# Tổng quan

Tài liệu này ghi lại thông tin train và đánh giá kết quả train model

Các kiến thức và khái niệm cần nắm để giải thích được tại sao kết quả và đánh giá model nó lại như vậy.

Cách xử lý và tối ưu model.

[**Tổng quan 1**](#_76wo9x1dfbn)

[Đặc tả kỹ thuật 1](#_a04esaim4oqj)

[Môi trường train 1](#_vkkf6wzev6a9)

[Tập dữ liệu dùng để train 2](#_9zhm2qgesnai)

[Lịch sử train và đánh giá model 3](#_ihkglw3jr0f7)

[FPTAI/vibert-base-cased 3](#_xn5my3dk383z)

[uitnlp/visobert 8](#_cunz9jy0my3)

[vinai/phobert-base 12](#_ifabwxdqk0wl)

[Các kiến thức và khái niệm cần nắm 15](#_v6xz9bmfubi9)

[Rủi ro khi set max\_length 256 15](#_eti11nt3aquw)

[Nếu tăng max\_length lên 512 tokens 16](#_paoq2da5yxjr)

[return\_attention\_mask 16](#_azko4ruyc3mq)

[Learning rate 17](#_jm5a7bvuygoi)

[Batch Size (Kích thước Batch) 18](#_h53do3r6v5gs)

[What’s threshold mean? 22](#_3r3mzhsriz4e)

[Recall (tỉ lệ phát hiện đúng) và Precision (độ chính xác) 24](#_diti763b3a27)

[Loss & epoch 25](#_b4axzqikxxdl)

[Mục Đích của Validation Loss 26](#_hp36cl4jj8f1)

[Nếu bạn không có dữ liệu riêng để kiểm tra validation loss 26](#_gpwcl37ayg3m)

[True Positive (TP) False Positive (FP) True Negative (TN) False Negative (FN) 27](#_ijnv9hxlno23)

[1 0 bình luận xây dựng hoặc không liên quan gì tới TP và FP FN và FP 28](#_59kpuw9f1bd9)

[Phobert-base với epoch 3 thì không có được f1 score. Vì sao mình vẫn chọn? 28](#_5sqw0fy9y87k)

# Đặc tả kỹ thuật

### Môi trường train

<https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz>

### Tập dữ liệu dùng để train

<https://www.kaggle.com/datasets/thngphmnh/voz-comments/data>

* Dữ liệu dạng excel [comments\_data.xlsx](https://docs.google.com/spreadsheets/d/10PBhLCkkMsUOQ-vFGGIKXxyP45E0AOHa/edit#gid=1919373188&range=A1)

Pre-trained model

| No. | Model name | Link |
| --- | --- | --- |
| 1 | vinai/phobert-base | <https://huggingface.co/vinai/phobert-base> |
| 2 | FPTAI/vibert-base-cased | <https://huggingface.co/FPTAI/vibert-base-cased> |
| 3 | uitnlp/visobert | <https://huggingface.co/uitnlp/visobert> |

| **Thông số** | **Value** | **Note** |
| --- | --- | --- |
| Số lượng records | 4954 |  |
| Labels | [1 0] | 1 mang tính xây dựng  0 không |
| batch\_size | 16 |  |
| train\_size | 90% |  |
| val\_size | 10% |  |
| Max\_length | 256 | Giới hạn của 3 model vinai/phobert-base, FPTAI/vibert-base-cased,  uitnlp/visobert là 512 token. [Rủi ro khi set max\_length](#_eti11nt3aquw) |
| Learning rate | lr=5e-5 | 0.00005 [Link](#_jm5a7bvuygoi) |

# Lịch sử train và đánh giá model

### FPTAI/vibert-base-cased

| No. | Revision | Link | Note |
| --- | --- | --- | --- |
| 1 | **FPTAI/vibert-base-cased 1 epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170101269> |  |
| 2 | **FPTAI/vibert-base-cased 2 epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170101301> |  |
| 3 | **FPTAI/vibert-base-cased 3 epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170156232> |  |

1. Revision 1 (**FPTAI/vibert-base-cased 1 epochs**)

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 1 |
| 2 | threshold | 0.5 |

Epoch 1/1: 100%|██████████| 279/279 [1:31:36<00:00, 19.70s/it, loss=0.73]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy | 0.6230 |
| 2 | Precision | 0.5600 |
| 3 | Recall | 0.6452 |
| 4 | F1 Score | 0.5996 |

Evaluating: 100%|██████████| 31/31 [03:07<00:00, 6.05s/it]

Accuracy: 0.6230

Precision: 0.5600

Recall: 0.6452

F1 Score: 0.5996

**Accuracy (Độ chính xác)**: 62.3% - Phần trăm tổng số dự đoán chính xác so với tổng số mẫu. Độ chính xác này cho thấy mô hình có khả năng dự đoán đúng khoảng 62.3% trường hợp.

**Precision (Độ chính xác của dự đoán tích cực)**: 56.0% - Khi model dự đoán một mẫu là tích cực, có 56% cơ hội là dự đoán đó chính xác. Độ chính xác này tương đối thấp, điều này có nghĩa là mô hình đang phân loại một số mẫu không phải tích cực thành tích cực.

**Recall (Độ nhạy)**: 64.52% - Tỷ lệ phần trăm các mẫu tích cực thực sự được mô hình dự đoán chính xác so với tổng số mẫu tích cực thực tế. Điều này cho thấy mô hình có khả năng "bắt" được khoảng 64.52% các trường hợp tích cực. Tuy nhiên, vẫn còn một số trường hợp tích cực mà mô hình không nhận diện được.

**F1 Score**: 59.96% - Là giá trị trung bình điều hòa của Precision và Recall. F1 Score cung cấp một cái nhìn toàn diện hơn về hiệu suất của mô hình, đặc biệt là khi có sự mất cân bằng giữa các lớp. Một F1 Score cao chỉ ra rằng cả hai chỉ số Precision và Recall đều cao. Trong trường hợp này, F1 Score cho thấy mô hình này có một hiệu suất khá cân bằng giữa việc không bỏ sót các trường hợp tích cực và giảm thiểu số lượng dự đoán sai.

1. Revision 2 (**FPTAI/vibert-base-cased 2 epochs**)

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 2 |
| 2 | threshold | 0.5 |

Epoch 1/2: 100%|██████████| 279/279 [1:45:52<00:00, 22.77s/it, loss=0.794]

Epoch 2/2: 100%|██████████| 279/279 [1:45:49<00:00, 22.76s/it, loss=0.508]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy |  |
| 2 | Precision |  |
| 3 | Recall |  |
| 4 | F1 Score |  |

Evaluating: 100%|██████████| 31/31 [03:30<00:00, 6.79s/it]

Accuracy: 0.6371

Precision: 0.5595

Recall: 0.4700

F1 Score: 0.5109

**Loss giảm**: Từ epoch thứ nhất sang epoch thứ hai, loss giảm từ 0.794 xuống 0.508, đây là một dấu hiệu tốt cho thấy mô hình đang học và cải thiện qua từng bước lặp.

