**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

****

**BÀI TẬP LỚN**

**HỌC MÁY CƠ BẢN**

**VIETNAMESE STUDENT FEEDBACK ANALYSIS USING MACHINE LEARNING**

NGÀNH: KHOA HỌC MÁY TÍNH

SINH VIÊN: **1. PHAN MẠNH CƯỜNG**

LỚP: **124221**

NGƯỜI HƯỚNG DẪN: **TS. HOÀNG QUỐC VIỆT**

**HƯNG YÊN – 2025**

NHẬN XÉT

**Nhận xét của giảng viên hướng dẫn:**

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

**GIẢNG VIÊN HƯỚNG DẪN**

**Hoàng Quốc Việt**

**MỤC LỤC**

[DANH SÁCH HÌNH VẼ 3](#_Toc186763179)

[DANH SÁCH BẢNG BIỂU 4](#_Toc186763180)

[DANH SÁCH TỪ VIẾT TẮT 5](#_Toc186763181)

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 6](#_Toc186763182)

[1.1 Bài toán 6](#_Toc186763183)

[1.2 Trình bài dữ liệu bài toán 7](#_Toc186763184)

[1.3 Tiền xử lý dữ liệu 8](#_Toc186763185)

[1.4 Thống kê dữ liệu 9](#_Toc186763186)

[1.5 Trực quan hóa dữ liệu 9](#_Toc186763187)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 10](#_Toc186763188)

[2.1 Biểu diễn dữ liệu văn bản 10](#_Toc186763189)

[2.1.1 Bag of Words 10](#_Toc186763190)

[2.1.2 Term Frequency-Inverse Document Frequency (TF-IDF) 11](#_Toc186763191)

[2.1.3 Word embedding 12](#_Toc186763192)

[2.2 Feature Engineering 13](#_Toc186763193)

[2.2.1 MaxAbsScaler 13](#_Toc186763194)

[2.2.2 Principal Component Analysis 13](#_Toc186763195)

[2.2.3 Synthetic Minority Over-sampling Technique (SMOTE) 14](#_Toc186763196)

[2.3 Tinh chỉnh siêu tham số (HyperParameter Tuning) 15](#_Toc186763197)

[2.4 Kiểm định chéo (Cross Validation) 15](#_Toc186763198)

[2.5 Machine Learning & Deep Learning models 16](#_Toc186763199)

[2.5.1 Machine Learning 16](#_Toc186763200)

[2.5.2 Deep Learning 23](#_Toc186763201)

[2.6 Đánh giá mô hình (Model evaluation) 32](#_Toc186763202)

[2.6.1 Confusion Matrix 32](#_Toc186763203)

[2.6.2 Metrics đánh giá 34](#_Toc186763204)

[CHƯƠNG 3: GIẢI PHÁP 37](#_Toc186763205)

[3.1 37](#_Toc186763206)

[3.1.1 37](#_Toc186763207)

[KẾT LUẬN 38](#_Toc186763208)

[TÀI LIỆU THAM KHẢO 39](#_Toc186763209)

DANH SÁCH HÌNH VẼ

[Hình 2.1: Synthetic Minority Oversampling Techique 14](#_Toc186763169)

[Hình 2.2: Quy trình Ensemble Stacking 23](#_Toc186763170)

[Hình 2.3: Quy trình tính toán RNN 24](#_Toc186763171)

[Hình 2.4: Quy trình tính toán LSTM 26](#_Toc186763172)

[Hình 2.5: Mô hình BiLSTM 28](#_Toc186763173)

[Hình 2.6: Quy trình tính toán GRU 29](#_Toc186763174)

[Hình 2.7: Kiến trúc Transformers 31](#_Toc186763175)

[Hình 2.8: Cơ chế Attention 31](#_Toc186763176)

[Hình 2.9: Confusion matrix 33](#_Toc186763177)

[Hình 2.10: ROC-AUC 36](#_Toc186763178)

DANH SÁCH BẢNG BIỂU

**No table of figures entries found.**

DANH SÁCH TỪ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Từ đầy đủ | Giải thích |
| 1 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# GIỚI THIỆU BÀI TOÁN

* 1. Bài toán

Trong bối cảnh giáo dục hiện đại, việc hiểu và phân tích phản hồi của sinh viên ngày càng trở thành một lĩnh vực quan trọng. Phản hồi của sinh viên chứa đựng thông tin giá trị không chỉ giúp cải thiện chất lượng giảng dạy mà còn nâng cao trải nghiệm học tập của họ. Việc áp dụng các công nghệ học máy (Machine Learning) và học sâu (Deep Learning) vào phân tích phản hồi sinh viên mang lại tiềm năng to lớn trong việc hỗ trợ các tổ chức giáo dục đưa ra quyết định dựa trên dữ liệu.

Bài toán phân tích phản hồi sinh viên tập trung vào việc nhận diện và phân loại nội dung trong các phản hồi, chẳng hạn như xác định các ý kiến tích cực, tiêu cực, hoặc trung lập. Dữ liệu phân tích có thể đến từ các khảo sát, bài đánh giá khóa học, hoặc các cuộc thảo luận trực tuyến. Những dữ liệu này thường mang tính chủ quan cao và đòi hỏi các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để khai thác hiệu quả.

Một số phương pháp truyền thống như Bag of Words (BoW) và TF-IDF (Term Frequency-Inverse Document Frequency) được sử dụng phổ biến để biểu diễn văn bản. BoW tập trung vào việc đếm tần suất các từ trong văn bản, trong khi TF-IDF cải thiện bằng cách xác định mức độ quan trọng của từ dựa trên tần suất xuất hiện của nó trong một tập dữ liệu lớn hơn. Dù đơn giản và dễ triển khai, những phương pháp này có hạn chế trong việc xử lý ngữ cảnh và cấu trúc ngôn ngữ phức tạp, đặc biệt là trong các đoạn văn bản dài và chứa nhiều cảm xúc đa chiều.

Sự xuất hiện của các mô hình học sâu như LSTM, GRU, và Transformer đã mở ra hướng đi mới cho việc phân tích phản hồi. Kết hợp với BoW, TF-IDF, hoặc các phương pháp nhúng từ tiên tiến như Word2Vec, GloVe, hoặc BERT, các mô hình này không chỉ cải thiện khả năng biểu diễn văn bản mà còn đạt được độ chính xác vượt trội trong việc phân tích phản hồi sinh viên.

Việc ứng dụng các kỹ thuật tiên tiến này giúp tự động hóa quy trình phân tích quy mô lớn, cung cấp thông tin giá trị để hỗ trợ nhà trường và giảng viên:

