Fakultas Informatika
School of Computing
Telkom University

Machine Learning

Evaluation Metrics

ADF

Today's Agenda

- Performance Metrics
- Validation Data

Performance Metrics

Learning Definition

- computer programs that automatically improve their performance through experience (seeing training data)
- First, we need to define how to calculate the performance

Prediction Type

- Regression
  - Mean Squared Error, Root Mean Squared Error
  - Mean Absolute Error
  - Mean Absolute Percentage Error
- Classification
  - Accuracy
  - F1-Score, Precision, Recall

Regression

Time Series Prediction

- Let's say we build a time series prediction to predict rainfall using past rainfall history (3 series)
- We build the data as follow
- Then from the training data, we train two prediction models

t-3	t-2	t-1	t
89.4	381.5	193.4	208.5
381.5	193.4	208.5	200.5
193.4	208.5	200.5	365.7
208.5	200.5	365.7	165.6
200.5	365.7	165.6	183.8
365.7	165.6	183.8	101
165.6	183.8	101	24.2
183.8	101	24.2	0.5
101	24.2	0.5	24
24.2	0.5	24	234.5
0.5	24	234.5	318.2
24	234.5	318.2	271.1
234.5	318.2	271.1	353.3
318.2	271.1	353.3	557.1

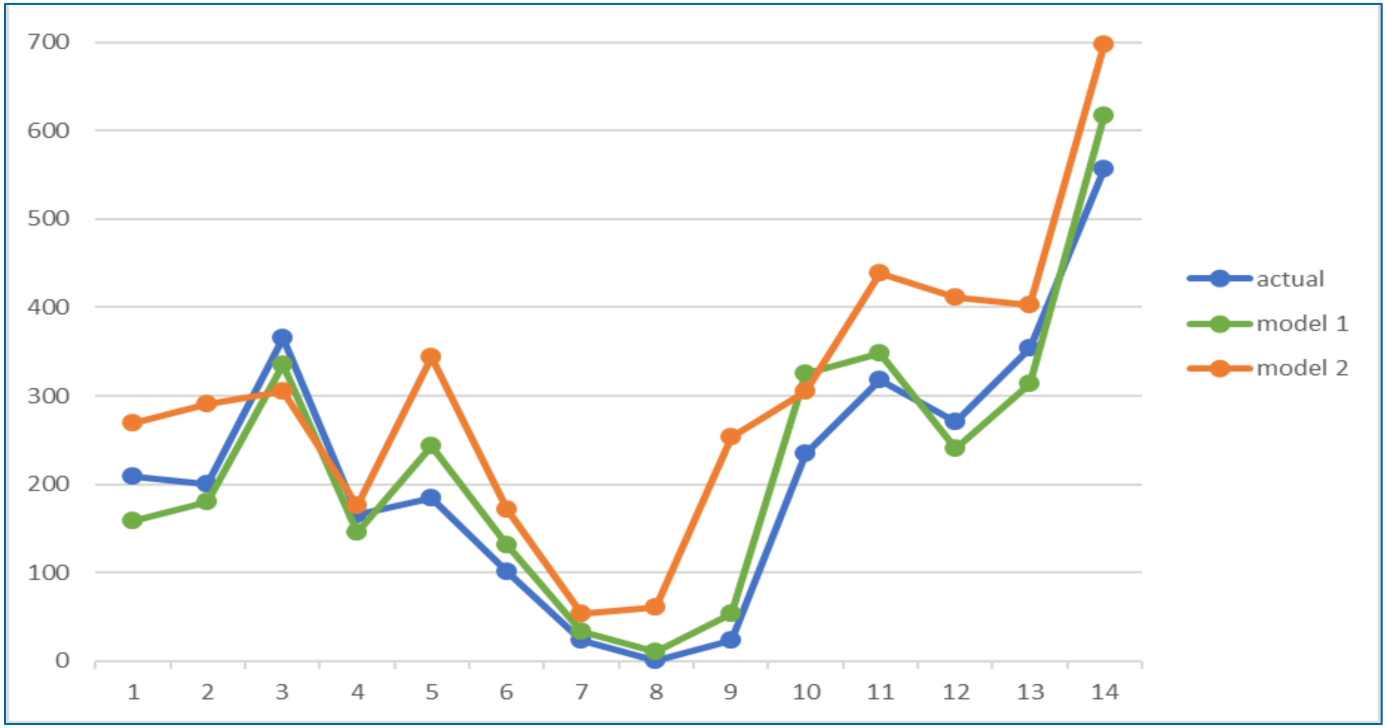
Time Series Prediction

- From those two model, the prediction results are as follow
- Which model is better?
- Let’s try to visualize it