**Accuracy tăng nhẹ**: Độ chính xác (Accuracy) tăng từ 0.6230 lên 0.6371. Mặc dù đây là một cải thiện, nhưng sự tăng trưởng không đáng kể, cho thấy rằng mô hình có thể cần thêm thời gian hoặc cần những điều chỉnh về kiến trúc hoặc dữ liệu để cải thiện đáng kể.

**Precision và Recall**: Precision giảm nhẹ từ 0.5600 xuống 0.5595, trong khi Recall giảm khá nhiều từ 0.6452 xuống 0.4700. Điều này cho thấy mô hình của bạn đã trở nên ít "tự tin" hơn trong việc dự đoán các trường hợp tích cực thực sự, dẫn đến việc bỏ lỡ một số trường hợp tích cực đúng (false negatives) và tăng số lượng dự đoán tích cực sai (false positives).

**F1 Score giảm**: F1 Score giảm từ 0.5996 xuống 0.5109, phản ánh sự giảm về cả precision và recall. F1 Score là một chỉ số quan trọng vì nó cung cấp một cái nhìn cân bằng giữa precision và recall, đặc biệt hữu ích khi phân phối của các lớp không đồng đều.

Đánh giá:

Mô hình của bạn đã thấy một số tiến triển, nhưng rõ ràng là vẫn còn không ít thách thức. Dưới đây là một số gợi ý để cải thiện mô hình:

Thử nghiệm với số epochs lớn hơn: Đôi khi, mô hình cần nhiều vòng lặp hơn để học sâu hơn từ dữ liệu.

Kiểm tra lại dữ liệu: Đảm bảo dữ liệu được làm sạch và tiền xử lý đúng cách. Thêm vào đó, việc tăng cường dữ liệu (data augmentation) có thể giúp mô hình học được những đặc điểm phức tạp hơn.

Chỉnh sửa kiến trúc mô hình: Thử nghiệm với các kiến trúc mô hình khác nhau hoặc điều chỉnh các tham số hiện tại để xem liệu có cải thiện được hiệu suất hay không.

Fine-tuning hyperparameters: Thực hiện tìm kiếm hyperparameter để tìm ra bộ giá trị tối ưu cho learning rate, kích thước batch, v.v.

Cross-validation: Sử dụng kỹ thuật cross-validation để đảm bảo mô hình không bị overfitting và có khả năng tổng quát hóa tốt trên dữ liệu chưa thấy.

1. Revision 3 (**FPTAI/vibert-base-cased 3 epochs**)

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 3 |
| 2 | threshold | 0.5 |

Epoch 1/3: 100%|██████████| 279/279 [1:36:15<00:00, 20.70s/it, loss=0.656]

Epoch 2/3: 100%|██████████| 279/279 [1:35:46<00:00, 20.60s/it, loss=0.58]

Epoch 3/3: 100%|██████████| 279/279 [1:35:46<00:00, 20.60s/it, loss=0.469]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy | 0.6452 |
| 2 | Precision | 0.5874 |
| 3 | Recall | 0.6093 |
| 4 | F1 Score | 0.5982 |

Evaluating: 100%|██████████| 31/31 [03:09<00:00, 6.10s/it]

Accuracy: 0.6452

Precision: 0.5874

Recall: 0.6093

F1 Score: 0.5982

Giảm Loss: Loss giảm liên tục qua từng epoch, từ 0.656 xuống 0.469, cho thấy mô hình đang học hiệu quả từ dữ liệu và cải thiện khả năng dự đoán của mình.

Tăng Accuracy: Độ chính xác của mô hình đã tăng lên 0.6452, một cải thiện nhỏ nhưng đáng kể so với lần đánh giá trước. Điều này cho thấy mô hình của bạn đang trở nên tốt hơn trong việc phân loại đúng các mẫu dữ liệu.

Tăng Precision và Recall: Precision tăng từ 0.5595 lên 0.5874, và Recall tăng từ 0.4700 lên 0.6093. Cả hai chỉ số này đều cho thấy mô hình của bạn không chỉ dự đoán chính xác hơn mà còn giảm thiểu được số lượng dự đoán sai (cả dự đoán tích cực sai và bỏ sót dự đoán tích cực).

Tăng F1 Score: F1 Score tăng từ 0.5109 lên 0.5982, phản ánh sự cải thiện đáng kể về cả precision và recall. F1 Score là một chỉ số quan trọng khi bạn cần cân bằng giữa precision và recall, đặc biệt trong trường hợp dữ liệu không cân xứng giữa các lớp.

Đánh giá:

Kết quả sau ba epochs cho thấy mô hình của bạn đang trên đường đi đúng và cải thiện đều đặn qua mỗi lần lặp. Đây là những dấu hiệu tích cực cho thấy việc điều chỉnh và tối ưu hóa mô hình đang mang lại kết quả. Để tiếp tục cải thiện mô hình, bạn có thể:

Tiếp tục huấn luyện thêm: Nếu có dấu hiệu mô hình chưa bắt đầu overfitting, việc tiếp tục huấn luyện có thể giúp mô hình học sâu hơn và cải thiện thêm.

Thử nghiệm kiến trúc mới: Nếu mô hình bắt đầu bão hòa và không cải thiện nhiều qua các lần lặp, bạn có thể xem xét thử nghiệm với các kiến trúc mô hình khác.

Tinh chỉnh Hyperparameters: Điều chỉnh các hyperparameters như learning rate, kích thước batch, hoặc sử dụng các kỹ thuật regularization có thể giúp mô hình đạt được hiệu suất tốt hơn.

Augmentation và Dữ liệu thêm: Xem xét việc tăng cường dữ liệu hoặc thu thập thêm dữ liệu để giúp mô hình học được những mẫu phức tạp hơn và cải thiện khả năng tổng quát hóa.

### uitnlp/visobert

| No. | Revision | Link | Note |
| --- | --- | --- | --- |
| 1 | **uitnlp/visobert 1 epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170102312> |  |
| 2 | **Visobert 2epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170128567> |  |
| 3 | **Visobert 3 epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170128717> |  |

1. Revision 1 (**uitnlp/visobert 1 epochs**)

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 1 |
| 2 | threshold | 0.5 |

Epoch 1/1: 100%|██████████| 279/279 [1:36:49<00:00, 20.82s/it, loss=0.815]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy |  |
| 2 | Precision |  |
| 3 | Recall |  |
| 4 | F1 Score |  |

Evaluating: 100%|██████████| 31/31 [03:18<00:00, 6.41s/it]

Accuracy: 0.6371

Precision: 0.7333

Recall: 0.2115

F1 Score: 0.3284

Loss giảm đáng kể, điều này cho thấy mô hình đã bắt đầu học từ dữ liệu.