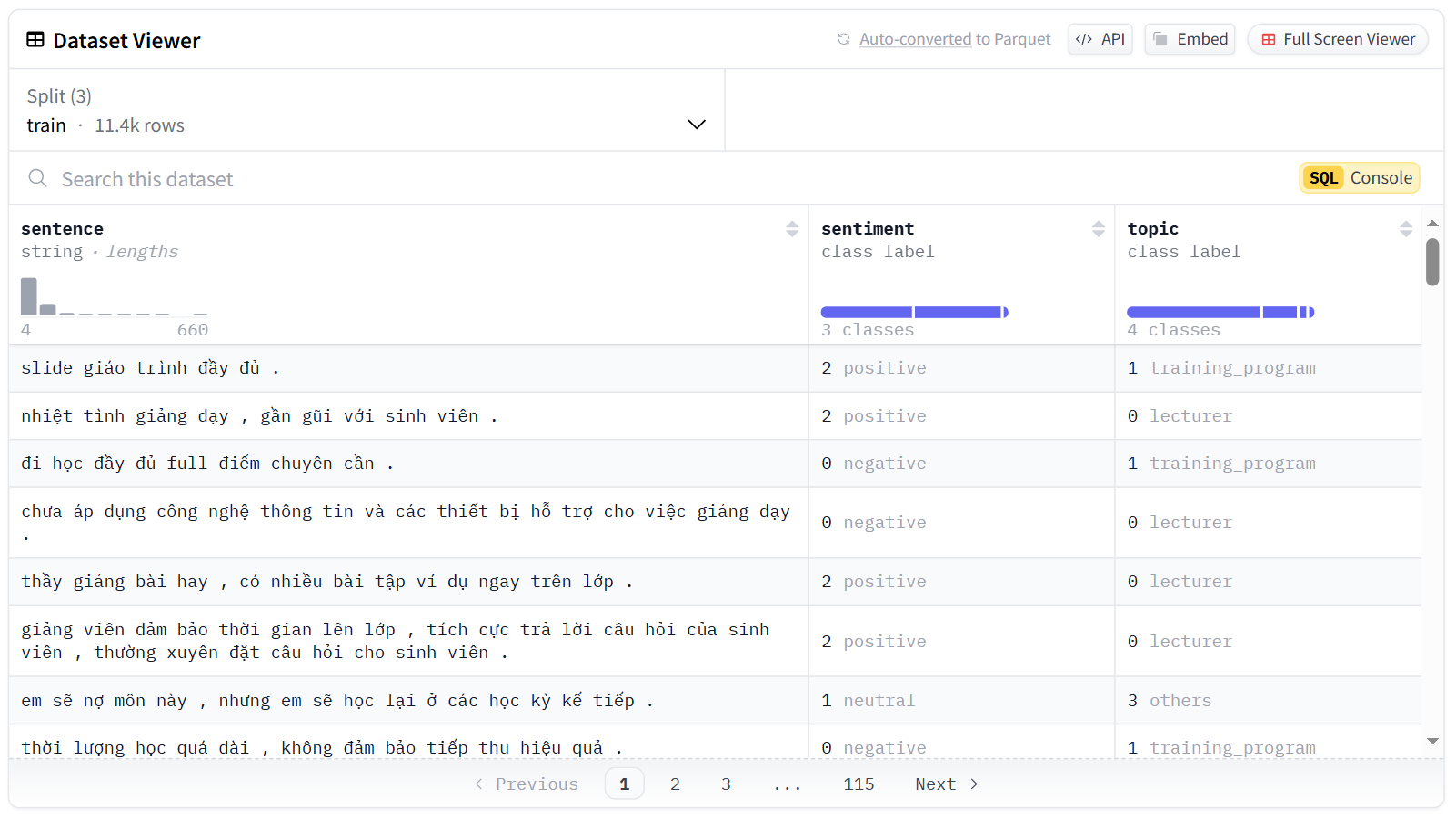
* **Nâng cao chất lượng giảng dạy**: Phân tích phản hồi giúp giảng viên điều chỉnh nội dung và phương pháp giảng dạy phù hợp hơn.
* **Cải thiện trải nghiệm sinh viên:** Xác định những khó khăn hoặc vấn đề sinh viên gặp phải để đưa ra giải pháp kịp thời.
* **Hỗ trợ ra quyết định chiến lược:** Cung cấp dữ liệu định lượng và định tính để tối ưu hóa chương trình đào tạo.

Mục tiêu cuối cùng của bài toán là xây dựng một hệ thống phân tích phản hồi sinh viên hiệu quả, góp phần nâng cao chất lượng giáo dục và tạo môi trường học tập lý tưởng. Điều này không chỉ giúp các tổ chức giáo dục cải thiện dịch vụ mà còn thúc đẩy sự phát triển của công nghệ xử lý ngôn ngữ tự nhiên trong lĩnh vực giáo dục.

* 1. Trình bài dữ liệu bài toán

Dữ liệu được lấy từ trang web:

<https://huggingface.co/datasets/uitnlp/vietnamese_students_feedback>



Đây là bộ dữ liệu phân loại cảm xúc và chủ đề dựa trên văn bản từ phản hồi của sinh viên với nhà trường.

Dữ liệu bài toán gồm có các đặc trưng sau:

* Sentence: Nội dung bao gồm 16175 câu phản hồi của sinh viên.
* Sentiment: Cảm xúc của câu phản hồi.
  + Nhãn 0: Tiêu cực
  + Nhãn 1: Trung tính
  + Nhãn 2: Tích cực
* Topic: Chủ đề của câu phản hồi.
  + Nhãn 0: Giảng viên
  + Nhãn 1: Chương trình giảng dạy
  + Nhãn 2: Cơ sở vật chất
  + Nhãn 3: Khác
  1. Tiền xử lý dữ liệu

1. Xóa các các bản ghi trùng lặp.
2. Làm sạch dữ liệu văn bản.
3. Biểu diễn dữ liệu dạng văn bản.
4. Phân chia dữ liệu huấn luyện và kiểm tra.
5. Xử lý mất cân bằng dữ liệu.
6. Chuẩn hóa dữ liệu (MaxAbsScaler).
   1. Thống kê dữ liệu
7. Thống kê mô tả dữ liệu.
8. Thống kê thông tin các cột.
9. Thống kê số lượng các bản ghi trùng lặp.
10. Hiển thị số lượng từ trong từ điển trước và sau khi làm sạch dữ liệu.
    1. Trực quan hóa dữ liệu
11. Biểu đồ hiển thị số lượng mỗi nhãn trước và sau khi xử lý mất cân bằng.
12. Biểu đồ phân bố độ dài các câu.
13. Biểu đồ hiển thị top những từ xuất hiện nhiều nhất trước và sau khi làm sạch dữ liệu.
14. Biểu đồ WordCloud hiển thị các từ trước và sau khi làm sạch dữ liệu

# CƠ SỞ LÝ THUYẾT

* 1. Biểu diễn dữ liệu văn bản

Biểu diễn văn bản là quá trình chuyển đổi dữ liệu văn bản thô thành các dạng số hóa, giúp máy tính có thể hiểu và xử lý. Quá trình này đóng vai trò quan trọng trong các bài toán xử lý ngôn ngữ tự nhiên, tạo điều kiện cho các mô hình học máy và học sâu phân tích, dự đoán và trích xuất thông tin từ văn bản.

* + 1. Bag of Words

1. Định nghĩa

BoW (Bag of Words) là một phương pháp đơn giản và phổ biến trong xử lý ngôn ngữ tự nhiên (NLP) để biểu diễn văn bản dưới dạng số. Phương pháp này tập trung vào tần suất xuất hiện của các từ trong văn bản mà không xét đến ngữ cảnh hay thứ tự từ. BoW thường được sử dụng trong các bài toán phân loại văn bản, phân tích cảm xúc, và tìm kiếm thông tin.

1. Cấu trúc

**Từ điển**: Tập hợp tất cả các từ duy nhất trong tập dữ liệu.

**Vector đặc trưng:** Mỗi văn bản được biểu diễn bằng một vector, trong đó mỗi phần tử là tần suất xuất hiện của từ tương ứng trong từ điển.

1. Ưu nhược điểm

Ưu điểm:

* Đơn giản và dễ triển khai.
* Hiệu quả với các tập dữ liệu nhỏ và văn bản ngắn.

Nhược điểm:

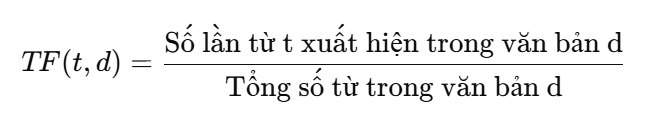
* Không xét đến ngữ cảnh hoặc mối quan hệ giữa các từ.
* Ma trận biểu diễn thưa và kích thước lớn khi từ điển mở rộng.
* Khó xử lý các văn bản phức tạp hoặc có ngữ nghĩa sâu.
  + 1. Term Frequency-Inverse Document Frequency (TF-IDF)

1. Định nghĩa

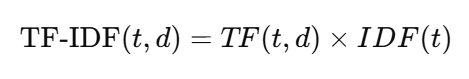
TF-IDF là một phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên (NLP) để mã hóa văn bản thành các đặc trưng số. Mục tiêu của TF-IDF là xác định mức độ quan trọng của một từ trong một văn bản so với một tập hợp các văn bản khác. Phương pháp này đặc biệt hữu ích trong các bài toán phân loại văn bản và tìm kiếm thông tin.