t-3	t-2	t-1	actual	model 1	model 2
89.4	381.5	193.4	208.5	159	269
381.5	193.4	208.5	200.5	181	291
193.4	208.5	200.5	365.7	336	306
208.5	200.5	365.7	165.6	146	176
200.5	365.7	165.6	183.8	244	344
365.7	165.6	183.8	101	131	171
165.6	183.8	101	24.2	34.2	54.2
183.8	101	24.2	0.5	10.5	60.5
101	24.2	0.5	24	54	254
24.2	0.5	24	234.5	325	305
0.5	24	234.5	318.2	348	438
24	234.5	318.2	271.1	241	411
234.5	318.2	271.1	353.3	313	403
318.2	271.1	353.3	557.1	617	697

Time Series Prediction

- Which model is better?



- Visualization makes data easier to understand

Error Regression

- Use Mean Squared Error (MSE) to automatically measures the average squared difference between the estimated values and the actual value

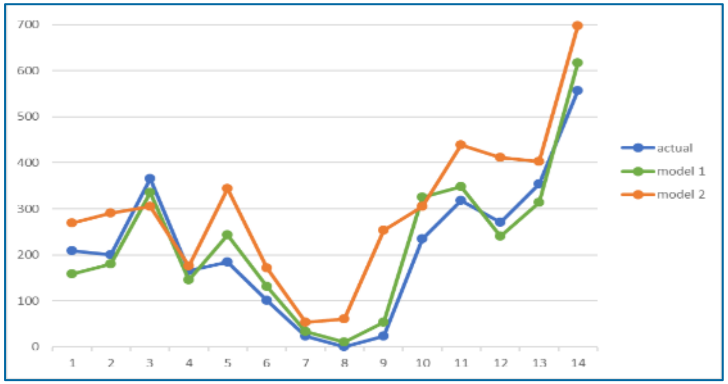
$$\boxed{\mathrm{MSE} = \frac{1}{n}\sum_{i-1}^{n}\left(Y_i - \widehat{Y}_i\right)^2}$$

- a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.

actual	model 1	model 2
208.5	159	269
200.5	181	291
365.7	336	306
165.6	146	176
183.8	244	344
101	131	171
24.2	34.2	54.2
0.5	10.5	60.5
24	54	254
234.5	325	305
318.2	348	438
271.1	241	411
353.3	313	403
557.1	617	697

Time Series Prediction

$$\mathrm{MSE} = \frac{1}{n}\sum_{i-1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$$



actual	model 1	error
208.5	159	50
200.5	181	20
365.7	336	30
165.6	146	20
183.8	244	-60
101	131	-30
24.2	34.2	-10
0.5	10.5	-10
24	54	-30
234.5	325	-90
318.2	348	-30
271.1	241	30
353.3	313	40
557.1	617	-60
MSE		1,179

actual	model 2	error
208.5	269	-60
200.5	291	-90
365.7	306	60
165.6	176	-10
183.8	344	-160
101	171	-70
24.2	54.2	-30
0.5	60.5	-60
24	254	-230
234.5	305	-70
318.2	438	-120
271.1	411	-140
353.3	403	-50
557.1	697	-140
MSE		11,736

Error Regression Variant

- Mean Squared Error (**MSE**)
  - Also called L2 Norm of the difference
- Root Mean Squared Error (**RMSE**)
  - Also called standard error
- Mean Absolute Error (**MAE**)
  - Also called L1 Norm of the difference
- Mean Absolute Percentage Error (**MAPE**)
  - Express the difference as percentage

$$\text{MSE} = \frac{1}{n}\sum_{i-1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$$

$$\text{RMSE} = \sqrt{\text{MSE}}$$

$$\text{MAE} = \frac{1}{n}\sum_{i-1}^{n}\left|Y_i - \widehat{Y}_i\right|$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i-1}^{n}\left|\frac{Y_i - \widehat{Y}_i}{Y_i}\right|$$

Problems with MSE/RMSE

- Closer to zero are better, but how well depends on the case
  - RMSE = 100 in IDR exchange rate predictions might be good, but RMSE = 100 in USD exchange rate prediction is bad
- Does not describe average error alone and has other implications that are more difficult to tease out and understand
- However, it avoids the use of taking the absolute value, which is undesirable in many mathematical calculations

Problems with MAPE

- Data with zero values cause division by zero
- For forecasts which are too low the percentage error cannot exceed 100%, but for forecasts which are too high there is no upper limit to the percentage error

- However, it has a quite intuitive interpretation in terms of relative error.