Tuy nhiên, Recall thấp chỉ ra rằng mô hình có khả năng bỏ sót một số trường hợp tích cực thực sự.

F1 Score thấp phản ánh việc mô hình không cân bằng giữa Precision và Recall.

1. Revision 2 (**Visobert 2epochs**)

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 2 |
| 2 | threshold | 0.5 |

Epoch 1/2: 100%|██████████| 279/279 [1:14:43<00:00, 16.07s/it, loss=0.513]

Epoch 2/2: 100%|██████████| 279/279 [1:15:03<00:00, 16.14s/it, loss=0.685]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy |  |
| 2 | Precision |  |
| 3 | Recall |  |
| 4 | F1 Score |  |

Evaluating: 100%|██████████| 31/31 [02:35<00:00, 5.03s/it]

Accuracy: 0.6976

Precision: 0.6667

Recall: 0.6486

F1 Score: 0.6575

Cải thiện đáng kể về tất cả các chỉ số, bao gồm Accuracy, Precision, Recall và F1 Score. Điều này cho thấy thêm một epoch huấn luyện đã giúp mô hình học được nhiều hơn từ dữ liệu.

Mặc dù có sự tăng đáng chú ý, nhưng cần lưu ý sự tăng của Loss trong epoch thứ hai. Điều này có thể là dấu hiệu của việc overfitting, nơi mô hình bắt đầu học quá mức từ dữ liệu huấn luyện mà không khái quát hóa tốt trên dữ liệu mới.

1. Revision 3 (**Visobert 3 epochs**)

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 3 |
| 2 | threshold | 0.5 |

Epoch 1/3: 100%|██████████| 279/279 [1:43:41<00:00, 22.30s/it, loss=0.535]

Epoch 2/3: 100%|██████████| 279/279 [1:44:06<00:00, 22.39s/it, loss=0.224]

Epoch 3/3: 100%|██████████| 279/279 [1:44:55<00:00, 22.56s/it, loss=0.515]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy |  |
| 2 | Precision |  |
| 3 | Recall |  |
| 4 | F1 Score |  |

Evaluating: 100%|██████████| 31/31 [03:30<00:00, 6.78s/it]

Accuracy: 0.6815

Precision: 0.6034

Recall: 0.6796

F1 Score: 0.6393

Accuracy giảm nhẹ so với sau 2 epochs, nhưng Recall tăng, điều này có nghĩa là mô hình giờ đây có khả năng phát hiện một tỷ lệ cao hơn các trường hợp tích cực thực sự.

Precision giảm, dẫn đến việc mô hình dự đoán nhiều kết quả tích cực sai hơn.

F1 Score, dù cao hơn sau 1 epoch, nhưng thấp hơn so với sau 2 epochs, cho thấy sự cân bằng giữa Precision và Recall không được tốt như sau 2 epochs.

Kết luận và hướng cải thiện tiếp theo:

* Mô hình có xu hướng cải thiện với số lượng epoch tăng lên, nhưng cũng có dấu hiệu của overfitting và mất cân bằng giữa Precision và Recall.
* Cân nhắc sử dụng early stopping hoặc regularization để tránh overfitting nếu tiếp tục tăng số lượng epochs.
* Đánh giá việc sử dụng một learning rate scheduler có thể giúp điều chỉnh learning rate trong quá trình huấn luyện để cải thiện hiệu suất của mô hình.

### vinai/phobert-base

| No. | Revision | Link | Note |
| --- | --- | --- | --- |
| 1 | **vinai/phobert-base 1 epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170197205> |  |
| 2 | **vinai/phobert-base 2 epoch** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170198072> |  |
| 3 | **vinai/phobert-base 1 epochs** | <https://www.kaggle.com/code/thngphmnh/phantichdulieuvoz?scriptVersionId=170198108> |  |

1. Revision 1

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 1 |
| 2 | threshold | 0.3 |

Epoch 1/1: 100%|██████████| 279/279 [1:30:55<00:00, 19.55s/it, loss=0.705]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy |  |
| 2 | Precision |  |
| 3 | Recall |  |
| 4 | F1 Score |  |

Evaluating: 100%|██████████| 31/31 [03:00<00:00, 5.83s/it]

Accuracy: 0.4476

Precision: 0.4258

Recall: 0.9466

F1 Score: 0.5873

**Threshold Tuning**: The choice of threshold **(0.3)** had a clear impact, resulting in some predictions being classified as positive (1). This is a good direction, but you might want to experiment further with this threshold. Adjusting it can help find a better balance between precision and recall, depending on what's more important for your application.

1. Revision 2

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 1 |
| 2 | threshold | 0.3 |

Epoch 1/2: 100%|██████████| 279/279 [1:37:57<00:00, 21.07s/it, loss=0.642]

Epoch 2/2: 100%|██████████| 279/279 [1:38:11<00:00, 21.12s/it, loss=0.478]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy |  |
| 2 | Precision |  |
| 3 | Recall |  |
| 4 | F1 Score |  |

Evaluating: 100%|██████████| 31/31 [03:16<00:00, 6.34s/it]

Accuracy: 0.6411

Precision: 0.5640

Recall: 0.5805

F1 Score: 0.5721

1. Revision 3

**Params training model**

| **No.** | **Hyperparams** | **Value** |
| --- | --- | --- |
| 1 | epoch | 1 |
| 2 | threshold | 0.3 |

Epoch 1/3: 100%|██████████| 279/279 [1:45:24<00:00, 22.67s/it, loss=0.677]

Epoch 2/3: 100%|██████████| 279/279 [1:44:50<00:00, 22.55s/it, loss=0.728]

Epoch 3/3: 100%|██████████| 279/279 [1:44:46<00:00, 22.53s/it, loss=0.549]

**Evaluation Results**

| **No.** | **Metrics** | **Value** |
| --- | --- | --- |
| 1 | Accuracy |  |
| 2 | Precision |  |
| 3 | Recall |  |
| 4 | F1 Score |  |

Evaluating: 100%|██████████| 31/31 [03:31<00:00, 6.83s/it]

Accuracy: 0.5827

Precision: 0.0000

Recall: 0.0000

F1 Score: 0.0000

# Các kiến thức và khái niệm cần nắm

#### Rủi ro khi set max\_length 256

Ví dụ: comments

Cái biển a nói thì nó dành cho đường chính nhưng vị trí cắm hơi ngu nó phải cắm xa cái đường nhánh ra 1 đoạn để người đi ra từ đường nhánh còn nhìn được. Nhưng có lẽ làm thế thì lại quá gần ngã rẽ sắp tới. Cộng thêm cái việc thiếu mũi tên chỉ hướng nữa. Thường thì trong SG tôi thấy đường 4 làn trở đi đều có giá long môn. Ko hiểu sao đường này ko có. Nói chung là do bên giao thông cắm biển kẻ vạch như shit. Nhưng nếu là tôi thì thấy cái biển chỉ làn kia cũng đi làn bên trái cho chắc cú đỡ rắc rối. Đi làn khác cho dù ko sai luật nhưng conan vịn vào cũng mất thời gian tranh cãi. Ko đáng để mạo hiểm.