1. Cấu trúc

* F (Term Frequency): đo lường tần suất xuất hiện của một từ trong văn bản. Công thức tính TF như sau:



* IDF (Inverse Document Frequency): đo lường mức độ quan trọng của một từ trong toàn bộ tập văn bản. Công thức tính IDF là:



1. Ưu nhược điểm

Ưu điểm:

* Giúp mô hình hiểu được mức độ quan trọng của từ.
* Không yêu cầu từ điển cố định.

Nhược điểm:

* Không tính đến ngữ cảnh.
* Không thể hiện mối quan hệ từ trong câu, khó mô tả với dữ liệu phức tạp.
* Kích thước từ điển lớn và ma trận từ rất thưa ro có nhiều số 0.
  + 1. Word embedding

1. Định nghĩa

Word Embedding là một phương pháp biểu diễn từ dưới dạng vector số trong không gian liên tục, trong đó các từ có ý nghĩa tương tự sẽ có các vector gần nhau. Phương pháp này giúp mã hóa ngữ nghĩa và ngữ cảnh của từ, vượt qua những hạn chế của các phương pháp truyền thống như BoW hay TF-IDF. Word Embedding thường được sử dụng trong các bài toán xử lý ngôn ngữ tự nhiên hiện đại như phân loại văn bản, dịch máy, và phân tích cảm xúc.

1. Cấu trúc

* Vector nhúng: Mỗi từ được biểu diễn bằng một vector với số chiều cố định, thường được học thông qua mô hình như Word2Vec, GloVe, hoặc BERT.
* Không gian vector: Các từ tương tự về ngữ nghĩa nằm gần nhau trong không gian vector.

1. Ưu nhược điểm

Ưu điểm:

* Mã hóa được ngữ nghĩa và ngữ cảnh.
* Giảm kích thước không gian đặc trưng so với BoW và TF-IDF.
* Tương thích tốt với các mô hình học sâu hiện đại.

Nhược điểm:

* Cần nhiều dữ liệu và tài nguyên tính toán để huấn luyện.
* Hiệu quả phụ thuộc vào chất lượng và kích thước của tập dữ liệu huấn luyện.
* Khó xử lý các từ hoặc cụm từ hiếm nếu không có trong tập huấn luyện.
  1. Feature Engineering

Feature Engineering là một bước quan trọng trong quy trình xử lý dữ liệu, giúp cải thiện hiệu suất của mô hình học máy bằng cách chuyển đổi hoặc tạo ra các đặc trưng quan trọng từ dữ liệu thô. Dưới đây là các kỹ thuật thường được sử dụng:

* + 1. MaxAbsScaler

MaxAbsScaler là một kỹ thuật chuẩn hóa đặc trưng, biến đổi các giá trị trong tập dữ liệu về khoảng [-1, 1] bằng cách chia mỗi giá trị cho giá trị tuyệt đối lớn nhất trong cột tương ứng.

Ưu điểm:

* Bảo toàn dạng thưa (sparse) của dữ liệu, phù hợp với dữ liệu có nhiều số 0.
* Đơn giản, hiệu quả với dữ liệu có giá trị lớn nhưng không lệch quá nhiều.

Nhược điểm:

* Không xử lý tốt khi dữ liệu chứa nhiễu hoặc các ngoại lệ.
  + 1. Principal Component Analysis

PCA là một kỹ thuật giảm chiều dữ liệu, chuyển đổi các đặc trưng ban đầu thành các thành phần chính (principal components), đảm bảo giữ lại phần lớn thông tin của dữ liệu gốc.

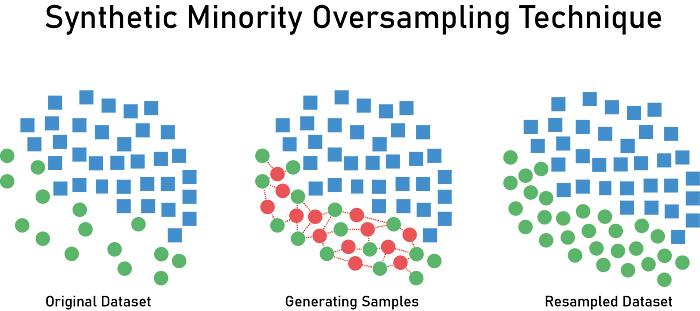
Ưu điểm:

* Giảm chiều dữ liệu, tăng tốc độ huấn luyện mô hình.
* Loại bỏ mối quan hệ tương quan giữa các đặc trưng.

Nhược điểm:

* Khó diễn giải ý nghĩa của các thành phần chính.
* Mất thông tin khi giảm chiều dữ liệu quá mức.
  + 1. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE là một kỹ thuật xử lý dữ liệu mất cân bằng bằng cách tạo thêm các mẫu giả lập cho lớp thiểu số. Phương pháp này sử dụng nội suy giữa các điểm dữ liệu hiện có để sinh ra các mẫu mới.



Hình 2.1: Synthetic Minority Oversampling Techique

Ưu điểm:

* Cân bằng dữ liệu hiệu quả mà không làm mất dữ liệu ban đầu.
* Tăng cường khả năng học của mô hình trên lớp thiểu số.

Nhược điểm:

* Có thể sinh ra các mẫu không tự nhiên, dẫn đến giảm tính chính xác.
* Không xử lý được sự chồng lấn giữa các lớp dữ liệu.
  1. Tinh chỉnh siêu tham số (HyperParameter Tuning)

Tinh chỉnh siêu tham số là quá trình điều chỉnh các tham số của mô hình học máy (siêu tham số) để tối ưu hóa hiệu suất của mô hình. Siêu tham số không được học trực tiếp từ dữ liệu mà được thiết lập trước khi huấn luyện mô hình, như số lượng cây trong rừng ngẫu nhiên, tỷ lệ học trong mô hình mạng nơ-ron, hoặc độ sâu của cây quyết định.

Phương pháp:

* **Grid Search**: Kiểm tra tất cả các kết hợp có thể của siêu tham số trong một phạm vi đã định.
* **Random Search**: Chọn ngẫu nhiên các giá trị từ một phạm vi và kiểm tra chúng, giúp giảm thời gian tính toán.
* **Bayesian Optimization**: Sử dụng các mô hình thống kê để tối ưu hóa siêu tham số, thay vì thử tất cả các kết hợp có thể.

Ưu điểm:

* Giúp tối ưu hóa hiệu suất mô hình.
* Cải thiện khả năng dự đoán của mô hình.

Nhược điểm:

* Tốn thời gian và tài nguyên tính toán, đặc biệt đối với mô hình phức tạp.
* Không phải lúc nào cũng dễ dàng xác định phạm vi siêu tham số hợp lý.
  1. Kiểm định chéo (Cross Validation)

Kiểm định chéo là một kỹ thuật đánh giá mô hình học máy nhằm giảm thiểu vấn đề overfitting và tăng tính tổng quát của mô hình. Quá trình kiểm định chéo chia dữ liệu thành nhiều phần (folds), mỗi phần sẽ lần lượt được sử dụng làm dữ liệu kiểm tra trong khi các phần còn lại được sử dụng làm dữ liệu huấn luyện.

Phương pháp:

* **K-fold Cross Validation:** Dữ liệu được chia thành K phần, và mô hình được huấn luyện và đánh giá K lần, mỗi lần sử dụng một phần khác nhau làm dữ liệu kiểm tra.
* **Leave-One-Out Cross Validation (LOOCV):** Mỗi quan sát trong dữ liệu sẽ được sử dụng làm dữ liệu kiểm tra một lần duy nhất, và phần còn lại làm dữ liệu huấn luyện.
* **Stratified K-fold Cross Validation:** Phân chia dữ liệu sao cho tỷ lệ các lớp trong mỗi fold giống với tỷ lệ của toàn bộ tập dữ liệu.

Ưu điểm:

* Cung cấp một ước tính chính xác hơn về hiệu suất mô hình.
* Giảm thiểu bias và overfitting bằng cách sử dụng tất cả dữ liệu để huấn luyện và kiểm tra.