Classification

Binary Classification

- Let’s say you want to build a classifier to determine if someone is addicted to smartphones based on the length and frequency of their daily smartphone usage

- To train the model, you first collect data by surveying several of your friends

Binary Classification

- From **100** people you surveyed, you determine that
  - 52 people labeled as addicted, and
  - 48 people labeled as not addicted

A	B	C	D	E	F	G	H	I	addicted
8.4	7.5	5.8	31	4.0	47	34	47.9	0.9	YES
1.6	6.7	16.4	32	2.1	37	27	26.2	0.3	YES
2.5	0.4	2.1	47	2.3	65	54	37.2	0.4	NO
2.2	2.3	7.4	38	1.9	17	26	33.9	1.2	YES
4.1	11.6	12.1	54	3.6	55	33	34.4	0.0	NO
2.3	5.2	1.1	31	0.2	7	5	9.3	0.7	NO
1.3	3.3	5.1	75	9.9	66	48	43.8	1.9	NO
4.3	3.2	3.4	72	6.3	57	61	84.0	4.8	YES
3.0	5.2	5.2	53	3.3	27	37	67.0	1.8	YES
...	...	...	...	...	...	...	...	...	...

Binary Classification

- Now using all that data, you train **3 classifier** models
- See the prediction on the next slide
- And try to determine which model is better

Binary Classification

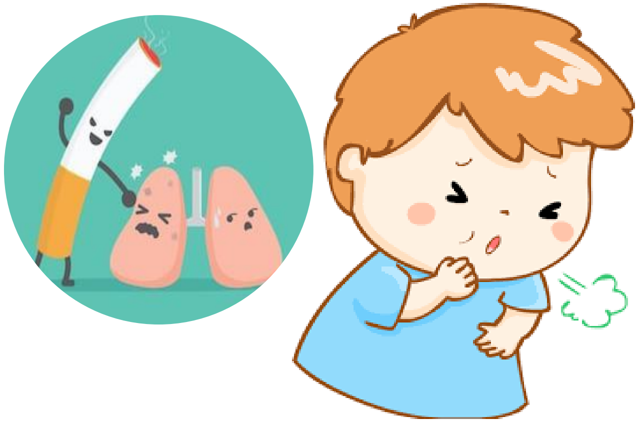
- Model 1:
  - 43 people from 52 addicted people and 1 person from 48 non-addicted are classified as addicted
- Model 2:
  - 51 people from 52 addicted people and 9 person from 48 non-addicted are classified as addicted
- Model 3:
  - 47 people from 52 addicted people and 5 person from 48 non-addicted are classified as addicted

Accuracy

- Ratio of correctly labeled (correctly predicted) compared to all data
- Model 1 $= \frac{43+47}{52+48} 100\% = \mathbf{90\%}$
- Model 2 $= \frac{51+39}{52+48} 100\% = \mathbf{90\%}$
- Model 3 $= \frac{47+43}{52+48} 100\% = \mathbf{90\%}$
- At this point, you might say that all models are equally good. Let's see another example

Binary Classification

- Now let's say you want to build a classifier to determine whether someone has a risk of getting lung cancer from the radiology images
- But turns out, it's quite hard to find the data
- Out of **100** data you collected, only 10 of them are having cancer