Với bình luận trên, nếu nó vượt quá 256 tokens sau khi token hóa, các phần cuối cùng của bình luận sẽ bị cắt bỏ. Điều này bao gồm thông tin liên quan đến nhận xét về việc điều chỉnh lái vào làn đúng để tránh rắc rối, và ý kiến về việc thực hiện các quyết định lái xe an toàn và hợp lý dựa trên tình hình giao thông và biển báo. Cụ thể, các phần nói về "đi làn khác cho dù ko sai luật nhưng conan vịn vào cũng mất thời gian tranh cãi. Ko đáng để mạo hiểm" có thể bị ảnh hưởng.

##### Nếu tăng max\_length lên 512 tokens

1. Tăng Yêu Cầu Bộ Nhớ:

Khi tăng max\_length, số lượng tokens được xử lý trong mỗi đầu vào sẽ tăng lên, điều này yêu cầu bộ nhớ (RAM và VRAM) nhiều hơn để lưu trữ và xử lý dữ liệu. Điều này có thể dẫn đến việc bạn cần phải giảm kích thước batch để tránh vượt quá bộ nhớ của GPU, ảnh hưởng đến hiệu quả và tốc độ huấn luyện.

2. Tăng Thời Gian Xử Lý:

Với max\_length lớn hơn, mỗi bước tính toán trong mô hình sẽ cần thêm thời gian để xử lý số lượng tokens lớn hơn. Điều này có thể làm tăng đáng kể thời gian cần thiết cho cả việc huấn luyện và suy luận, đặc biệt nếu bạn làm việc với một lượng lớn dữ liệu.

3. Có Thể Cải Thiện Độ Chính Xác:

Đối với các văn bản dài, việc tăng max\_length cho phép mô hình xem xét một phạm vi rộng lớn hơn của bối cảnh, có thể cải thiện hiểu biết và xử lý của mô hình đối với nội dung đầu vào. Điều này có thể dẫn đến việc cải thiện độ chính xác trong một số tác vụ, như phân loại văn bản, trả lời câu hỏi, hoặc dịch máy, nơi mà bối cảnh đầy đủ của đoạn văn bản là quan trọng.

4. Tăng Nguy Cơ Overfitting:

Trong một số trường hợp, việc sử dụng độ dài đầu vào lớn hơn có thể dẫn đến nguy cơ overfitting, đặc biệt nếu tập dữ liệu huấn luyện của bạn không đủ đa dạng hoặc không đủ lớn để mô hình hóa hiệu quả toàn bộ phạm vi của dữ liệu đầu vào có thể gặp phải.

5. Khả Năng Tương Thích:

Một số mô hình có thể không được thiết kế để xử lý max\_length tối đa 512 tokens một cách hiệu quả, hoặc có thể yêu cầu cấu hình hoặc tối ưu hóa bổ sung để làm việc với độ dài đầu vào lớn như vậy mà không gặp vấn đề về hiệu suất hoặc bộ nhớ.

Kết luận, việc tăng max\_length có thể mang lại lợi ích nhất định trong việc hiểu và xử lý văn bản, nhưng cũng đi kèm với những thách thức liên quan đến bộ nhớ và hiệu suất.

=> để 256 max\_length để train nhanh tiết kiệm time và bộ nhớ.

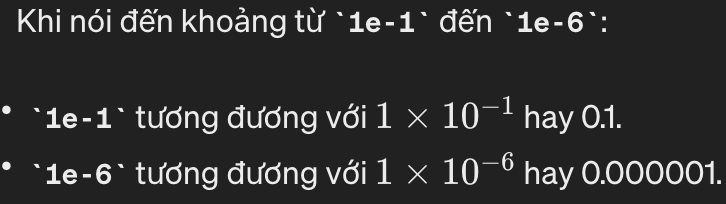
##### return\_attention\_mask

Mặc dù không phải là một hyperparameter, nhưng việc sử dụng attention mask là quan trọng để chỉ ra cho mô hình biết đâu là phần thực sự của dữ liệu và đâu là phần được thêm vào (padding). Điều này giúp mô hình tập trung vào thông tin có ý nghĩa trong quá trình học.

##### Learning rate

| # Prepare optimizer  optimizer = AdamW(model.parameters(), lr=5e-5) |
| --- |

Learning rate (lr) có ảnh hưởng đáng kể đến tốc độ và khả năng hội tụ của quá trình học trong mô hình học sâu. Dưới đây là cách thay đổi learning rate có thể ảnh hưởng:



1. Learning Rate Cao 

Tốc độ học nhanh: Mô hình có thể học nhanh hơn do làm cho những bước lớn trong không gian tham số mỗi lần cập nhật.

Rủi ro mất mát hội tụ: Nếu quá cao, learning rate có thể gây ra hiện tượng 'bouncing' quanh điểm tối ưu, hoặc thậm chí làm cho mô hình diverge và không bao giờ hội tụ.

2. Learning Rate Trung Bình 

Cân bằng giữa tốc độ học và ổn định: Đây thường là khoảng tối ưu cho nhiều tác vụ, nơi mô hình học ổn định và với tốc độ phù hợp, giảm thiểu nguy cơ bỏ qua các điểm tối ưu.

Phổ biến trong thực tế: Nhiều nghiên cứu và ứng dụng thực tế tìm thấy learning rate trong khoảng này thường mang lại hiệu suất tốt.

3. Learning Rate Thấp **(1e âm sáu tới năm E âm năm)**

Học chậm: Mô hình sẽ cập nhật tham số rất chậm, điều này có thể làm tăng đáng kể thời gian cần thiết để hội tụ.

Tối ưu hóa kỹ lưỡng: Lợi ích của một learning rate thấp là cho phép mô hình 'tinh chỉnh' các tham số một cách cẩn trọng, giúp tiếp cận gần với điểm tối ưu hơn mà không bị bỏ qua.

Ảnh hưởng của việc Thay Đổi Learning Rate:

Thay đổi learning rate có thể giúp mô hình hội tụ nhanh hơn hoặc chậm lại, tùy thuộc vào việc tăng hay giảm giá trị này.

Cải thiện hoặc làm giảm hiệu suất mô hình: Một learning rate không phù hợp có thể gây ra hiện tượng overfitting (quá khớp) hoặc underfitting (không khớp đủ).

Đòi hỏi thử nghiệm thực nghiệm: Tìm ra learning rate tối ưu thường đòi hỏi nhiều vòng lặp thử nghiệm và điều chỉnh dựa trên kết quả thực tế.