Nhược điểm:

* Tốn thời gian tính toán, đặc biệt với các mô hình phức tạp hoặc tập dữ liệu lớn.
* Các phương pháp kiểm định chéo có thể gây ra sự chồng chéo nếu dữ liệu không được chia đúng cách.
  1. Machine Learning & Deep Learning models
     1. Machine Learning
        1. Support Vector Machine (SVM)

1. Nền tảng lý thuyết

Support Vector Machine (SVM) là một thuật toán học máy có giám sát, được sử dụng phổ biến trong các bài toán phân loại và hồi quy. SVM được áp dụng để tìm ra siêu phẳng (hyperplane) tối ưu phân tách các lớp khác nhau trong không gian đặc trưng.

SVM hoạt động dựa trên nguyên tắc:

* Tìm kiếm siêu phẳng tối ưu có khoảng cách lớn nhất (maximum margin) đến các điểm dữ liệu gần nhất của mỗi lớp, gọi là các vector hỗ trợ (support vectors).
* Với các bài toán không thể phân tách tuyến tính, SVM sử dụng hàm kernel để ánh xạ dữ liệu từ không gian ban đầu sang không gian có chiều cao hơn, nơi dữ liệu có thể được phân tách tuyến tính.

1. Công thức tính toán

- Siêu phẳng phân tách tuyến tính:

Một siêu phẳng trong không gian d-chiều có thể được biểu diễn bởi phương trình:

w⋅x+b=0

* w: vector trọng số (weight vector).
* x: vector đặc trưng của dữ liệu.
* b: hằng số (bias).

- Hàm mục tiêu:

Khoảng cách giữa các siêu phẳng và các vector hỗ trợ là:

SVM tối ưu hóa bài toán:

Với điều kiện ràng buộc:

* : nhãn (label) của dữ liệu (+1 hoặc -1).
* : vector đặc trưng của dữ liệu.

c, Hàm Kernal

Khi dữ liệu không thể phân tách tuyến tính, SVM sử dụng hàm kernel để ánh xạ dữ liệu sang chiều không gian cao hơn. Một số hàm kernel phổ biến:

* Linear kernel
* Polynomial kernel
* RBF kernel (Gaussian)

1. Ưu điểm

* Tính tổng quát cao: Nhờ tối ưu hóa khoảng cách giữa các lớp, SVM thường đạt hiệu suất tốt trên tập dữ liệu chưa thấy.
* Hiệu quả trên dữ liệu phi tuyến: Hàm kernel cho phép xử lý các bài toán phức tạp, phi tuyến.
* Khả năng xử lý tốt với dữ liệu chiều cao: Thích hợp cho dữ liệu văn bản, vốn thường có số chiều rất lớn.

1. Nhược điểm

* Chi phí tính toán cao: SVM không hiệu quả trên các tập dữ liệu lớn do thời gian tính toán tăng nhanh với kích thước dữ liệu.
* Khó chọn tham số và kernel: Cần thử nghiệm nhiều để chọn được kernel và các siêu tham số (C, γ) phù hợp.
* Không hỗ trợ phân loại xác suất trực tiếp: SVM chỉ cho ra kết quả phân lớp, không đưa ra xác suất dự đoán.
  + - 1. Naïve Bayes

1. Nền tảng lý thuyết

Naive Bayes (NB) là một thuật toán học máy có giám sát, dựa trên Định lý Bayes và giả định rằng các đặc trưng đầu vào là độc lập với nhau (giả định "naive" - đơn giản). Dù giả định này không phải lúc nào cũng chính xác, nhưng NB vẫn thường đạt hiệu quả tốt, đặc biệt trong các bài toán phân loại.

1. Công thức tính toán

- Giả định độc lập có điều kiện:

NB giả định ràng tất cả các đặc trưng , , …, trong tập đặc trưng X là độc lập:

P(X|C) = P(|C) + P(|C) + … + P(|C)

Phương trình phân loại trở thành:

C = arg max(C)

- Tính xác suất:

* Xác suất tiên nghiệm P(C): Tỷ lệ xuất hiện của mỗi lớp trong tập huấn luyện.
* Xác suất có điều kiện : Tỷ lệ xuất hiện của đặc trưng trong các văn bản thuộc lớp C.

- NB trong phân loại:

* Multinomial Naive Bayes: Dùng cho dữ liệu rời rạc.
* Bernoulli Naive Bayes: Dùng cho dữ liệu nhị phân.
* Gaussian Naive Bayes: Dùng cho dữ liệu liên tục.

1. Ưu điểm

* Tốc độ nhanh: Naive Bayes rất nhẹ, phù hợp với tập dữ liệu lớn.
* Hiệu quả tốt trên dữ liệu nhiều chiều: Thích hợp cho dữ liệu văn bản, nơi mỗi từ là một đặc trưng.
* Không đòi hỏi nhiều siêu tham số: NB không cần tinh chỉnh tham số phức tạp.
* Khả năng mở rộng: NB có thể được mở rộng với các dạng phân phối khác nhau (Multinomial, Gaussian).

1. Nhược điểm

Giả định độc lập không thực tế: Các từ trong văn bản thường có liên quan với nhau, điều này có thể làm giảm độ chính xác.

Không linh hoạt với dữ liệu phức tạp: NB không thể nắm bắt tốt các mối quan hệ phức tạp giữa các đặc trưng.

Không xử lý tốt dữ liệu mất cân bằng: Nếu một lớp xuất hiện quá ít, NB có thể không dự đoán tốt lớp đó.

2.3.3 Logistic Regression

1. Nền tảng lý thuyết

Logistic Regression là một thuật toán học máy có giám sát thường được sử dụng cho bài toán phân loại nhị phân. Tuy nhiên, nó có thể được mở rộng để xử lý các bài toán phân loại đa nhãn (k>2) thông qua hai phương pháp chính:

* One-vs-Rest (OvR): Huấn luyện k mô hình logistic regression, mỗi mô hình phân biệt một lớp với các lớp còn lại.
* Softmax Regression (Multinomial Logistic Regression): Mở rộng logistic regression để dự đoán trực tiếp xác suất của từng lớp bằng hàm softmax.

1. Công thức tính toán

- Softmax Regression:

P(y = i|x) = ∀i∈{1,2,...,k}

Hàm softmax đảm bảo rằng:

* P(y = i|x) ∈ [0, 1].
* Tổng tất cả xác suất các nhãn bằng 1.

1. Ưu điểm

* Đơn giản và dễ hiểu: Logistic Regression dễ cài đặt và giải thích kết quả.
* Tính toán nhanh: Phù hợp với tập dữ liệu nhỏ đến trung bình.
* Khả năng mở rộng: Dễ dàng áp dụng cho bài toán đa nhãn thông qua OvR hoặc softmax.

d, Nhược điểm

* Hạn chế với dữ liệu phi tuyến: Logistic Regression chỉ hoạt động tốt khi dữ liệu có thể phân tách tuyến tính trong không gian đặc trưng.
* Không phù hợp với số lượng lớn đặc trưng: Khi dữ liệu có nhiều chiều, Logistic Regression có thể bị overfitting nếu không có kỹ thuật điều chuẩn.
* Phụ thuộc vào tiền xử lý: Hiệu quả của Logistic Regression phụ thuộc lớn vào cách trích xuất và biểu diễn đặc trưng.

2.3.4 XGBoots

1. Nền tảng lý thuyết

XGBoost (Extreme Gradient Boosting), được thiết kế cho bài toán phân loại. Đây là một thuật toán học máy mạnh mẽ dựa trên cây quyết định, áp dụng kỹ thuật boosting để cải thiện hiệu suất của mô hình.

**1. Boosting**

Boosting là một kỹ thuật ensemble học máy, nơi nhiều mô hình yếu (weak learners) được huấn luyện tuần tự, mỗi mô hình cố gắng sửa lỗi của mô hình trước đó. Mục tiêu là kết hợp các mô hình yếu để tạo ra một mô hình mạnh với hiệu suất cao hơn.

**2. Gradient Boosting**

Gradient Boosting tối ưu hóa mô hình thông qua việc giảm thiểu hàm mất mát bằng cách xây dựng tuần tự các cây quyết định, sử dụng đạo hàm gradient để hướng dẫn.

