Modul 3

Model Evaluation

Data Science Program

Purwadhika
Startup and Coding School

Agenda

Session 0: Brief review of past topic

Session 1: Introduction to Model Evaluation

Session 2: Model Selection

Session 3: Evaluation Metrics

Session 4: Recap

Assignment

Purwadhika
Startup and Coding School

Objective

Understand what are options for model evaluation

Understand a complete picture of model performance

Understand the concept of important evaluation metrics

High Performance Model

- **What** and **How to** produce high performance model.
- Play with our **data** to improve the performance.
- Use an **existing model** that suitable for our problem and do transfer learning and tuning.
- Further improve our model by **tuning** our model algorithm using hyperparameter.
- Use **ensemble** method such as stacking, boosting, and bagging to **combine** classifiers.

Introduction

- Goals: Train the model to give good prediction!

- We want to pick the *best* model suitable for our problem set.
- But how we choose the *best* model out of many?

- Does good prediction mean high accuracy?
- Or is it different from task to task?

Purwadhika
Startup and Coding School

Introduction

- We can evaluate our choices end-to-end to help us estimate our classifier better.

- Basically, we can evaluate our choice on these two stages:

1. Pre & During Training
2. Post Training

Model Selection

- In the pre/during training phase, we can do some test to check **which model, how much data** would give us potentially higher accuracy.
- By doing so, we get a solid line to continue to further process to build our model.

Model Evaluation

- In the post training phase, we can check whether our model **answer** our problem.
- We can also investigate other evaluation metrics that help us fully understand our model.
- Based on the evaluation, we can iterate to get better result.

Training vs Test

- Training Data is used for us to train model. As a result, we can get the **training accuracy**, which means how well our model behave on our training data.
- Training accuracy does not give a good indication of how well the model generalize over an independent data set.

Training vs Test

- Test data ideally should be independent from training data to give unbiased error estimation.
- Training accuracy does not necessarily mean that our model is good, because a high value of training accuracy might indicate **overfitting**.
- We should look for the model that might produce lowest test error.

Training vs Test

- If we have a large training data, we can build a good classifier.

- Even we can split it to be training data and test set, for later test the model.
- But by splitting, we reduce the number of data to use for training, which might introduce underfitting.

- Even worse, what if we only have small amount of data?

Target Shuffling

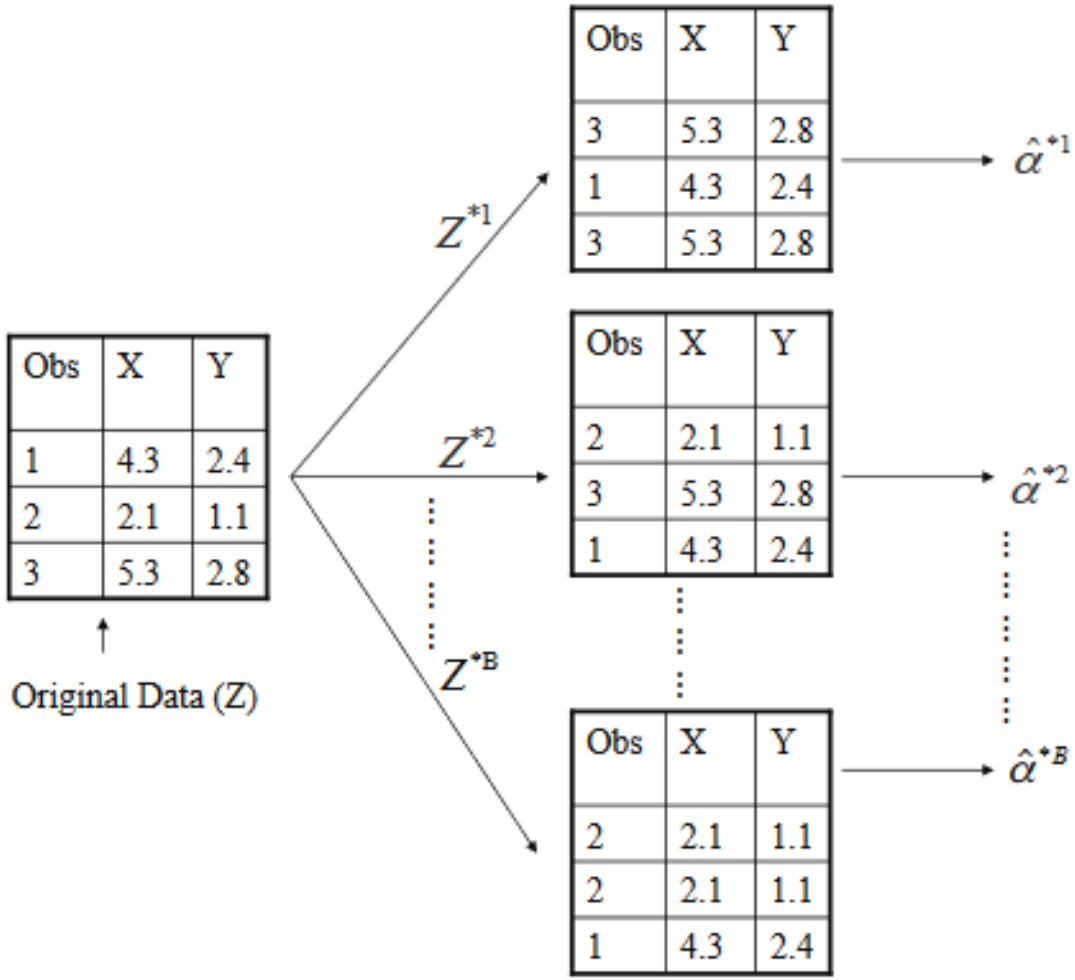
- We can train many classifier by shuffling our label (or target) and pick the best performance as our measurement.

