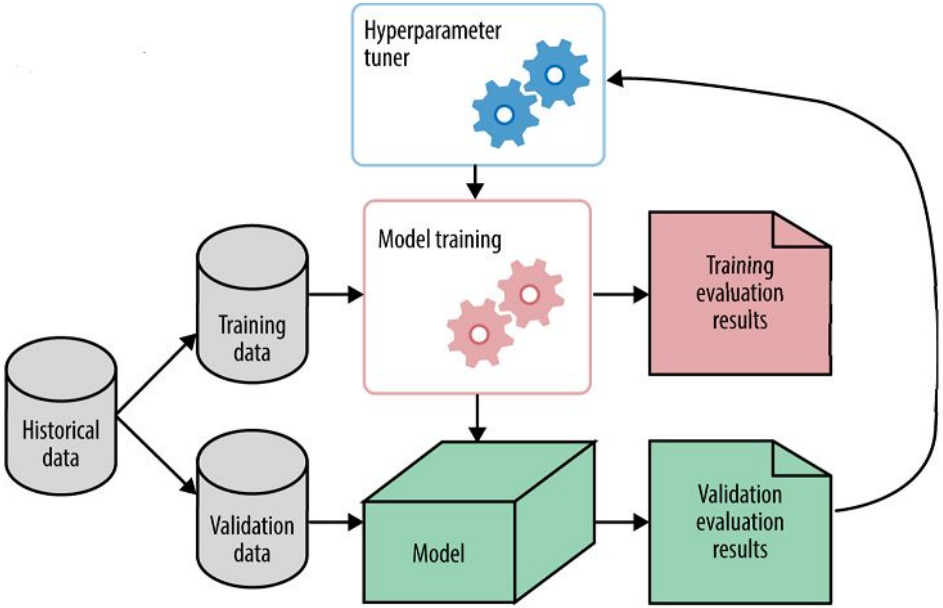
Designing Good Validation



Nội dung

1. Snooping on the leaderboard
2. Bias and Variance
3. Tunning model Validation system
4. Pytorch code examples

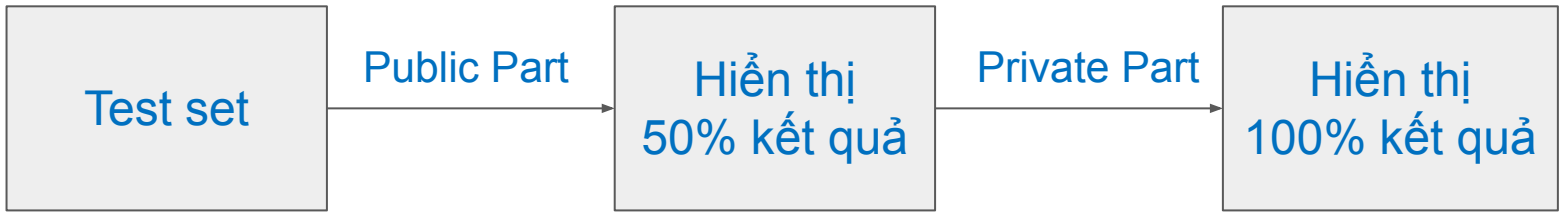
1 - Snooping on the Leaderboard

Phần lớn các cuộc thi đều chia **Test Set** 2 phần:

- **Public Part**: được hiển thị trên leaderboard
- **Private Part**: được sử dụng để tính cho kết quả cuối cùng

Chú ý:

- Các phần test set này thông thường sẽ được **chia ngẫu nhiên** và khi **toàn bộ Test Set** được **công bố** sẽ hoàn toàn **chứa Public Part** và **Private Part.**
- **Không** chứa dữ liệu nằm ngoài Public Part và Private Part.



1 - Snooping on the Leaderboard

1- Một số điều cần lưu ý khi đọc Leaderboard:

- Training data và testing data cần có cùng sự phân phối (distribution). **Tuy nhiên**, trong tập **test** thì **private part** và **public part** có thể **khác** nhau về distribution.

- Ngay cả khi Training data và testing data cùng một phân phối, thì việc **chênh lệch** phân phối giữa **private part** và **public part** làm **ảnh hưởng** đến quá trình **phán đoán** và **phân tích** trong nửa đầu thời gian cuộc thi (Training data và Public Part).

- Public test data (Public Part) chỉ nên được dùng cho việc submit kết quả lên Leaderboard. Không nên để Public test data tham gia vào quá trình huấn luyện model.

- Kết quả submit có thể **không thay đổi** trên **Public** Part **nhưng** thay đổi trên **Private** Part.