**3. XGBoost**

XGBoost là một cải tiến của Gradient Boosting, được tối ưu hóa về tốc độ và hiệu suất. Một số đặc điểm nổi bật:

* Tối ưu hóa hiệu suất: Tích hợp xử lý song song và quản lý bộ nhớ hiệu quả.
* Hàm mất mát tùy chỉnh: Hỗ trợ hàm mất mát log-loss cho bài toán phân loại.
* Regularization (L1 và L2): Giảm overfitting thông qua điều chuẩn trọng số.
* Xử lý dữ liệu thiếu: Tự động xử lý các giá trị bị thiếu trong dữ liệu.

1. Công thức tính toán

Các tham số quan trọng:

* max\_depth: Độ sâu tối đa của mỗi cây quyết định (quyết định khả năng phân chia dữ liệu).
* learning\_rate: Tốc độ học, xác định mức độ điều chỉnh mô hình qua mỗi cây.
* n\_estimators: Số lượng cây quyết định trong rừng.
* subsample: Tỷ lệ mẫu dữ liệu sử dụng cho mỗi cây (để giảm overfitting).
* colsample\_bytree: Tỷ lệ cột được chọn cho mỗi cây.
* objective: Hàm mục tiêu, thường là "binary: logistic" cho phân loại nhị phân hoặc "multi: softprob" cho đa nhãn.
* reg\_lambda và reg\_alpha: Điều chuẩn L2 và L1.

1. Ưu điểm

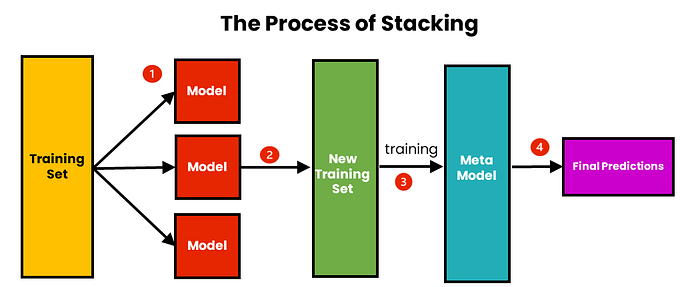
* Hiệu suất cao: XGBoost thường đạt kết quả tốt trên nhiều bài toán thực tế.
* Khả năng tổng quát hóa tốt: Điều chỉnh regularization giúp giảm overfitting.
* Tùy biến cao: Hỗ trợ nhiều tham số và hàm mất mát tùy chỉnh.
* Xử lý dữ liệu thiếu: Không yêu cầu tiền xử lý dữ liệu thiếu.

d, Nhược điểm

* Chi phí tính toán cao: Việc huấn luyện có thể mất nhiều thời gian, đặc biệt với dữ liệu lớn.
* Nhạy cảm với siêu tham số: Cần tinh chỉnh tham số (tuning) để đạt hiệu suất tối ưu.
* Không phù hợp cho dữ liệu ít: XGBoost cần một lượng dữ liệu đáng kể để phát huy hiệu quả.
  + - 1. Ensemble Stacking

Định nghĩa:

Ensemble Stacking là một kỹ thuật kết hợp các mô hình học máy mạnh mẽ để tạo ra một mô hình dự đoán cuối cùng tốt hơn. Các mô hình cơ sở (base models) được huấn luyện độc lập, và một mô hình học (thường là mô hình hồi quy hoặc phân loại) sẽ học từ các dự đoán của các mô hình cơ sở để đưa ra kết quả cuối cùng.



Hình 2.2: Quy trình Ensemble Stacking

Ưu điểm:

* Tăng độ chính xác của mô hình bằng cách kết hợp các dự đoán từ nhiều mô hình.
* Giảm thiểu bias và variance trong dự đoán.

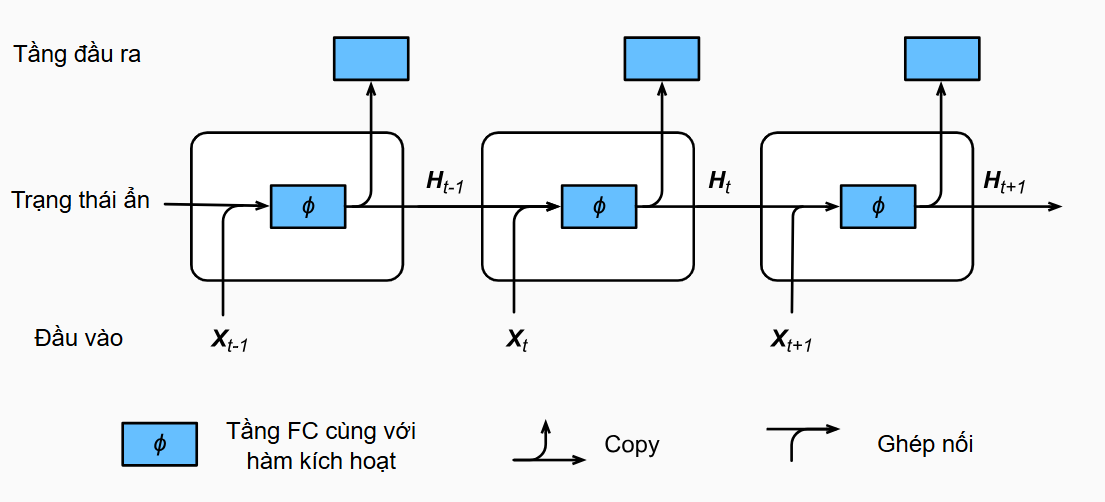
Nhược điểm:

* Tốn thời gian và tài nguyên tính toán hơn so với các mô hình đơn giản.
* Khó giải thích vì mô hình kết hợp nhiều mô hình khác nhau.
  + 1. Deep Learning
       1. RNN

RNN là một loại mạng nơ-ron đặc biệt được thiết kế để xử lý dữ liệu chuỗi. Với khả năng giữ lại thông tin của các bước trước trong quá trình học, RNN có thể học các mối quan hệ giữa các phần tử trong chuỗi dữ liệu, giúp giải quyết các bài toán như dịch máy, nhận dạng giọng nói, phân tích chuỗi thời gian, và nhiều bài toán khác.

Tuy nhiên, RNN truyền thống gặp phải một số vấn đề như **vanishing gradient** (mất dần độ dốc) khi xử lý các chuỗi dài. Điều này làm cho việc học các phụ thuộc dài hạn trở nên khó khăn. Mặc dù vậy, RNN vẫn là một lựa chọn phổ biến trong các bài toán dữ liệu tuần tự, khi các chuỗi không quá dài và có tính chất tuần tự đơn giản.

**Cấu trúc:**

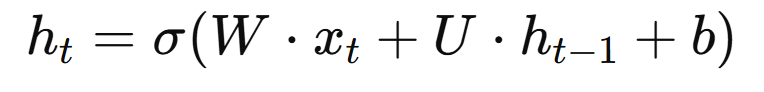
****

Hình 2.3: Quy trình tính toán RNN

Mạng RNN được xây dựng trên một cấu trúc tuần tự, trong đó đầu ra tại mỗi thời điểm t phụ thuộc vào đầu vào tại thời điểm t và trạng thái ẩn của thời điểm trước đó. Cấu trúc cơ bản của RNN chỉ có một lớp ẩn và các trọng số được chia sẻ qua các bước thời gian.

**Quy trình tính toán**:

* Tại mỗi thời điểm t, RNN nhận đầu vào và trạng thái ẩn ​ từ bước thời gian trước để tính toán trạng thái ẩn mới .
* Công thức tính trạng thái ẩn mới là:



Trong đó:

* : Đầu vào tại thời điểm ttt.
* : Trạng thái ẩn của thời điểm trước.
* W, U: Ma trận trọng số.
* b: Bias.
* σ: Hàm kích hoạt (thường là sigmoid hoặc tanh)

Ưu điểm

* Đơn giản và dễ hiểu: Cấu trúc của RNN khá đơn giản và dễ cài đặt, thích hợp cho các bài toán chuỗi cơ bản.
* Hiệu quả cho chuỗi ngắn: Khi dữ liệu có chuỗi ngắn và mối quan hệ giữa các yếu tố trong chuỗi đơn giản, RNN có thể học tốt.
* Tiết kiệm bộ nhớ: Vì trạng thái ẩn được chia sẻ qua các bước thời gian, RNN giúp tiết kiệm bộ nhớ trong việc huấn luyện.