- 1. Train a classifier and observe that is has A_1 percent accuracy.
  2. Shuffle your labels, train another classifier, and observe that is has A_2 percent accuracy.
  3. Repeat Step 2 multiple times and find B = best(A).

Bootstrap

- Bootstrap, as we discussed previously, is used to estimate parameter by using sampling with replacement.
- If we don’t have a large amount of data, we can use bootstrap to simulate obtaining independent data by sampling original data.

Bootstrap



Purwadhika
Startup and Coding School

Bootstrap

- Let’s try bootstrap to give idea.
- Duration: ~10min

Purwadhika
Startup and Coding School

Cross Validation

- We can use all of our data for training as well as testing using cross-validation.
- Cross validation split data into training and test to do a **model checking**, not model building.
- One of the common form of cross validation is **k-fold cross validation.**

k-Fold CV

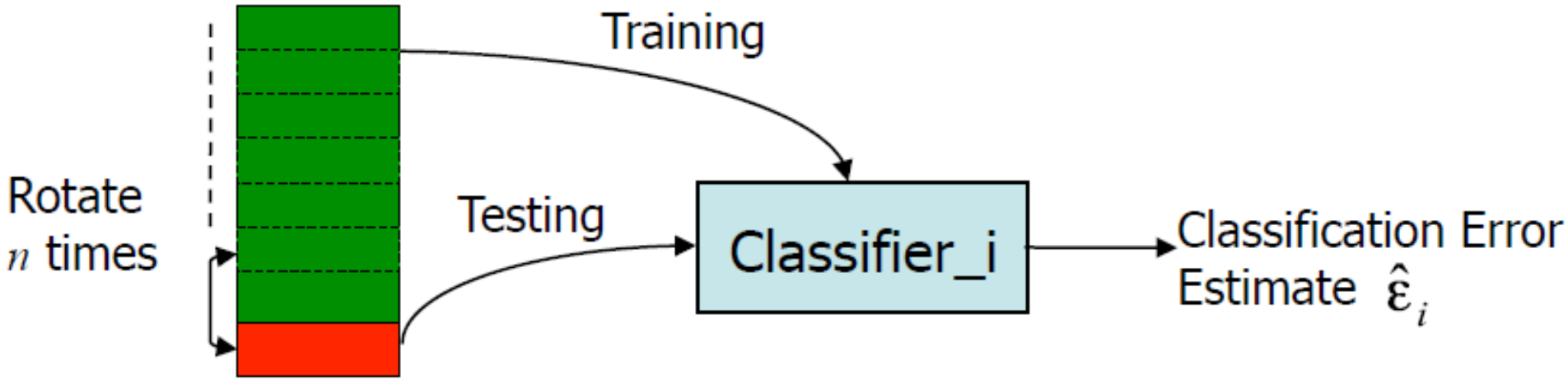
- K-Fold CV divide the data into **k subsets.**
- Each time, **one** subset is used for validation set and the other **k-1** subset is used for training.
- Repeat the process for **k** times.
- The error estimation is **averaged** over all k trials to get total effectiveness of our model.

k-Fold CV

- This significantly reduces bias as we are using most of the data for fitting.
- Also significantly reduces variance as most of the data is also being used in validation set.

- How can we determine **k**?