Binary Classification

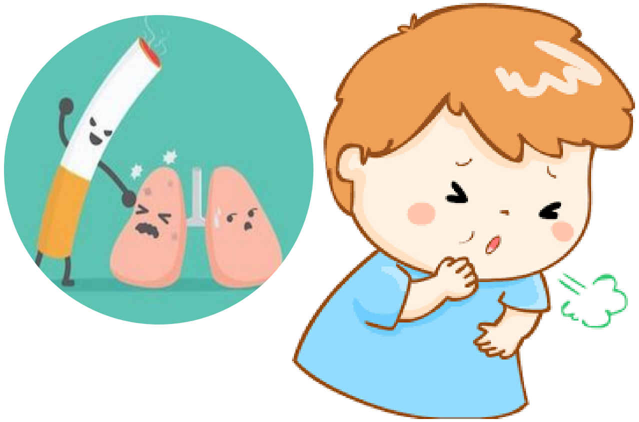
- But then you get on with it, and train another **3 classifiers**
- See the prediction on the next slide
- Again, try to determine which model is better

Binary Classification

- Model 1:
  - 1 from 10 cancer data are classified as having cancer
- Model 2:
  - 9 from 10 cancer data and 8 from 90 non-cancer data are classified as having cancer
- Model 3:
  - 5 from 10 cancer data and 4 from 90 non-cancer data are classified as having cancer

Accuracy Paradox

- If we compare them using Accuracy
- Model 1 = $\frac{1+90}{10+90}100\% = \mathbf{91\%}$
- Model 2 = $\frac{9+82}{10+90}100\% = \mathbf{91\%}$
- Model 3 = $\frac{5+86}{10+90}100\% = \mathbf{91\%}$
- Are these assessments correct?
- Are they equally good? Or equally bad?



Positive Class

Positive Class

- In medical testing, positive class usually means the presence of a condition, such as disease.

- In more general binary classification, however, positive class means the class that is deemed more important to be classified
  - What class you are interested
  - What class is more sensitive
  - What class you want your model to be active at

Positive Class

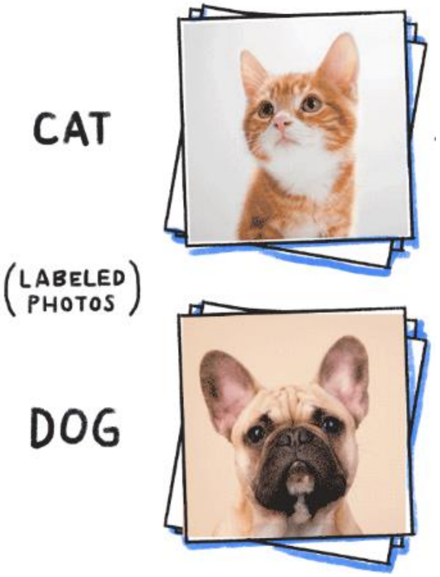
- From the previous example, we can decide that the addicted data are the positive class, and the non-addicted data as negative class
  - Since we're interested in finding who is addicted

- Similarly for the second example, we can say that the cancer data are the positive class, and the non-cancer data as negative class
  - Since the case (cancer) is more sensitive (important)

Positive Class

- However if both classes are equally important, then any of the classes can be the positive class, while the other class is the negative class

- For example, if you want to build a classifier to recognize **cats and dogs**, then either the cats or dogs class can be the positive class



Positive Class

- Positive class almost always labeled as 1,
  while negative class usually labeled either as 0 or -1
- In medical-related classification problem,
  the positive class is usually the disease class.
  - Classifying **cancer** → cancer is positive class
  - Classifying **pneumonia** → pneumonia is positive class
- But that is not always, depending on the case

Positive Class

- For example, in a healthy environment, any deadly disease case is the positive class
  - Since we want to immediately recognize the case and treat the person

- However in the case of zombie outbreak/apocalypse, then you might say that the normal person is the positive class

Positive Class

- In face detection problem, the face images are positive class, while any other non-face images are negative class
- In face spoofing detection, you can set that the fake-face images are positive class, while the real-face images are negative class
- Again, it depends on how you see the case

Positive Class

- Another thing that can be considered is the amount of data.
- If data is unbalanced,
  data with fewer number usually become the positive class.
- Example:
  - Churn prediction
  - Spam filtering

True Positive and True Negative

- For binary classification, a positive data that is correctly classified (recognized) as positive is called True Positive (TP)
  - A sick person is detected and recognized by the system

- While a negative data that is correctly rejected (classified as negative) is called True Negative (TN)
  - A healthy person is not classified as sick person

False Positive and False Negative

- Negative data that is incorrectly classified as Positive is called False Positive (FP) or False Alarm
  - A healthy person is detected as sick person