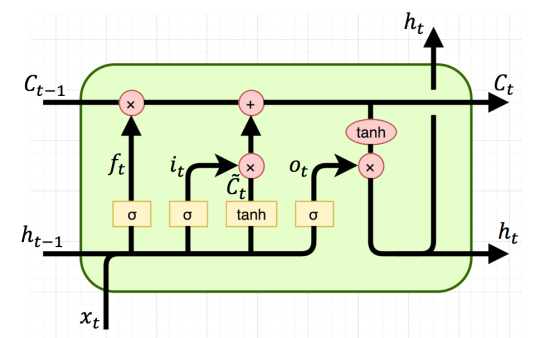
Nhược điểm

* Vanishing Gradient: Khi chuỗi trở nên dài, vấn đề vanishing gradient sẽ xuất hiện, làm cho mô hình khó học các phụ thuộc dài hạn trong chuỗi.
* Khó xử lý chuỗi dài: Mô hình này không hiệu quả khi đối mặt với các chuỗi dài hoặc dữ liệu có các mối quan hệ phức tạp.
* Khả năng mở rộng hạn chế: Mặc dù RNN có thể xử lý chuỗi dữ liệu, nhưng khả năng xử lý dữ liệu phức tạp của nó vẫn hạn chế so với các biến thể như LSTM và GRU.
  + - 1. LSTM

1. Nền tảng lý thuyết

LSTM (Long Short-Term Memory) và BiLSTM (Bidirectional LSTM) là hai loại kiến trúc mạng nơ-ron tái hồi (RNN) phổ biến, được thiết kế để xử lý dữ liệu tuần tự và giải quyết vấn đề về vanishing gradient thường gặp ở RNN truyền thống.

- Cấu trúc:



Hình 2.4: Quy trình tính toán LSTM

Mỗi cell LSTM có ba cổng chính:

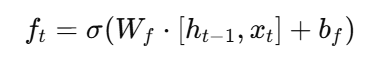
Forget Gate: Xác định thông tin nào từ trạng thái trước đó cần được quên.

Input Gate: Quyết định thông tin mới nào sẽ được thêm vào trạng thái.

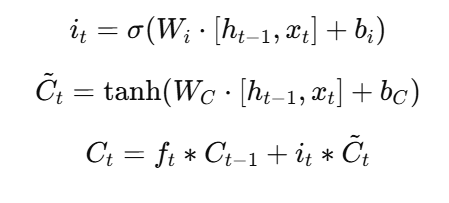
Output Gate: Xác định thông tin nào từ trạng thái hiện tại sẽ được đưa ra làm đầu ra.

1. Công thức toán học

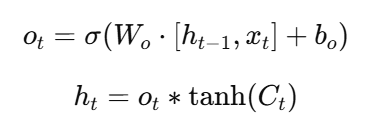
* Forget Gate:



* Input Gate:



* Output Gate:



Trong đó:

* đầu vào tại thời điểm t.
* trạng thái ẩn ở thời điểm trước.
* trạng thái bộ nhớ thời điểm t.
* σ: hàm sigmoid.
* tanh: hàm kích hoạt tanh.

1. Công dụng

* Phân tích chuỗi thời gian (time series analysis).
* Phân tích văn bản (sentiment analysis, dịch máy).
* Nhận dạng giọng nói.
* Dự đoán chuỗi (sequence prediction).

1. Ưu điểm

* Giải quyết tốt vấn đề phụ thuộc dài hạn.
* Hiệu quả trên các chuỗi dài với mối quan hệ phức tạp.

1. Nhược điểm

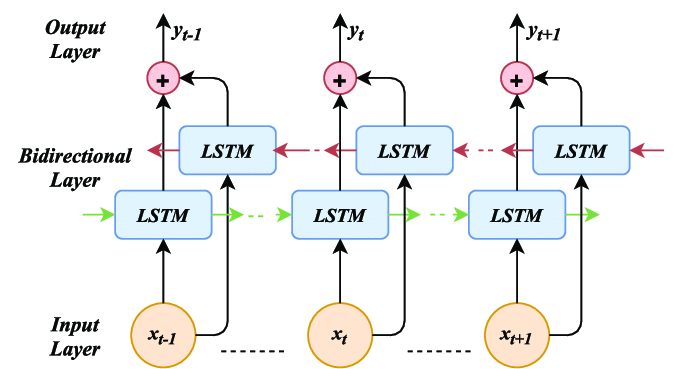
* Tốn tài nguyên tính toán (nhiều tham số).
* Dễ bị overfitting nếu dữ liệu ít.

2.4.2 BiLSTM

1. Nền tảng lý thuyết

BiLSTM (Bidirectional LSTM) mở rộng từ LSTM, cho phép mô hình học thông tin từ cả hai hướng (từ quá khứ đến hiện tại và từ tương lai đến hiện tại). Điều này hữu ích cho các bài toán mà ngữ cảnh trước và sau đều quan trọng.

1. Cấu trúc



Hình 2.5: Mô hình BiLSTM

BiLSTM sử dụng hai LSTM:

* Forward LSTM: Xử lý chuỗi từ trái sang phải.
* Backward LSTM: Xử lý chuỗi từ phải sang trái.

Đầu ra cuối cùng của BiLSTM là sự kết hợp (thường là ghép nối) giữa đầu ra của LSTM forward và backward.

1. Ưu điểm

* Sử dụng thông tin từ cả hai chiều, giúp hiểu ngữ cảnh toàn diện hơn.
* Hiệu quả với các bài toán chuỗi ngữ cảnh phức tạp.

1. Nhược điểm

* Tăng gấp đôi chi phí tính toán do sử dụng hai LSTM.
* Đòi hỏi nhiều tài nguyên hơn LSTM đơn hướng.

2.4.3 GRU

1. Nền tảng lý thuyết

GRU là một biến thể của mạng nơ-ron hồi tiếp (RNN), được thiết kế để giải quyết các vấn đề của RNN truyền thống, như vanishing gradient. GRU được phát triển sau LSTM và có cấu trúc đơn giản hơn, nhưng vẫn hiệu quả trong việc xử lý dữ liệu tuần tự.

- Cấu trúc:

GRU bao gồm hai cổng chính:

Ảnh có chứa phim hoạt hình, hình mẫu, thiết kế

Mô tả được tạo với mức tin cậy cao

Hình 2.6: Quy trình tính toán GRU

* Reset Gate: Quyết định mức độ thông tin từ trạng thái trước đó được quên.
* Update Gate: Quyết định mức độ thông tin mới được thêm vào và duy trì trạng thái trước đó.

1. Công thức tính học

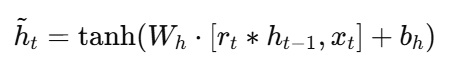
* Rest Gate:



* Update Gate:



* Trạng thái ấn mới:



* Trạng thái ẩn cuối cùng:



Trong đó:

* đầu vào tại thời điểm t.
* trạng thái ẩn ở thời điểm trước đó.
* giá trị của rest gate.
* giá trị của updata gate.
* , , : ma trận trọng số.
* , , vector bias
* σ: hàm sigmoid.
* tanh: hàm kích hoạt tanh.

