kiểm tra thông tin của tập dữ liệu, data validation |

|  |
| --- |
|  |
| Hình 6. Feature Engineering |

|  |
| --- |
|  |
| Hình 7. Data preprocesing( Tiền xử lý dữ liệu). |

|  |
| --- |
|  |
| Hình 8. Xây dựng pipeline bao gồm việc tiền xử lý và phân loại |

|  |
| --- |
|  |
| Hình 9. Phân chia tập dữ liệu training/ validation |

|  |
| --- |
|  |
| Hình 10. Huấn luyện mô hình với pipeline |

|  |
| --- |
|  |
| Hình 11. Đánh giá hiệu suất mô hình máy học sử dụng dữ liệu huấn luyện |

|  |
| --- |
|  |
| Hình 12. Đánh giá mô hình dựa trên tập validation |

|  |
| --- |
|  |
| Hình 13. Code kiểm tra độ chính xác của mô hình tương ứng với từng max\_depth |

|  |
| --- |
|  |
| Hình 14. Dự đoán và Hiển thị cây quyết định |

## Kết quả và đánh giá

* 1. **Kết quả**

Áp dụng thành công mô hình pipeline phân loại bằng cây quyết định.

* Accuracy: 0.7846
* Precision: 0.7662
* Recall: 0.6211
  1. **Đánh giá**

Nhìn chung, mô hình có độ chính xác tương đối cao, nhưng cần cải thiện thêm để tăng độ hiệu quả trong việc dự đoán phân loại.

# Kết luận

Pipeline trong machine learning là một cấu trúc giúp kết hợp và tổ chức các bước xử lý dữ liệu và huấn luyện mô hình thành một chuỗi logic và liên tục. Điều này bao gồm các bước tiền xử lý dữ liệu như chuẩn hóa, mã hóa, xử lý giá trị thiếu và các bước huấn luyện mô hình. Pipeline không chỉ giúp tự động hóa quá trình này mà còn tối ưu hóa hiệu suất của mô hình thông qua việc tự động tuning tham số và tổ chức logic các bước. Điều này giúp dễ dàng tái sử dụng, tích hợp linh hoạt vào các dự án khác nhau và giảm thiểu việc lặp lại công việc.

Cây quyết định là một trong những phương pháp mô hình hóa dữ liệu đơn giản và mạnh mẽ trong machine learning. Điểm mạnh lớn nhất của cây quyết định là tính giải thích rõ ràng và dễ hiểu, cho phép hiểu được quy trình ra quyết định của mô hình một cách trực quan. Khả năng xử lý cả dữ liệu dạng số liệu và phân loại làm cho cây quyết định trở nên linh hoạt và được ứng dụng rộng rãi trong nhiều lĩnh vực.

Tuy nhiên, cây quyết định cũng có nhược điểm. Một trong những vấn đề lớn nhất là khả năng bị overfitting khi cây quá sâu hoặc quá phức tạp. Khi mô hình quá tập trung vào việc "nhớ" dữ liệu huấn luyện thay vì học các quy luật tổng quát, hiệu suất trên dữ liệu mới sẽ bị giảm. Điều này yêu cầu sự cân nhắc trong việc điều chỉnh tham số để tránh overfitting và đảm bảo tính tổng quát hóa của mô hình.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | "Decision Tree in Machine Learning," GeeksforGeeks, 04 12 2023. [Online]. Available: https://www.geeksforgeeks.org/decision-tree-introduction-example/. |
| [2] | phamdinhkhanh, "Bài 1- Kỹ thuật feature engineering," 07 01 2019. [Online]. Available: https://phamdinhkhanh.github.io/2019/01/07/Ky\_thuat\_feature\_engineering.html. |
| [3] | T. Vũ, "Machine learning pipeline là gì? Lợi ích khi triển khai machine learning pipeline," 10 2022. [Online]. Available: https://pipedrive.hvn.vn/loi-ich-khi-xay-dung-machine-learning-pipeline-la-gi/. |
| [4] | N. C. Thắng, "K-Fold cross validation, đánh giá model hiệu quả hơn khi có ít dữ liệu," 18 01 2021. [Online]. Available: https://miai.vn/2021/01/18/k-fold-cross-validation-tuyet-chieu-train-khi-it-du-lieu/. |
| [5] | "Giới thiệu về k-fold cross validation," 03 01 2020. [Online]. Available: https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/. |
| [6] | "Data preprocessing là gì? Bước không thể thiếu trong khai thác dữ liệu," 19 12 2022. [Online]. Available: https://meeyland.com/tin-tuc/data-preprocessing-la-gi-buoc-khong-the-thieu-trong-khai-thac-du-lieu-378177990. |
| [7] | "Feature engineering là gì," 29 10 2015. [Online]. Available: https://ongxuanhong.wordpress.com/2015/10/29/feature-engineering-la-gi/#:~:text=Feature%20Engineering%20l%C3%A0%20q%C3%BAa%20tr%C3%ACnh,h%C3%ACnh%20d%E1%BB%B1%20%C4%91o%C3%A1n%20hi%E1%BB%87n%20t%E1%BA%A1i.. |
| [8] | phamdinhkhanh, "Sklearn Pipeline," 2021. [Online]. Available: https://phamdinhkhanh.github.io/deepai-book/ch\_appendix/index\_pipeline.html?fbclid=IwAR0tQBticDTCDwVoMOYbKPtiUX55YQaGuHfq8RtZVE0JBJvelViiZL7y4R8. |
| [9] | T. Vu, "Machine Learning pipeline," 2021. [Online]. Available: https://machinelearningcoban.com/tabml\_book/ch\_intro/pipeline.html. |