- Positive data that is incorrectly classified as Negative is called False Negative (FN) or Miss
  - A sick person is undetected by the system

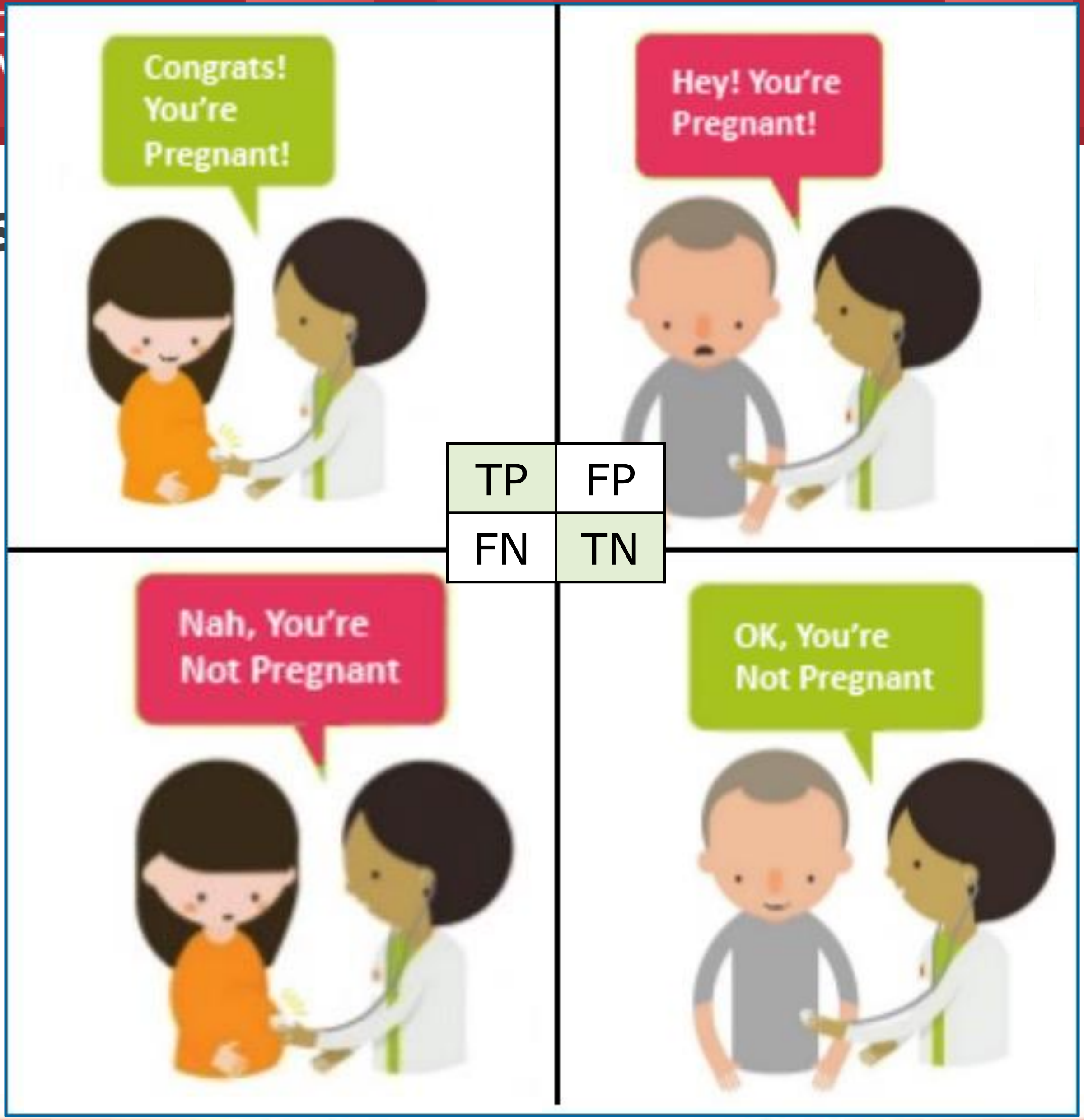
Confusion Matrix

Confusion Matrix

- Error Matrix
- Table layout to better visualize the performance of an algorithm