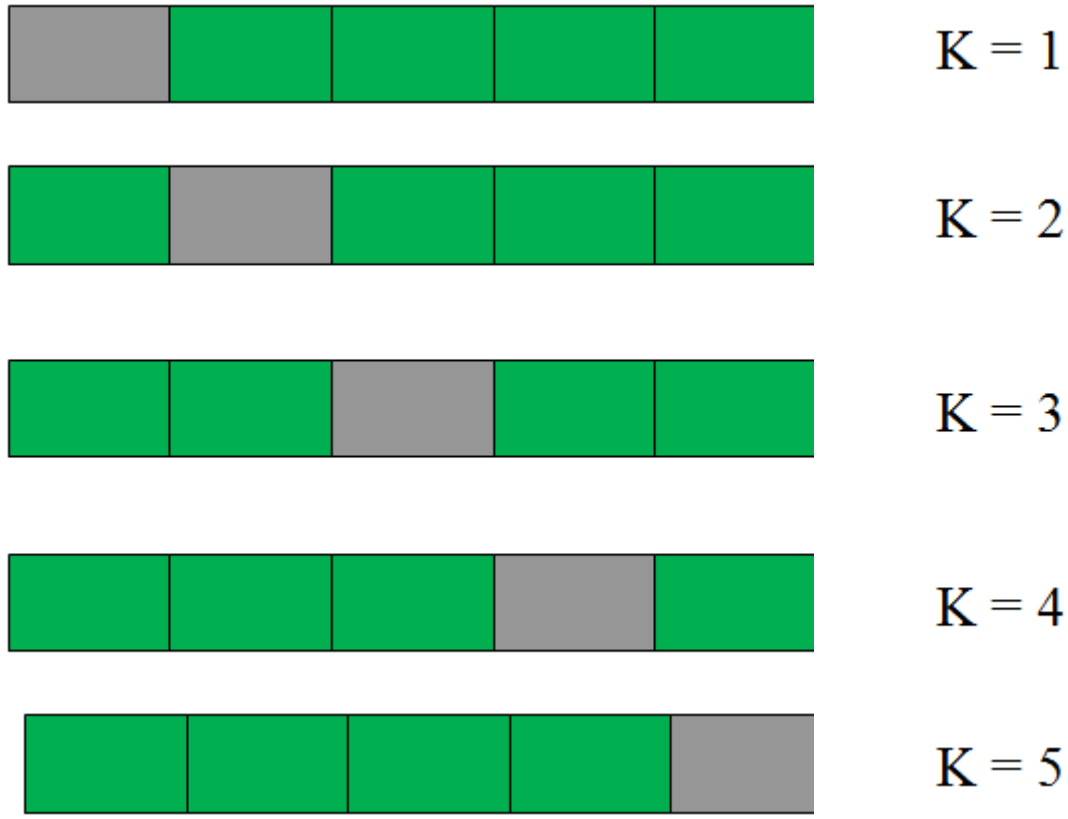
k-Fold CV



$$\hat{\epsilon} = \frac{1}{n}\sum_{i} \hat{\epsilon}_i$$

Purwadhika
Startup and Coding School

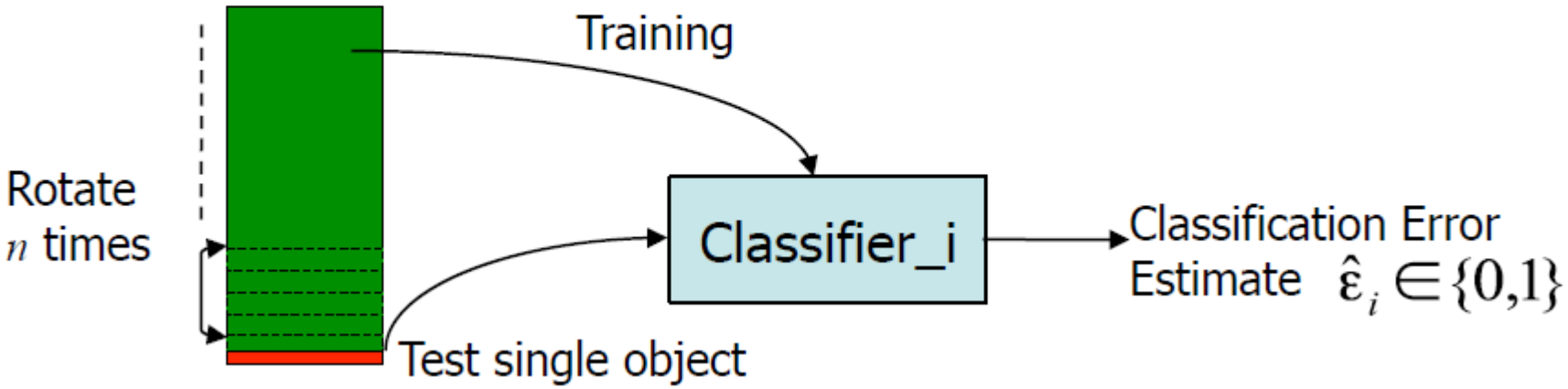
5-Fold CV



Purwadhika
Startup and Coding School

LOOCV

- If k = n, we have **Leave-one-out CV.**



Purwadhika
Startup and Coding School

Cross Validation

- Let’s try!
- Duration: ~15-20min

Purwadhika
Startup and Coding School

Model Selection

- Let say we already define our problem set, already gather data from internal and external, we know we can use CV for model checking. And now we are ready to train our model.
- Which model should we choose? Should we choose the one we are familiar with? Or the simplest one? Or the most complicated?