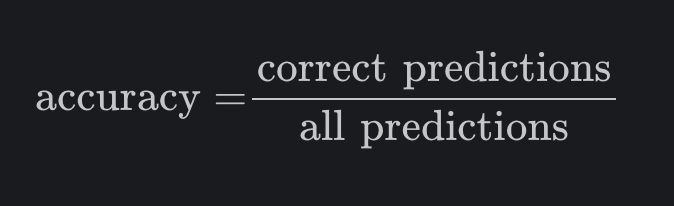
Vì vậy, việc chọn lựa và tinh chỉnh learning rate phù hợp là một bước quan trọng và cần thiết trong quá trình xây dựng và tối ưu hóa mô hình học sâu.

##### Batch Size (Kích thước Batch)

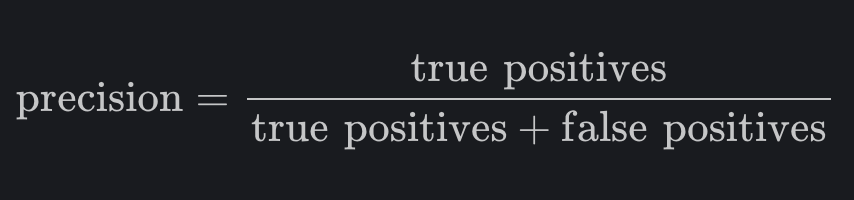
Ý nghĩa: Kích thước batch ảnh hưởng đến quá trình học của mô hình về mặt ổn định và hiệu quả tính toán. Batch nhỏ giúp cập nhật trọng số thường xuyên hơn với chi phí tính toán cao và ổn định kém, trong khi batch lớn làm giảm chi phí tính toán nhưng có thể dẫn đến việc mô hình khó tìm được điểm tối ưu.

Cách điều chỉnh: Bắt đầu với kích thước nhỏ (ví dụ, 32 hoặc 64) và tăng dần đến khi bạn thấy hiệu suất không cải thiện hoặc giảm. Lưu ý vấn đề về bộ nhớ khi tăng kích thước batch.

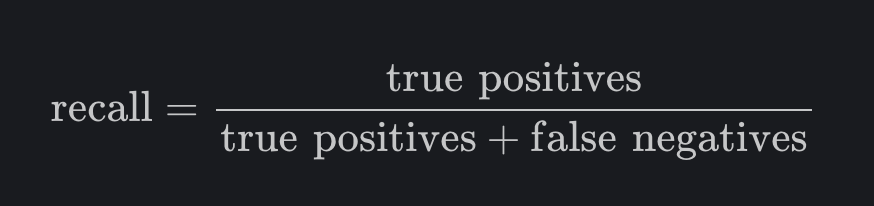
Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.



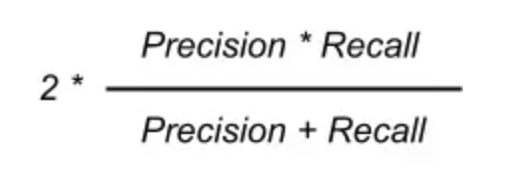
Precision is defined as the fraction of relevant examples (true positives) among all of the examples which were predicted to belong in a certain class.

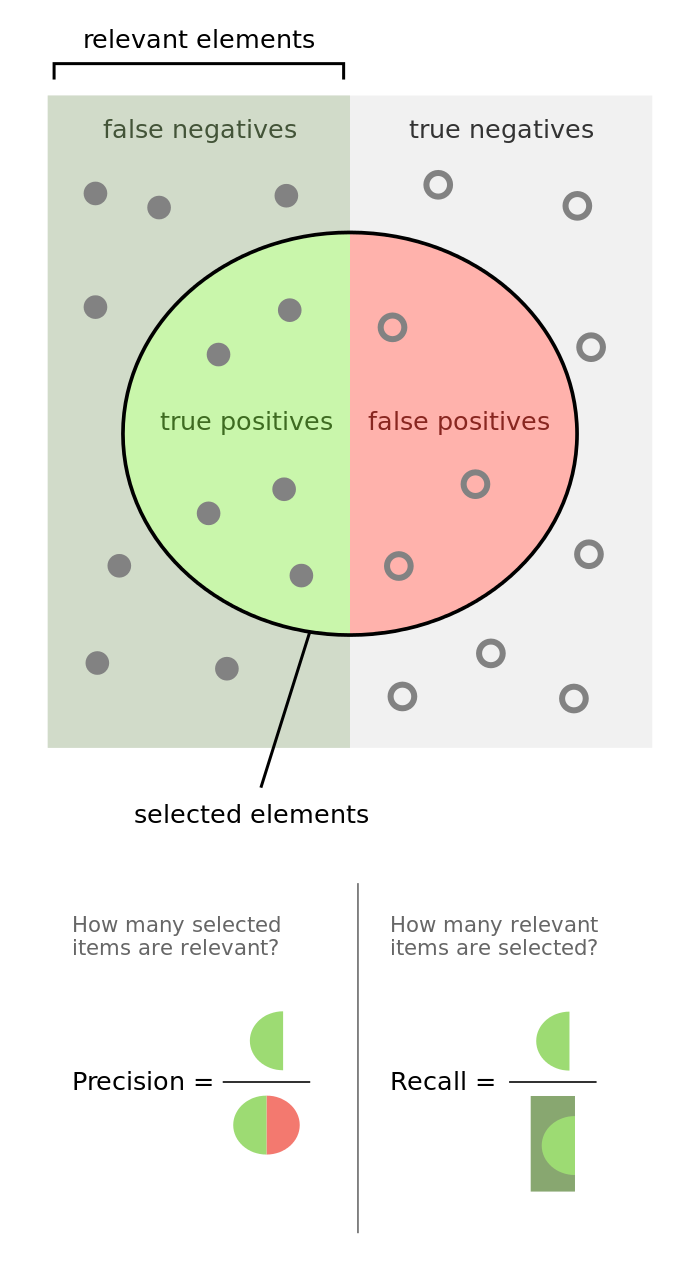


Recall is defined as the fraction of examples which were predicted to belong to a class with respect to all of the examples that truly belong in the class.



F1 score combines both of these precision and recall and symmetrically represents them in the formula:





Evaluating ML model

https://www.jeremyjordan.me/evaluating-a-machine-learning-model/

F1-score

https://arize.com/blog-course/f1-score/#:~:text=F1%20score%20computes%20the%20average,of%20the%20predictions%20were%20correct.

##### What’s threshold mean?

The concept of a threshold in the context of machine learning, especially in classification tasks, plays a critical role in determining how the raw output of a model (often in the form of probabilities) is translated into class labels.

When a model predicts probabilities, each number represents the model's confidence that a given observation belongs to a particular class. For a binary classification task (where there are two possible classes), each output probability typically corresponds to the likelihood of the positive class, with the remainder (1 minus the probability) indicating the likelihood of the negative class.