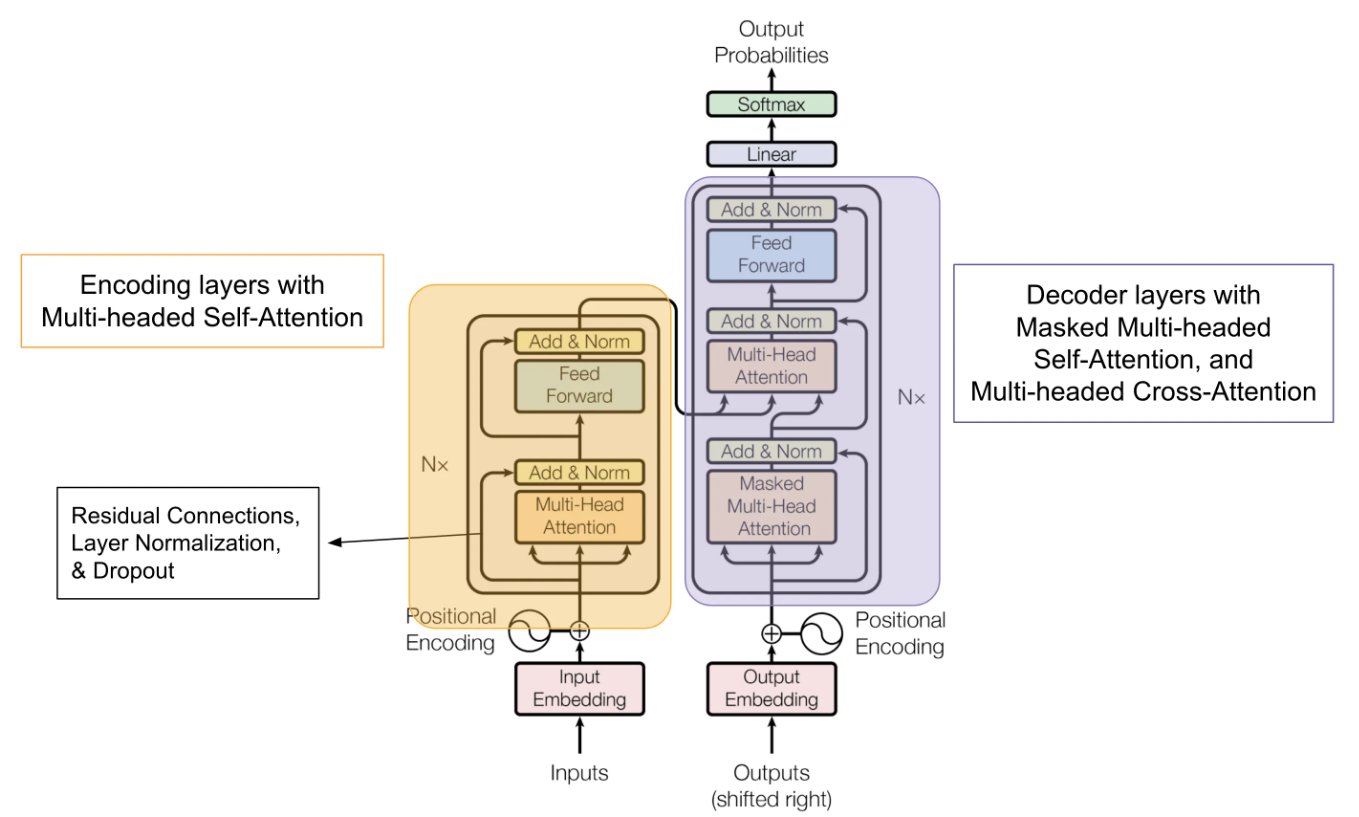
1. Ưu điểm

* Đơn giản hơn LSTM: GRU có ít tham số hơn, dẫn đến tốc độ huấn luyện nhanh hơn.
* Hiệu quả trên chuỗi ngắn: phù hợp với các bài toán mà chuỗi không có phụ thuộc dài hạn quá phức tạp.
* Tránh vanishing gradient: giải quyết tốt vấn đề gradient bị mất trong RNN truyền thống.

1. Nhược điểm

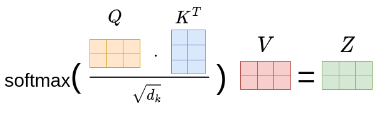
* Kém hiệu quả hơn trên chuỗi phức tạp: với các chuỗi có phụ thuộc dài và phức tạp, LSTM thường hoạt động tốt hơn.
* Khả năng điều chỉnh thấp hơn LSTM: GRU không có cổng forget riêng, dẫn đến việc kiểm soát thông tin phức tạp kém hơn.
  + - 1. Transformers

Transformers là một mô hình mạng nơ-ron mạnh mẽ được sử dụng trong các bài toán xử lý ngôn ngữ tự nhiên, đặc biệt là trong các mô hình như BERT và GPT. Được giới thiệu trong bài báo "Attention is All You Need", transformers sử dụng cơ chế attention để xác định mối quan hệ giữa các phần khác nhau trong câu, giúp mô hình hiểu ngữ cảnh một cách chính xác hơn.



Hình 2.7: Kiến trúc Transformers

Cơ chế Attention: Cơ chế attention giúp mô hình tập trung vào các phần quan trọng trong chuỗi đầu vào, thay vì chỉ sử dụng thông tin tuần tự như trong RNN. Mỗi từ trong câu có thể "chú ý" đến các từ khác, giúp mô hình học được mối quan hệ và ngữ cảnh giữa chúng.



Hình 2.8: Cơ chế Attention

Ưu điểm:

* Khả năng xử lý mối quan hệ giữa các từ trong câu một cách linh hoạt và hiệu quả.
* Dễ dàng song song hóa quá trình huấn luyện, giảm thời gian tính toán so với LSTM và GRU.
* Hiệu quả cao trong các tác vụ NLP, như dịch máy, phân loại văn bản, và sinh văn bản.

Nhược điểm:

* Yêu cầu nhiều tài nguyên tính toán, đặc biệt là khi huấn luyện trên dữ liệu quy mô lớn.
* Cần lượng lớn dữ liệu huấn luyện để đạt được kết quả tốt nhất.
  1. Đánh giá mô hình (Model evaluation)

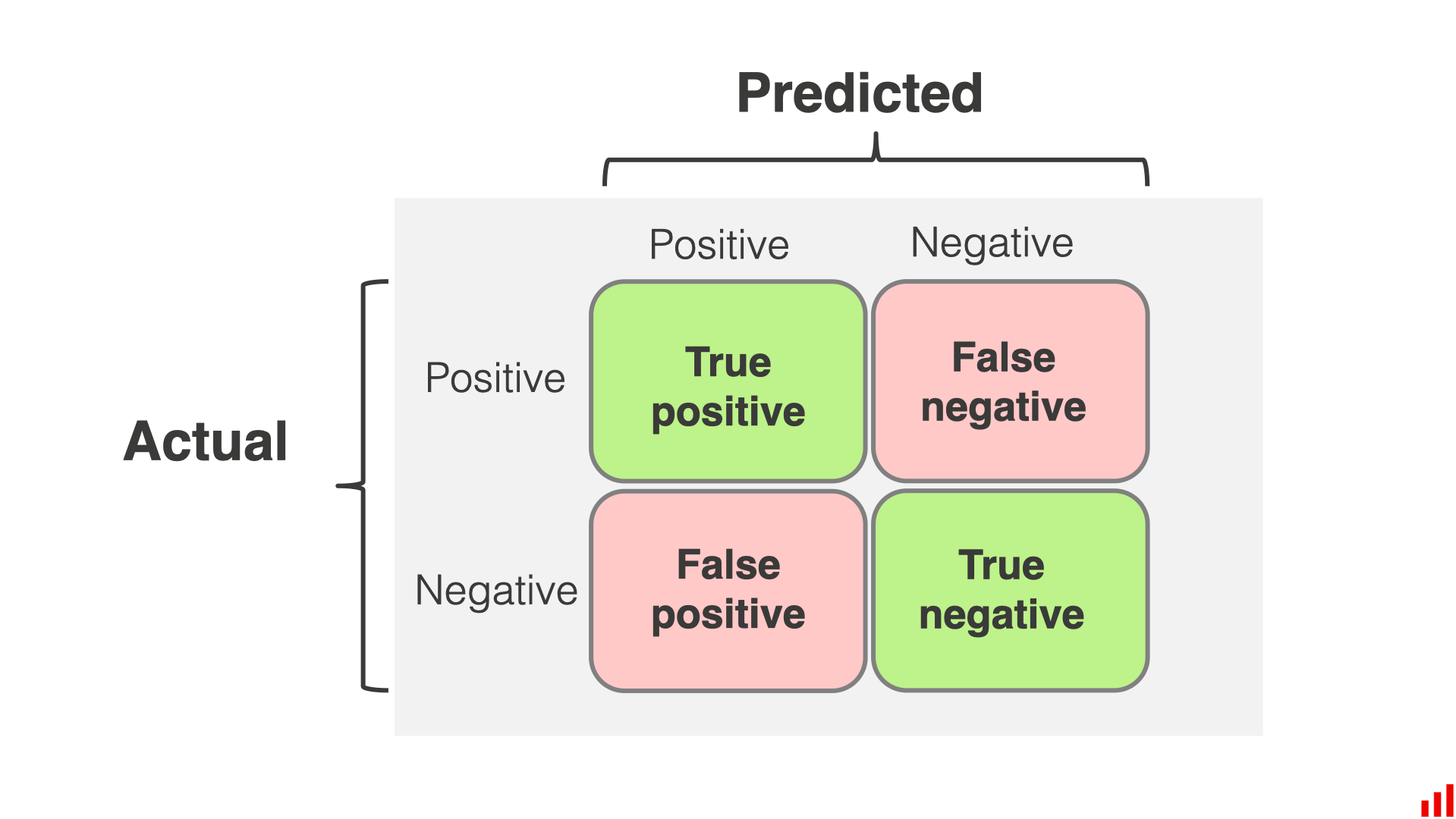
Đánh giá mô hình là một bước quan trọng trong quá trình huấn luyện và triển khai mô hình học máy. Các kỹ thuật và chỉ số đánh giá giúp chúng ta hiểu rõ hơn về hiệu suất của mô hình đối với tập dữ liệu kiểm tra, từ đó đưa ra các cải tiến cần thiết. Hai công cụ phổ biến trong việc đánh giá mô hình là Confusion Matrix và các Metrics đánh giá.