Model Selection

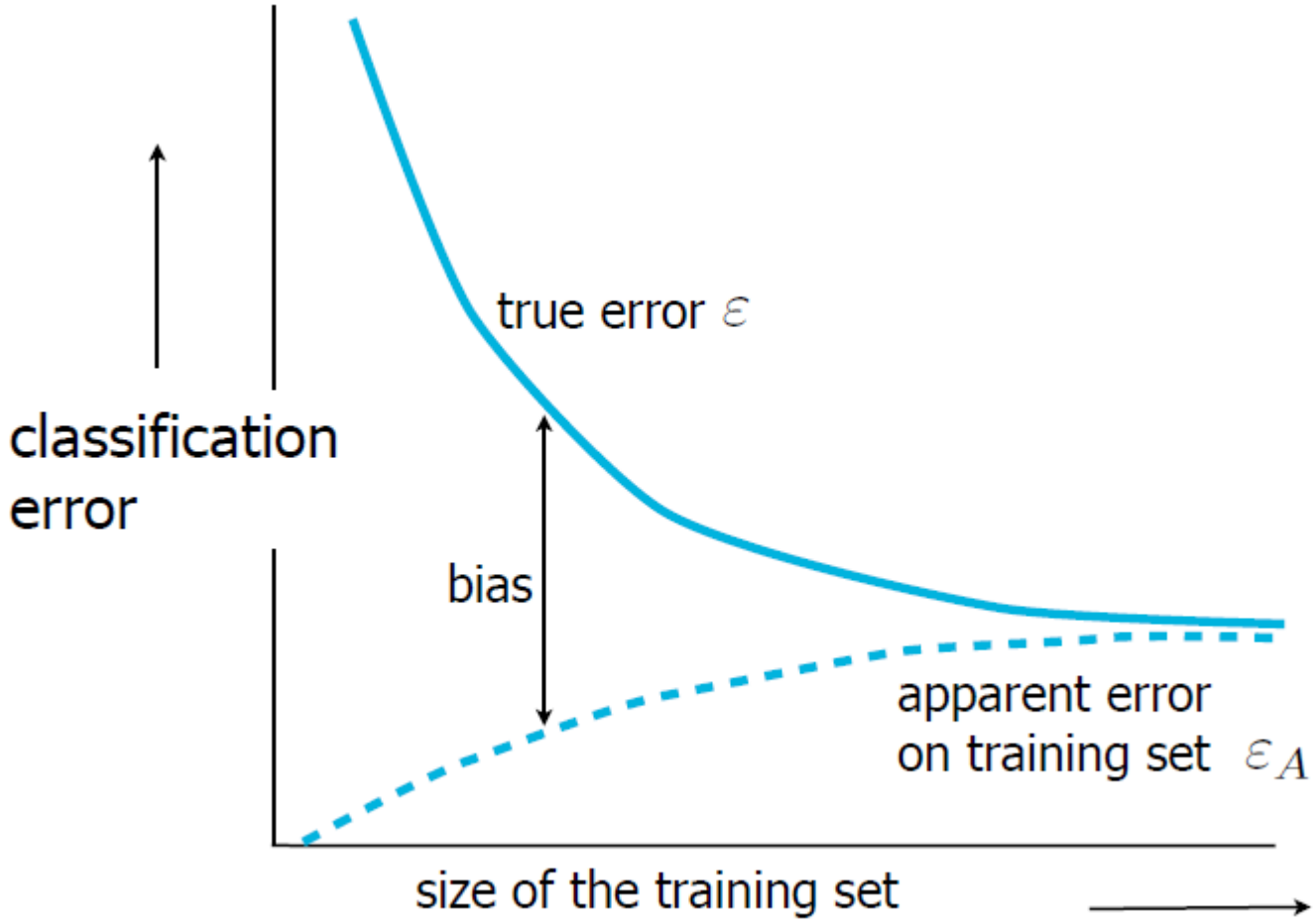
- We can do all. We can pick with model that we are familiar, or we are confident that give higher accuracy, or intuition.
- But we can do better! We can examine the apparent error of the various classifiers and continue from there.