A threshold is a value that determines the cut-off point for deciding between classes. Here's a breakdown:

* Probability Threshold: This is most commonly used in binary classification tasks. For each prediction, if the probability of the positive class is greater than or equal to the threshold, the prediction is classified as positive. Otherwise, it's classified as negative. By default, many models use a threshold of 0.5. However, this can be adjusted.
* Adjusting the Threshold:
  + Lowering the Threshold: Means that you require less confidence to predict the positive class. This can increase the recall (the proportion of actual positives that were correctly identified) but may also increase the false positives, potentially lowering precision (the proportion of positive identifications that were actually correct).
  + Raising the Threshold: Requires more confidence to label an observation as positive, potentially increasing precision but at the risk of missing actual positives, thus lowering recall.

The choice of threshold affects the trade-off between precision and recall and can be adjusted based on what's more important for your specific application. For instance, in medical diagnostics, a high recall might be preferred to ensure as few positive cases are missed as possible, even if it means more false positives that need further testing. In contrast, for a spam email filter, high precision might be more desirable to avoid filtering out legitimate emails, even if some spam emails are missed.

Selecting the optimal threshold often involves analyzing precision-recall curves, considering the costs of false positives and false negatives, and understanding the specific needs and constraints of your application.

Threshold tuning, hay điều chỉnh ngưỡng, là một quá trình trong xử lý ngôn ngữ tự nhiên (NLP) và học máy, đặc biệt là trong các bài toán phân loại. Mục tiêu là xác định một ngưỡng giá trị nào đó để phân biệt giữa các lớp hoặc nhãn. Ngưỡng này thường được áp dụng cho đầu ra của một mô hình học máy, chẳng hạn như xác suất dự đoán một lớp cụ thể.

Về mặc định, ngưỡng thường được đặt ở 0.5 trong bài toán phân loại nhị phân. Điều này có nghĩa là:

Nếu xác suất dự đoán một mẫu thuộc về lớp tích cực (hoặc "1") lớn hơn hoặc bằng 0.5, mẫu đó được phân loại vào lớp tích cực.

Nếu xác suất nhỏ hơn 0.5, mẫu được phân loại vào lớp tiêu cực (hoặc "0"). Khi bạn điều chỉnh ngưỡng xuống 0.3, điều này có nghĩa là:

Một mẫu chỉ cần có xác suất từ 0.3 trở lên để được xem xét thuộc về lớp tích cực. Điều này làm giảm "yêu cầu" để một mẫu được xếp vào lớp tích cực.

Ảnh hưởng của việc điều chỉnh ngưỡng như sau:

**Tăng Tỉ lệ Phát Hiện Đúng (True Positive Rate hoặc Sensitivity)**: Khi giảm ngưỡng, mô hình có khả năng phân loại nhiều mẫu hơn vào lớp tích cực, điều này có thể tăng số lượng True Positives (các trường hợp mô hình dự đoán đúng là tích cực).

**Tăng False Positive Rate (FPR)**: Đồng thời, giảm ngưỡng cũng có nghĩa là sẽ có nhiều trường hợp False Positives (các trường hợp mô hình sai khi phân loại một mẫu tiêu cực thành tích cực) hơn, vì một số mẫu tiêu cực với xác suất dưới 0.5 nhưng trên 0.3 sẽ được phân loại nhầm.

**Giảm Precision**: Precision (độ chính xác) là tỷ lệ của số mẫu dự đoán đúng là tích cực so với tổng số mẫu được dự đoán là tích cực (đúng và sai). Do số lượng False Positives tăng lên, precision có thể giảm.

**Có thể ảnh hưởng đến Accuracy**: Độ chính xác tổng thể của mô hình cũng có thể bị ảnh hưởng tùy thuộc vào cấu trúc dữ liệu cụ thể và mức độ điều chỉnh của ngưỡng.

**Trade-off giữa Precision và Recall**: Việc giảm ngưỡng thường dẫn đến trade-off giữa precision và recall (sensitivity). Khi bạn cố gắng tăng recall bằng cách giảm ngưỡng, bạn thường sẽ phải chấp nhận một mức độ giảm precision.

Tùy thuộc vào mục tiêu cụ thể của bài toán và chi phí liên quan đến các loại lỗi khác nhau (false positives so với false negatives), việc điều chỉnh ngưỡng có thể giúp cải thiện hiệu suất mô hình theo cách phù hợp với yêu cầu đó.

##### Recall (tỉ lệ phát hiện đúng) và Precision (độ chính xác)

Là hai chỉ số quan trọng trong đánh giá hiệu suất của các mô hình phân loại trong học máy và xử lý ngôn ngữ tự nhiên (NLP). Cả hai đều đóng vai trò quan trọng nhưng thường có mối quan hệ đối lập, nghĩa là cải thiện một chỉ số có thể làm giảm chỉ số kia. Dưới đây là định nghĩa cụ thể của hai chỉ số này:

Recall (Tỉ lệ phát hiện đúng): Là tỷ lệ giữa số lượng mẫu tích cực được mô hình phân loại đúng (True Positives) so với tổng số mẫu thực sự tích cực trong dữ liệu (tổng của True Positives và False Negatives). Recall cao cho thấy rằng mô hình có khả năng phát hiện tốt các mẫu tích cực.

Precision (Độ chính xác): Là tỷ lệ giữa số lượng mẫu tích cực được mô hình phân loại đúng so với tổng số mẫu được mô hình dự đoán là tích cực (tổng của True Positives và False Positives). Precision cao cho thấy rằng khi mô hình dự đoán một mẫu là tích cực, có khả năng cao là dự đoán đó chính xác.

Vì sao lại muốn tăng Recall?

Tăng Recall đôi khi được ưu tiên trong các tình huống mà việc bỏ sót mẫu tích cực (False Negatives) có hậu quả nghiêm trọng hơn so với việc phân loại nhầm mẫu tiêu cực thành tích cực (False Positives). Điều này phụ thuộc vào bối cảnh cụ thể của vấn đề. Dưới đây là một số lý do tại sao việc tăng Recall có thể được ưu tiên:

Trong y tế: Việc phát hiện các trường hợp mắc bệnh (ví dụ: ung thư) quan trọng hơn việc đảm bảo mọi dự đoán đều chính xác. Một False Negative (bỏ sót một trường hợp bệnh) có thể có hậu quả nghiêm trọng về sức khỏe, trong khi một False Positive (cảnh báo nhầm) có thể chỉ dẫn đến các xét nghiệm tiếp theo để xác minh.

Trong phát hiện gian lận: Việc phát hiện mọi giao dịch gian lận là quan trọng để bảo vệ tài sản, ngay cả khi điều đó có nghĩa là phải xem xét thêm một số giao dịch hợp pháp được dự đoán nhầm là gian lận.

Trong an ninh và giám sát: Trong các hệ thống giám sát an ninh, việc bỏ sót một mối đe dọa có thể dẫn đến hậu quả nghiêm trọng, do đó việc tăng Recall để đảm bảo hầu hết các mối đe dọa đều được phát hiện là rất quan trọng, ngay cả khi điều này có nghĩa là phải xử lý một số báo động giả.