* + 1. Confusion Matrix

Confusion Matrix (Ma trận nhầm lẫn) là một công cụ đánh giá mô hình phân loại, giúp so sánh giữa các dự đoán của mô hình và các giá trị thực tế trong tập kiểm tra. Ma trận này cung cấp cái nhìn tổng quan về hiệu suất của mô hình, từ đó giúp chúng ta nhận diện được các lỗi mà mô hình mắc phải.

Ma trận nhầm lẫn cho một bài toán phân loại nhị phân sẽ có 4 ô:

* True Positive (TP): Số lượng mẫu mà mô hình dự đoán đúng là lớp dương (positive).
* False Positive (FP): Số lượng mẫu mà mô hình dự đoán sai là lớp dương, trong khi thực tế là lớp âm (negative).
* True Negative (TN): Số lượng mẫu mà mô hình dự đoán đúng là lớp âm.
* False Negative (FN): Số lượng mẫu mà mô hình dự đoán sai là lớp âm, trong khi thực tế là lớp dương.



Hình 2.9: Confusion matrix

**Ưu điểm**:

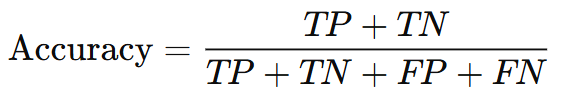
* Giúp phân tích chi tiết các loại lỗi mà mô hình mắc phải (ví dụ: sai thành lớp dương hay lớp âm).
* Cung cấp cái nhìn rõ ràng về hiệu suất của mô hình trong từng lớp.

**Nhược điểm**:

* Không cung cấp thông tin về các lỗi liên quan đến độ chính xác tổng thể của mô hình, mà chỉ tập trung vào các phân loại chính xác và sai lệch giữa các lớp.
  + 1. Metrics đánh giá

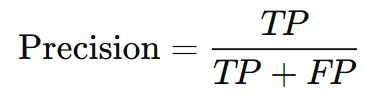
Các **metrics đánh giá** giúp chúng ta đo lường hiệu suất của mô hình từ nhiều góc độ khác nhau. Dưới đây là một số metric quan trọng trong việc đánh giá mô hình phân loại:

1. **Accuracy (Độ chính xác)**:
   * **Định nghĩa**: Là tỷ lệ giữa số mẫu được phân loại đúng trên tổng số mẫu.
   * **Công thức**:



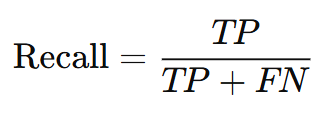
* + **Ưu điểm**: Dễ hiểu và tính toán nhanh.
  + **Nhược điểm**: Không hiệu quả khi dữ liệu mất cân bằng (class imbalance), vì mô hình có thể đạt độ chính xác cao bằng cách dự đoán đúng lớp chiếm ưu thế.

1. **Precision (Độ chính xác)**
   * **Định nghĩa**: Là tỷ lệ giữa số mẫu dự đoán đúng là lớp dương so với tổng số mẫu được mô hình dự đoán là lớp dương.
   * **Công thức**:



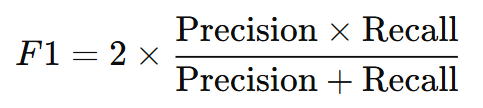
* + **Ưu điểm**: Chỉ ra được độ chính xác của mô hình khi dự đoán lớp dương.
  + **Nhược điểm**: Nếu lớp dương hiếm, precision có thể không phản ánh được khả năng mô hình phân biệt tốt giữa các lớp.

1. **Recall (Độ nhạy)**
   * **Định nghĩa**: Là tỷ lệ giữa số mẫu dự đoán đúng là lớp dương so với tổng số mẫu thực sự thuộc lớp dương.
   * **Công thức**:



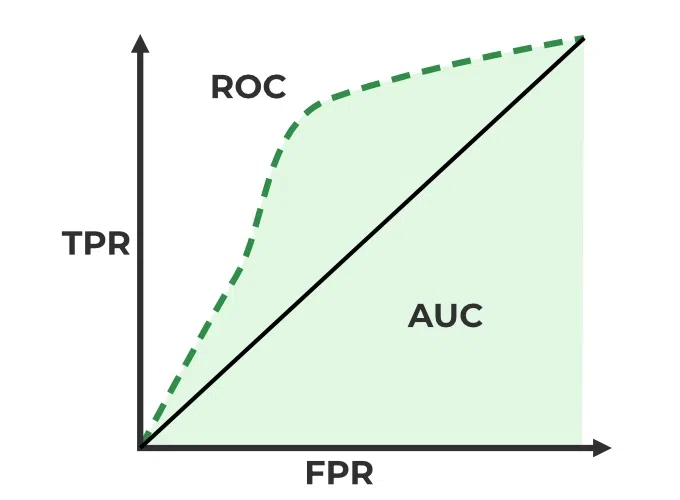
* + **Ưu điểm**: Giúp đánh giá khả năng của mô hình trong việc nhận diện tất cả các mẫu dương.
  + **Nhược điểm**: Một mô hình có recall cao có thể dẫn đến nhiều dự đoán sai (false positives).

1. **F1-Score**
   * **Định nghĩa**: F1-Score là trung bình hài hòa của Precision và Recall. Nó giúp cân bằng giữa độ chính xác và độ nhạy, đặc biệt hữu ích khi dữ liệu bị mất cân bằng.
   * **Công thức**:



* + **Ưu điểm**: Cung cấp cái nhìn cân bằng về độ chính xác và độ nhạy, đặc biệt trong các bài toán phân loại mất cân bằng.
  + **Nhược điểm**: Không phải lúc nào cũng cung cấp thông tin đầy đủ nếu không có sự cân nhắc về các yếu tố khác như Accuracy.

1. **ROC-AUC**



Hình 2.10: ROC-AUC

* + **Định nghĩa**: **Receiver Operating Characteristic (ROC)** là một biểu đồ thể hiện mối quan hệ giữa Tỷ lệ dương giả (False Positive Rate - FPR) và Tỷ lệ dương thật (True Positive Rate - TPR) ở các ngưỡng khác nhau. **AUC (Area Under the Curve)** là diện tích dưới đường cong ROC, đại diện cho khả năng phân biệt của mô hình.
  + **Ưu điểm**: Phản ánh được khả năng phân loại tổng thể của mô hình trên toàn bộ các ngưỡng phân loại.
  + **Nhược điểm**: Cần mô hình có độ phân giải tốt và phụ thuộc vào việc điều chỉnh các ngưỡng.

# GIẢI PHÁP



KẾT LUẬN

**Kết quả đạt được**

**Hạn chế của đề tài**

**Hướng phát triển**

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] |  |
| [2] |  |
| [3] |  |
| [4] |  |
| [5] |  |
| [6] |  |
| [7] |  |

x

x