Learning Curves

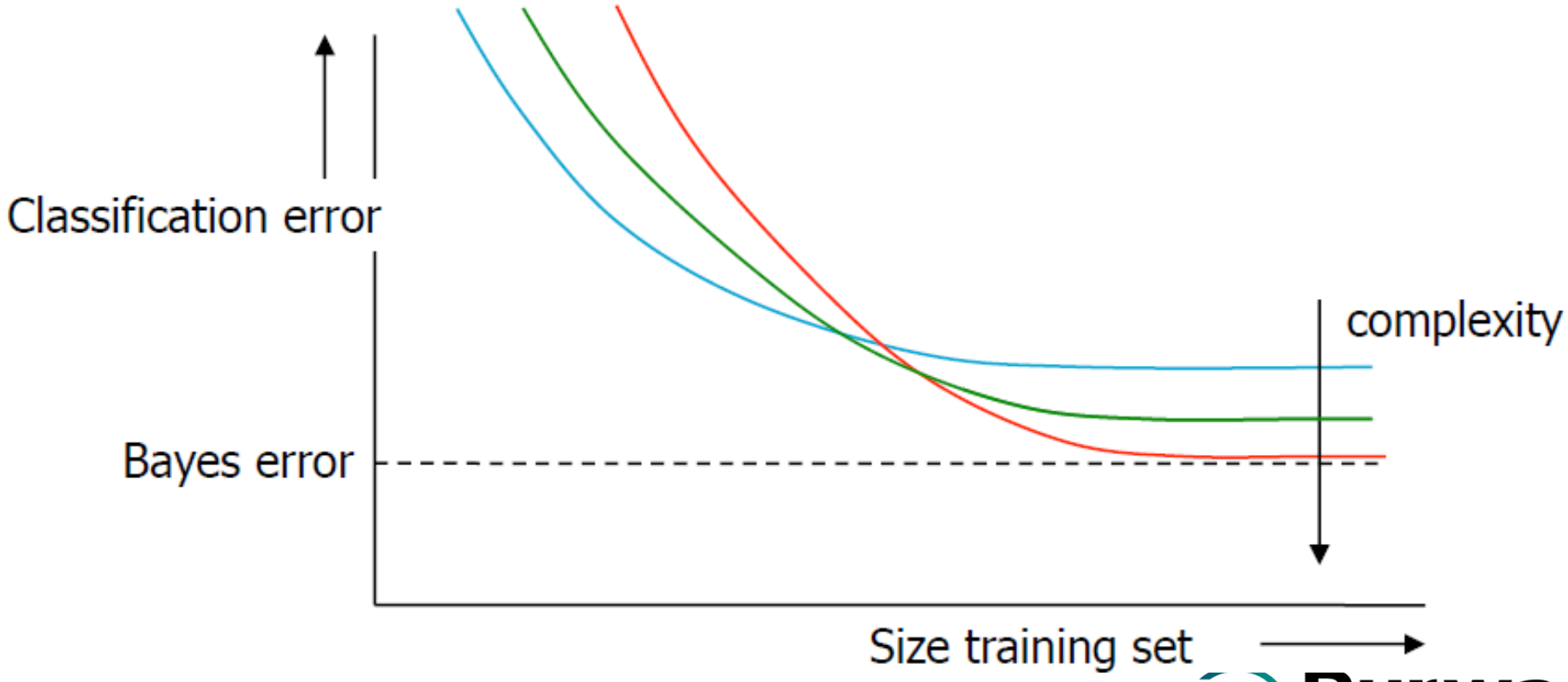
- Curves that plot classification (training and test) errors against the number of samples in training set.

- Give insight in:
  - Usefulness of additional data
  - Classifiers comparison

Learning Curves



Learning Curves



Learning Curves

- Let’s try!
- Duration: ~30-45min

Purwadhika
Startup and Coding School

Accuracy?

- **Most** of the time, accuracy can be used to determine the performance of our model. But sometimes, it does not give the full information we need.
- Consider this, two class classification problem with an **imbalanced** dataset for predicting class A with only 1 sample and class B with 999 sample.

$$Accuracy = \frac{\#\ correct\ prediction}{\#\ data}$$

Purwadhika
Startup and Coding School

Accuracy?

- You build classifier that 99% correctly classified the problem set. That's amazing! Or is it?
- Suppose you train a Dummy Classifier which take the most frequent class as its prediction (in our case is class B). We will end up with 99.9% accuracy!
- So, is your classifier better or worse than the dummy?

Accuracy?

- Let’s try to get the idea!
- Duration: ~15-20min

Purwadhika
Startup and Coding School

Accuracy?

- If you got the similar accuracy compare to base classifier, there might be some of these problems:

- Class Imbalance
-
- Missing Features
- Untuned classifier
- Or maybe accuracy does not show the whole information

Evaluation Metrics?

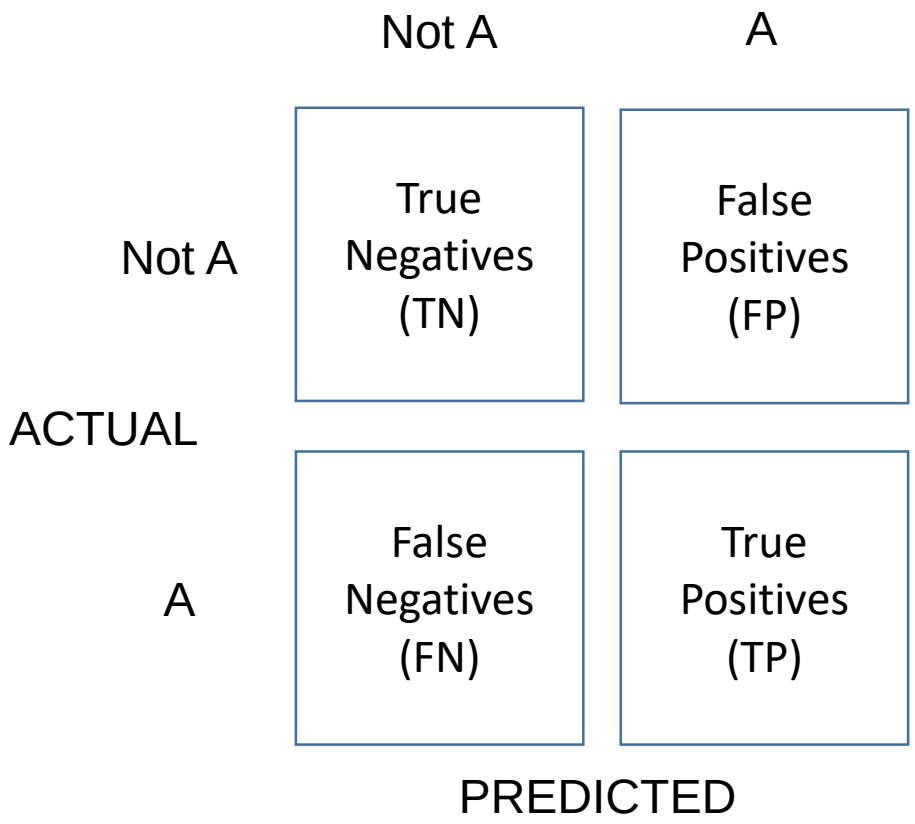
- So if accuracy is not showing the whole picture. Are there others metrics that we can check to evaluate our model?
- Yes! And it give us intuition on how our model works!