Tuy nhiên, việc tăng Recall không phải không có giá của nó. Điều này thường dẫn đến việc tăng số lượng False Positives, có thể làm tăng chi phí và công sức cho việc xác minh các dự đoán. Do đó, việc tìm kiếm một cân bằng tối ưu giữa Recall và Precision là quan trọng, tùy thuộc vào mục tiêu cụ thể và bối cảnh ứng dụng của mô hình.

##### Loss & epoch

Mối tương quan giữa loss và epoch trong quá trình huấn luyện một mô hình học máy là một chủ đề quan trọng và thường được nghiên cứu kỹ lưỡng. Dưới đây là một số điểm chính liên quan đến mối quan hệ giữa hai khái niệm này:

**Giảm Loss qua từng Epoch**: Một trong những mục tiêu chính của quá trình huấn luyện là giảm giá trị loss, hay sai số, của mô hình trên dữ liệu huấn luyện. Khi một epoch hoàn thành, mô hình đã được cập nhật để phản ánh toàn bộ dữ liệu huấn luyện một lần. Với mỗi epoch tiếp theo, mô hình có cơ hội "học" thêm từ dữ liệu, dẫn đến việc giảm loss.

**Hội Tụ**: Lý tưởng nhất là, loss sẽ giảm dần qua mỗi epoch cho đến khi đạt đến một điểm hội tụ, nơi mà giá trị loss không còn giảm đáng kể qua các epoch tiếp theo. Điểm hội tụ này đại diện cho việc mô hình đã học xong từ dữ liệu huấn luyện đến mức có thể.

**Overfitting**: Nếu loss trên dữ liệu huấn luyện tiếp tục giảm, nhưng loss trên dữ liệu kiểm thử (validation loss) bắt đầu tăng, đây có thể là dấu hiệu của overfitting. Điều này xảy ra khi mô hình "học" quá mức từ dữ liệu huấn luyện, bao gồm cả nhiễu, và mất khả năng tổng quát hóa trên dữ liệu mới.

**Underfitting**: Ngược lại, nếu loss vẫn còn cao sau nhiều epoch, đây có thể là dấu hiệu của underfitting - mô hình không đủ phức tạp để học từ dữ liệu hoặc không được huấn luyện đủ lâu.

**Điều Chỉnh Hyperparameters**: Quan sát sự thay đổi của loss qua các epoch có thể giúp trong việc điều chỉnh các hyperparameters, như tốc độ học (learning rate), để cải thiện quá trình học của mô hình.

Nhìn chung, mối tương quan giữa loss và epoch phản ánh quá trình học của mô hình và là một chỉ báo quan trọng về hiệu suất và sự phát triển của mô hình qua thời gian. Đánh giá sự thay đổi của loss qua các epoch giúp xác định các vấn đề như overfitting hoặc underfitting và hỗ trợ trong việc tinh chỉnh mô hình để đạt được hiệu suất tốt nhất.

##### Mục Đích của Validation Loss

**Tinh chỉnh mô hình:** Validation loss giúp nhà nghiên cứu hiểu được mô hình hoạt động tốt đến mức nào trên dữ liệu chưa từng thấy, cho phép họ tinh chỉnh các hyperparameters để cải thiện hiệu suất mô hình.

**Tránh Overfitting:** Một trong những vấn đề lớn trong học máy là overfitting, tức là mô hình hoạt động tốt trên dữ liệu huấn luyện nhưng không tổng quát hóa tốt trên dữ liệu mới. Quan sát sự thay đổi của validation loss qua các epoch giúp phát hiện overfitting. Nếu validation loss tăng trong khi training loss giảm, đây có thể là dấu hiệu của overfitting.

**So sánh mô hình:** Validation loss cũng được sử dụng để so sánh hiệu suất giữa các mô hình khác nhau hoặc giữa các phiên bản của cùng một mô hình với các hyperparameters khác nhau, giúp lựa chọn mô hình tốt nhất.

**Tính Toán Validation Loss**

Validation loss được tính toán bằng cách sử dụng một hàm mất mát (loss function), tương tự như cách tính training loss. Hàm mất mát Cross-Entropy Loss cho bài toán phân loại. Điểm khác biệt chính là validation loss được tính dựa trên dữ liệu kiểm định thay vì dữ liệu huấn luyện.

**Kết Luận**

Validation loss là một công cụ quan trọng trong việc phát triển và tinh chỉnh các mô hình học máy, giúp nhà phát triển mô hình có cái nhìn rõ ràng hơn về cách mô hình của họ tổng quát hóa trên dữ liệu mới và không thấy trong quá trình huấn luyện, từ đó giúp tránh overfitting và chọn lựa mô hình tốt nhất.

##### Nếu bạn không có dữ liệu riêng để kiểm tra validation loss

bạn sẽ gặp một số hạn chế trong quá trình phát triển và đánh giá mô hình của mình:

**Không thể Tinh chỉnh Mô hình Hiệu quả**

Validation loss giúp đánh giá mô hình trên dữ liệu không tham gia vào quá trình huấn luyện, cho phép bạn tinh chỉnh hyperparameters một cách hiệu quả. Không có dữ liệu kiểm định, bạn sẽ gặp khó khăn trong việc tối ưu hóa mô hình mà không rơi vào tình trạng overfitting.

**Khó Khăn trong Việc Phát hiện Overfitting**

Một trong những lợi ích chính của việc sử dụng validation set là để phát hiện overfitting. Nếu mô hình của bạn hoạt động tốt trên training set nhưng không tốt trên validation set, đó có thể là dấu hiệu của overfitting. Không có validation set, bạn có thể không nhận ra mô hình của mình đã quá khớp với training set mà không có khả năng tổng quát hóa tốt trên dữ liệu mới.

**Giảm Khả năng Đánh giá Tổng quát hóa của Mô hình**

Validation set giúp đánh giá khả năng tổng quát hóa của mô hình trước khi áp dụng nó trên test set hoặc dữ liệu thực tế. Không có validation set, khả năng đánh giá này bị hạn chế, làm tăng rủi ro khi triển khai mô hình vào sản phẩm thực tế.

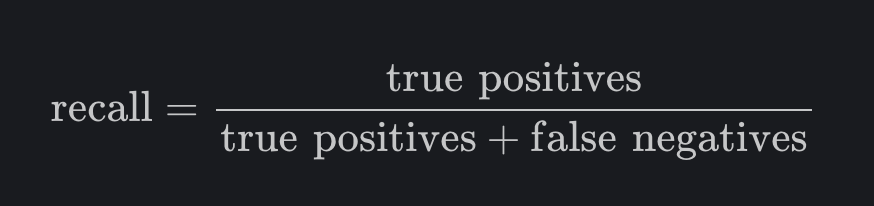
**Cách Tiếp cận khi Không Có Dữ liệu Kiểm định**

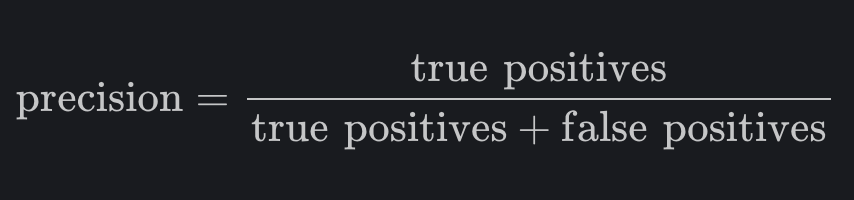
Nếu bạn không thể cung cấp dữ liệu riêng biệt cho việc kiểm định, một số phương pháp có thể giúp bạn đánh giá mô hình một cách tương đối:

**Sử dụng phương pháp chia tập dữ liệu:** Chia tập dữ liệu huấn luyện của bạn thành hai phần: một phần để huấn luyện và một phần nhỏ hơn để kiểm định. Cách này giúp đánh giá mô hình mà không cần dữ liệu bên ngoài.