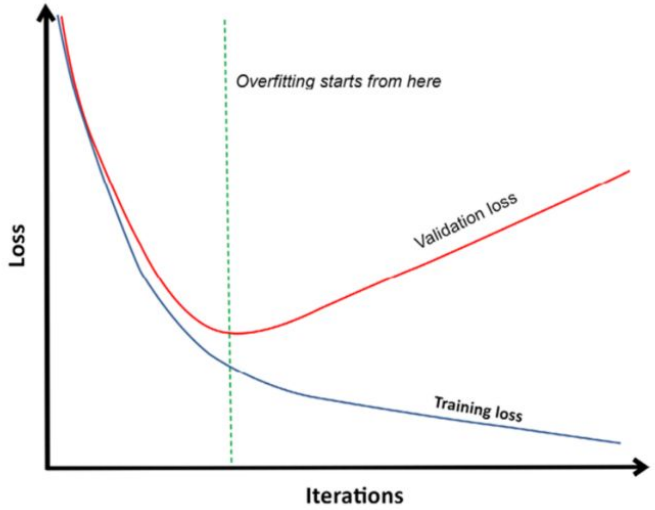
1 - Snooping on the Leaderboard

2- Một số chiến lược submit và theo dõi leaderboard:

- Sử dụng cross-validation trong quá trình đánh giá local scoring.
- Kiểm soát sự phân phối dữ liệu trong quá trình chia train-test-val.
- Kiểm tra local scoring và Leaderboard để xem giữa 2 kết quả có tương quan với nhau hay không.
- Sử dụng kỹ thuật ensemble.

2 - Bias and Variance

- Bias thể hiện việc model chưa đủ phức tạp để học và bao quát hết data.
- Variance thể hiện việc model quá phức tạp để học trên bộ dữ liệu. Model quá chú trọng vào chi tiết và nhiễu bên trong data .



2 - Bias and Variance

Quá trình overfitting được thể hiện ở nhiều cấp độ khác nhau:

- Ở mức training data, khi xây dựng một model quá phức tạp để training trên bộ data đơn giản.
- Ở mức độ validation set, khi tune model quá nhiều với một validation set cụ thể.
- Ở mức độ thể hiện trên public leaderboard, khi kết quả hiển thị còn quá xa so với mong đợi (local test cao nhưng public score lại thấp).
- Ở mức độ thể hiện trên private leaderboard, Public score hoàn toàn cao nhưng private score lại thấp dẫn đến điểm số chung cuộc thấp.

2 - Trying different splitting strategies

Hai chiến lược chia dữ liệu thường sử dụng trong việc huấn luyện và đánh giá model:

- **Holdout system:** rủi ro không chọn được đúng mẫu dữ liệu cần thể hiện.

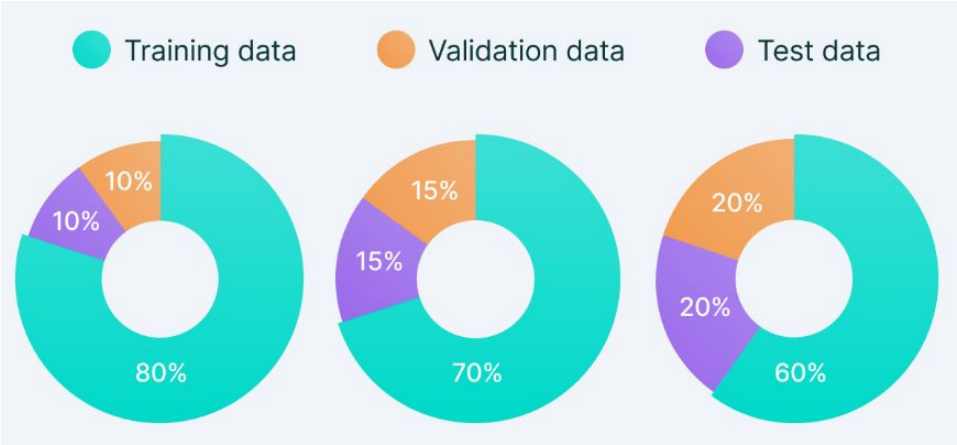
- **Probabilistic approach**: dựa vào xác suất và lấy trên nhiều mẫu để tổng hợp model. Một số cách tiếp cận phổ biến như: cross-validation, leave-one-out, bootstrap. Với mỗi chiến lược cross-validation sẽ có những cách lấy mẫu khác nhau tùy thuộc vào mục đích của dữ liệu như (simple random sampling, strtified sampling, sampling by groups, time sampling)

2 - Trying different splitting strategies

1- The basic train-test split

- Quá trình chia dữ liệu sẽ nhận tỷ lệ lấy mẫu không đều (lệch phân phối) và tỷ lệ này càng cao khi sử dụng trên tập dữ liệu nhỏ.

- Để đảm bảo tính nhất quán có thể sử dụng cơ chế phân tầng (**stratification**). Tham số **stratify** được cung cấp trong hàm **train_test_split**.