Confusion Matrix

- Confusion Matrix is a N x N matrix which N indicates the number of classes.
- In the 2 class problem, we will have 2x2 confusion matrix.
- It gives a more detailed view than overall accuracy / error rate.
- It is the only way to get information of class distribution of the actual value and predicted value.

Confusion Matrix

- For two class classification, this is the confusion matrix



Purwadhika
Startup and Coding School

Confusion Matrix

- Now we know the distribution of prediction. Then what?
- We can investigate our strategy to further improve our classifier.
- For example, we know from the confusion matrix that our model lack the power to predict certain class. We can incorporate other features that help predict that class better.

Confusion Matrix

- We can also calculate different metrics to evaluate our model.
- **Accuracy:** Overall how often the classifier is correct

$$\frac{TP+TN}{N}$$

- **Error:** Overall, how often the classifier is wrong

$$\frac{FP+FN}{N} \text{ or } 1 - Accuracy$$

Confusion Matrix

- **True Positive Rate (TPR)** also known as **Sensitivity** or **Recall**: For all positive instances, what fraction is correctly identified as positive.

$$\frac{TP}{TP+FN}$$

- **True Negative Rate (TNR)** also known as **Specificity:** For all negative instances, what fraction is correctly identified as negative.

$$\frac{TN}{TN+FP}$$

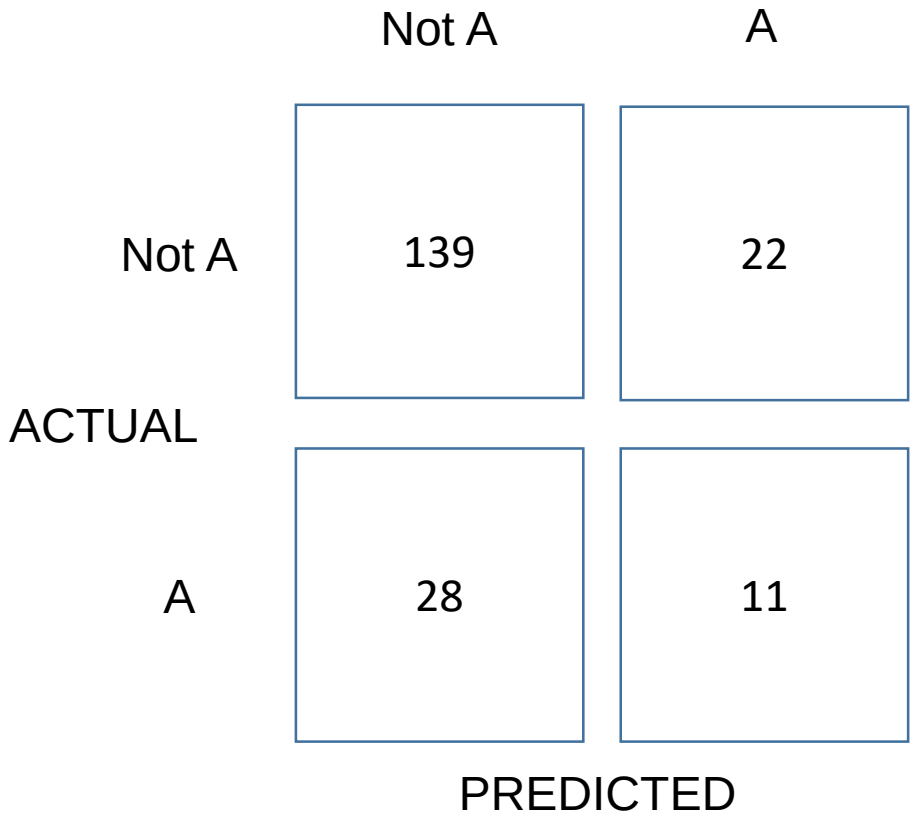
Confusion Matrix

- **Precision**: What fraction of positive predictions are correct?

$$\frac{TP}{TP+FP}$$

Confusion Matrix

- Can you calculate the metrics based on this confusion metrics?