**Sử dụng Cross-validation:** Cross-validation, đặc biệt là k-fold cross-validation, là một phương pháp mà tập dữ liệu được chia thành k phần. Mô hình được huấn luyện và đánh giá k lần, mỗi lần sử dụng một phần dữ liệu khác nhau làm tập kiểm định và phần còn lại làm tập huấn luyện. Điều này giúp đánh giá khả năng tổng quát hóa của mô hình mà không cần tới dữ liệu kiểm định riêng biệt.

##### True Positive (TP) False Positive (FP) True Negative (TN) False Negative (FN)





**True Positive (TP)**:

Định nghĩa: Số lượng mẫu mà mô hình đã dự đoán chính xác là positive.

Ví dụ: Trong tác vụ phát hiện email spam, một email được mô hình dự đoán là spam và thực tế cũng là spam thì đó được gọi là True Positive.

**False Positive (FP)**:

Định nghĩa: Số lượng mẫu mà mô hình dự đoán là positive nhưng thực tế lại là negative.

Ví dụ: Nếu một email không phải spam (negative) nhưng mô hình dự đoán nó là spam (positive), đó là một False Positive.

**True Negative (TN)**:

Định nghĩa: Số lượng mẫu mà mô hình dự đoán chính xác là negative.

Ví dụ: Trong tác vụ phát hiện email spam, một email không phải spam và cũng được mô hình dự đoán không phải là spam, đó là một True Negative.

**False Negative (FN)**:

Định nghĩa: Số lượng mẫu mà mô hình dự đoán là negative nhưng thực tế lại là positive.

Ví dụ: Nếu một email là spam (positive) nhưng mô hình dự đoán rằng nó không phải spam (negative), đó là một False Negative.

##### 1 0 bình luận xây dựng hoặc không liên quan gì tới TP và FP FN và FP

Các giá trị True Positive (TP), False Positive (FP), True Negative (TN), và False Negative (FN) có mối quan hệ trực tiếp với cách bạn định nghĩa các nhãn (label) trong bài toán phân loại của mình. Trong trường hợp của bạn, giả sử nhãn "1" được dùng để chỉ các mẫu thuộc lớp positive, và nhãn "0" cho các mẫu thuộc lớp negative. Dựa vào định nghĩa này, ta có:

True Positive (TP): Là số lượng các mẫu mà mô hình dự đoán là "1" (positive) và thực sự nhãn đúng của chúng cũng là "1". Nói cách khác, đây là các trường hợp mà mô hình dự đoán đúng mẫu là positive.

False Positive (FP): Là số lượng các mẫu mà mô hình dự đoán là "1" nhưng thực sự nhãn của chúng là "0" (negative). Đây là các trường hợp mô hình "nhầm lẫn", dự đoán một mẫu không phải positive là positive.

True Negative (TN): Là số lượng các mẫu mà mô hình dự đoán là "0" (negative) và thực sự nhãn đúng của chúng cũng là "0". Nói cách khác, đây là các trường hợp mà mô hình dự đoán đúng mẫu là negative.

False Negative (FN): Là số lượng các mẫu mà mô hình dự đoán là "0" nhưng thực sự nhãn của chúng là "1" (positive). Đây là các trường hợp mô hình bỏ sót, không nhận ra một mẫu positive.

Việc bạn đặt nhãn "1" cho positive và "0" cho negative là một cách tiếp cận thông thường trong các bài toán phân loại nhị phân. Cách bạn đánh nhãn này quy định cách tính và phân loại các giá trị TP, FP, TN, và FN khi đánh giá mô hình. Điều này ảnh hưởng trực tiếp đến việc tính toán các chỉ số đánh giá hiệu suất mô hình như độ chính xác, độ nhạy, độ đặc hiệu và F1 Score, vì các chỉ số này dựa trên việc phân loại đúng sai dựa trên nhãn thực tế và nhãn dự đoán.

##### Phobert-base với epoch 3 thì không có được f1 score. Vì sao mình vẫn chọn?

Dựa trên thông tin bạn đã cung cấp, việc chọn PhoBERT-base ngay cả khi epoch thứ 3 không có F1 Score có thể dựa trên một số lý do sau:

Hiệu suất ổn định ở các epoch trước: Nếu PhoBERT đã cho thấy hiệu suất ổn định và cao ở các epoch trước, điều này có thể chỉ ra rằng mô hình có khả năng học tốt từ dữ liệu. Sự thiếu thông tin về epoch thứ 3 có thể là kết quả của một sự cố kỹ thuật hoặc lỗi trong quá trình đánh giá, nhưng không nhất thiết phản ánh chất lượng tổng thể của mô hình.

Cân nhắc giữa các chỉ số khác: Precision, Recall, và F1 Score ở epoch 2 của PhoBERT cho thấy một sự cân bằng tốt giữa khả năng phát hiện đúng các trường hợp positive và giảm thiểu số lượng false positives. Điều này quan trọng cho các ứng dụng mà việc duy trì một sự cân bằng giữa việc phát hiện và tránh nhầm lẫn là quan trọng.

Độ tin cậy và danh tiếng của mô hình: PhoBERT là một mô hình được phát triển bởi VINAI, một tổ chức có uy tín trong lĩnh vực AI và NLP. Sự ủng hộ từ một tổ chức có danh tiếng có thể là một yếu tố quan trọng trong việc quyết định, đặc biệt nếu có sự hỗ trợ tích cực và cộng đồng sử dụng mạnh mẽ.

Tính linh hoạt và khả năng tùy chỉnh: Nếu PhoBERT cho thấy khả năng được tinh chỉnh (fine-tuned) tốt trên bộ dữ liệu cụ thể hoặc dễ dàng được tích hợp vào hệ thống hiện tại, đó có thể là lý do mạnh mẽ để chọn mô hình này.

Tối ưu cho bối cảnh sử dụng cụ thể: Có thể PhoBERT đã được chứng minh là có hiệu suất tốt trong những tình huống hoặc bộ dữ liệu tương tự với ứng dụng mục tiêu của bạn, khiến nó trở thành lựa chọn ưu tiên.