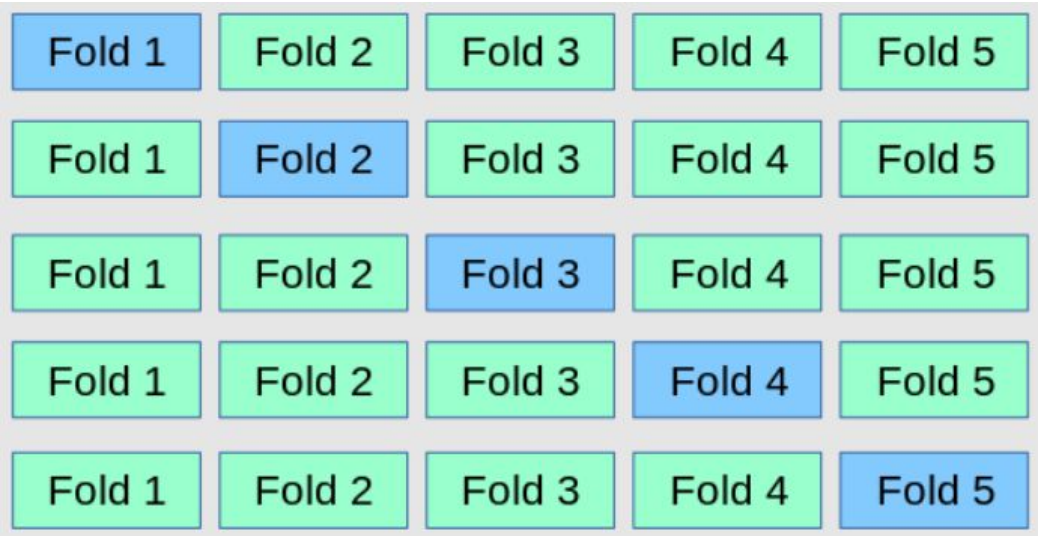
2 - Trying different splitting strategies

2- Probabilistic evaluation methods

K-FOLD CROSS-VALIDATION

K càng nhỏ (tối thiểu 2) số phần chia sẽ càng nhỏ và model sẽ hoạt động kém hơn với K lớn.

K càng cao, sẽ dễ bị mất đi những thuộc tính quan trọng của dữ liệu và làm giảm khả năng dự đoán trên unseen data

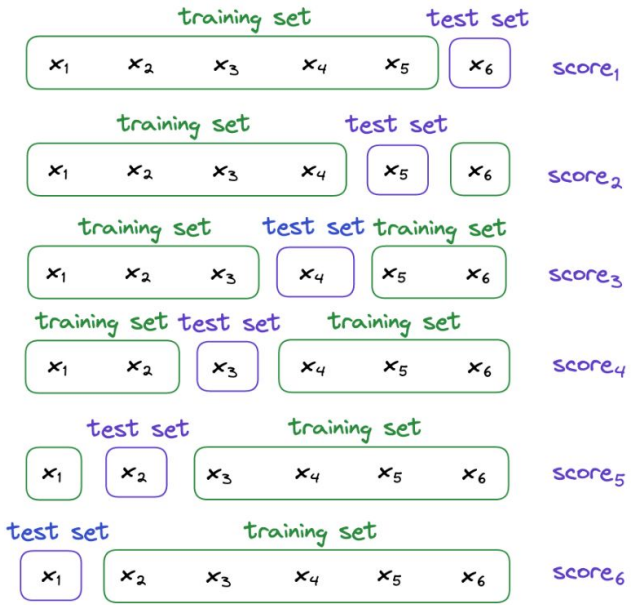


Chia Training data thành train-test và thực hiện K-FOLD trên tập train

2 - Trying different splitting strategies

2- Probabilistic evaluation methods

LEAVE-ONE-OUT sử dụng cơ chế training model n lần với n là kích thước của data set. Mỗi lần chỉ có 1 mẫu làm test data trong khi phần còn lại dùng để training model. Kết quả cuối cùng sẽ được tính **trung bình cộng** của các score.



3 - Tuning model validation system

Quá trình chia train-test-val được coi là một quá trình trial-and-error vì vậy cần phải kiểm tra tính nhất quán để việc chia đạt được hiệu quả:

- Kiểm tra tính nhất quán của local test - các lần cross-validation không quá khác biệt với nhau.
- Kiểm tra validation error (hoặc accuracy) trên local-test có phù hợp với validation error (hoặc accuracy) trên leaderboard hay không.

Hai vấn đề ảnh hưởng chính khi việc kiểm tra thất bại:

- Chưa đủ data để sử dụng
- Dữ liệu quá đa dạng và sự khác biệt phân phối giữa các phần là quá chênh lệch => thay đổi cơ chế chia fold

3 - Tuning model validation system

Hai cách khắc phục chính khi việc kiểm tra thất bại:

- Sử dụng K lớn - có thể làm giảm khả năng dự đoán của model trên unseen data. Tuy nhiên việc sử dụng các phần lớn hơn sẽ giúp model ổn định quá trình đánh giá hơn => đánh đổi hiệu suất học của model với K.

- Tính trung bình kết quả của k-fold validation.