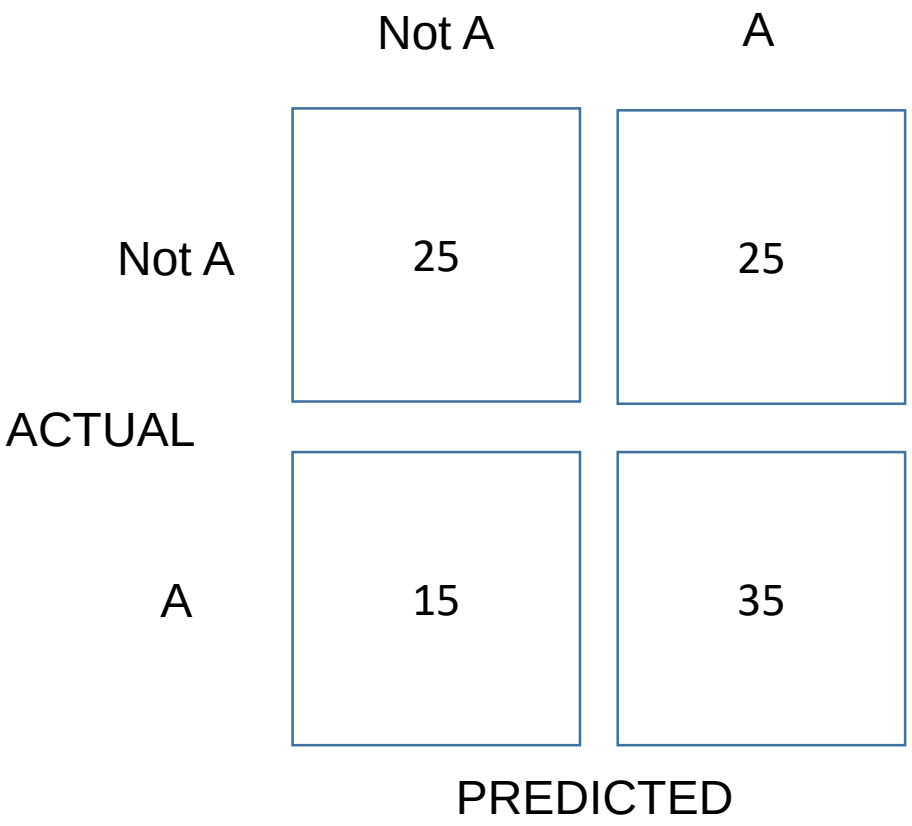


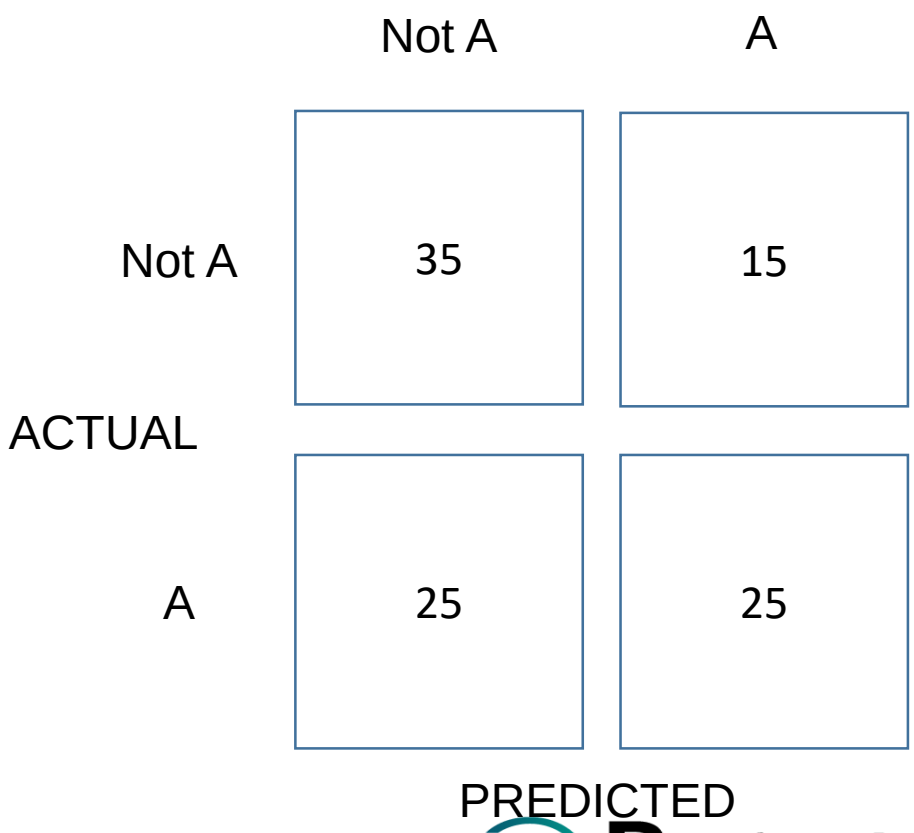
Purwadhika
Startup and Coding School

FP vs FN

- We know that TP and TN means we correctly classified our data. The higher these numbers the better the accuracy.
- But how about FP and FN? These two are sources of error to our classifier.
- Which one we should reduce to boost our classifier performance?

FP vs FN





Purwadhika
Startup and Coding School

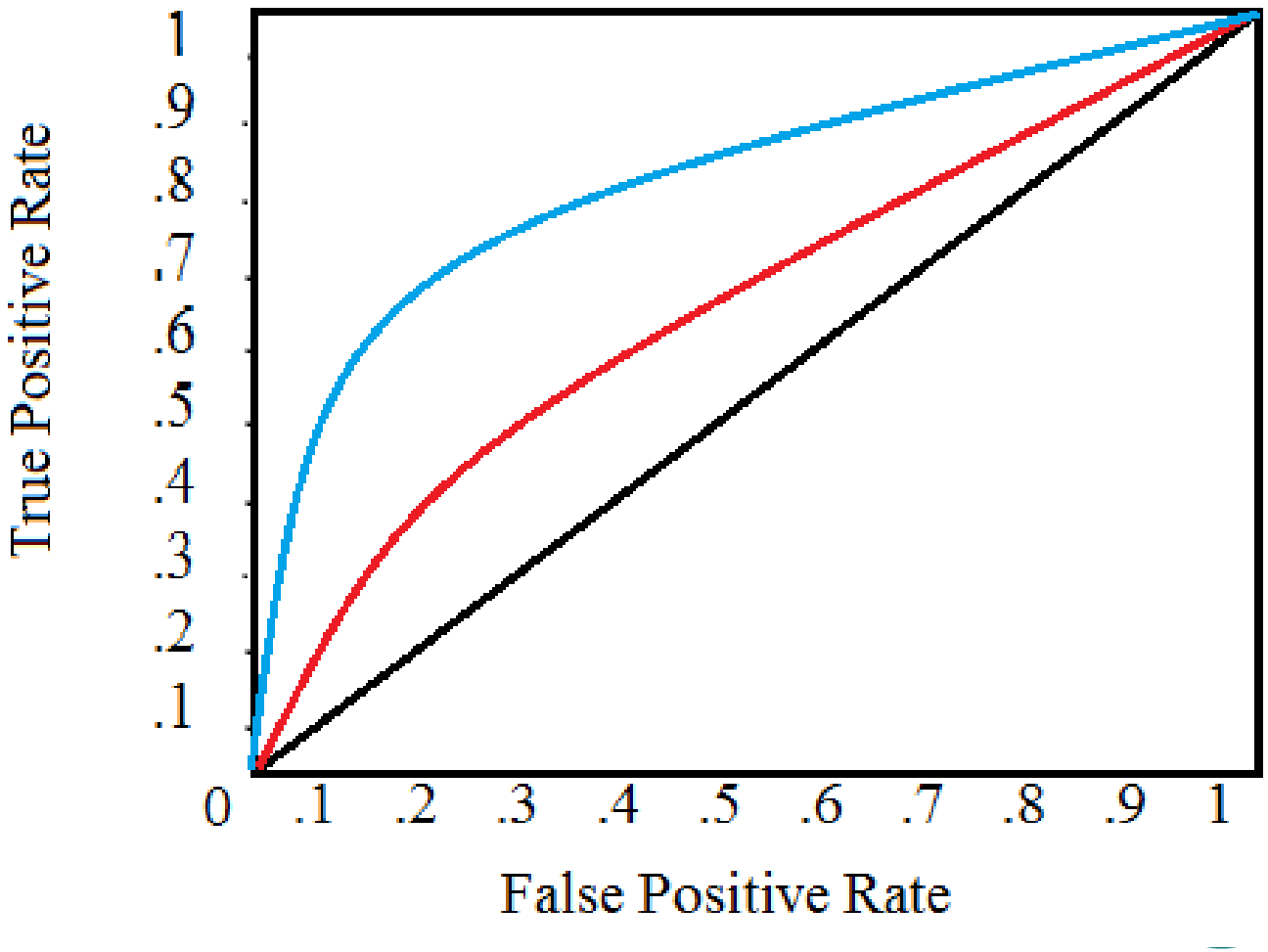
Confusion Matrix

- Let’s try build our confusion matrix!
- Duration: ~15min

ROC

- From confusion matrix, we can also get another way to evaluate our model.
- By calculating **Sensitivity** and **Specificity**, we can create ROC (Receiver-Operator Characteristic) curve. Both metrics have values in range [0, 1]
- We plot False Positive Rate (1 – Specificity) on X axis, and Sensitivity on Y axis.

ROC



AUC

- We can measure performance of classifier on ROC curve by integrating the curve or we can call it **Area Under Curve**.
- The perfect classifier will have AUC value of 1, will the worst will have value of 0.
- Using AUC value, we can compare performance between classifiers.

AUC

- Let’s try!
- Duration: ~20min

Purwadhika
Startup and Coding School

Recap

- We understand the **how** to do model selection.
- We understand the effect of amount of **training** data and how to cope with it.
- We know what other **metrics** to evaluate our model.