		Actual Class	
		+	-
Predicted Class	+	TP	FP
	-	FN	TN

Confus


Congrats! You're Pregnant!
Hey! You're Pregnant!
TP
FP
FN
TN
Nah, You're Not Pregnant
OK, You're Not Pregnant

Confusion Matrix

- If we look at 2nd example, we get confusion matrices as follow
  - Model 1: 1 cancer data classified as cancer
  - Model 2: 9 cancer and 8 non-cancer data classified as cancer
  - Model 3: 5 cancer data and 4 non-cancer data classified as cancer

Model 1		actual +	actual -	
pred	+	1	0	1
pred	-	9	90	99
		10	90	

Model 2		actual +	actual -	
pred	+	9	8	17
pred	-	1	82	83
		10	90	

Model 3		actual +	actual -	
pred	+	5	4	9
pred	-	5	86	91
		10	90	

Performance Metrics

- Accuracy
  - Ratio of correctly classified data compared to all data

$$\mathrm{ACC} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Error Rate} = 1 - \mathrm{ACC}$$

		actual	
		+	-
pred	+	TP	FP
	-	FN	TN

Performance Metrics

- True Positive Rate (TPR)
  - Also called Recall, or Sensitivity
  - Ratio of the correctly classified data as positive compared to all existing positive data

$$\boxed{\text{Recall} = \frac{TP}{TP + FN}}$$

		actual	
		+	-
pred	+	TP	FP
	-	FN	TN

  - How much positive data is recognized

Performance Metrics

- True Negative Rate
  - Also called Specificity
  - Ratio of the correctly classified data as negative compared to all existing negative data

$$\boxed{\text{Specificity} = \frac{TN}{TN + FP}}$$

  - How much negative data is recognized

		actual	
		+	-
pred	+	TP	FP
	-	FN	TN

Performance Metrics

- Precision
  - Ratio of the correctly classified data as positive compared to all positive prediction

$$\boxed{\text{Precision} = \frac{TP}{TP + FP}}$$

  - How much positive prediction is correct

		actual	
		+	-
pred	+	TP	FP
	-	FN	TN

Performance Metrics

- F1-Measure
  - Also called F1 Score, F Score, F Measure
  - Performance metric by considering both the precision and the recall

$$\boxed{F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}}$$

  - Focus on positive class and positive prediction

		actual	
		+	-
pred	+	TP	FP
	-	FN	TN

Performance Metrics

- False Positive Rate $= \frac{FP}{TN+FP}$
  - Ratio of the incorrectly labeled data as **positive** compared to all **negative data**
- False Negative Rate $= \frac{FN}{TP+FN}$
  - Ratio of the incorrectly labeled as **negative** compared to all **positive data**

		actual	
		+	-
pred	+	TP	FP
	-	FN	TN

Confusion Matrix

- Back to our 2nd example, we get each performance metric as

Model 1		actual		
		+	-	
pred	+	1	0	1
	-	9	90	99
		10	90	

Model 2		actual		
		+	-	
pred	+	9	8	17
	-	1	82	83
		10	90	

Model 3		actual		
		+	-	
pred	+	5	4	9
	-	5	86	91
		10	90	

	ACC	Recall	Precision	F1-Score
Model 1	91%	10%	100%	18.2%
Model 2	91%	90%	52.6%	66.7%
Model 3	91%	50%	55.6%	52.6%

Multiclass Classification

Multiclass Classification

- Let's say you want to build an image classifier to recognize three classes: cats, dogs, and horses
- Now you've collected several images, and you get
  - 100 cat images,
  - 120 dog images, and
  - 30 horse images

- Then you train the classifier

Multiclass Classification

- From the trained classifier, you get
- Out of 100 cat images, 82 classified as cats, 13 classified as dogs, and 5 classified as horses
- Out of 120 dog images, 91 classified as dogs, 5 classified as cats and 24 classified as horses
- Out of 30 horse images, 11 classified as horses, 3 classified as cats, and 16 classified as dogs

Confusion Matrix for Multiclass

- If we draw the confusion matrix, we get

		Actual Class		
		Cat	Dog	Horse
Predicted Class	Cat	82	5	3
	Dog	13	91	16
	Horse	3	24	11
Total Data		100	120	30

- To measure its performance, we can calculate the accuracy as

$$\boxed{\mathrm{ACC} = \frac{82 + 91 + 11}{100 + 120 + 30} = \frac{184}{250} = 0.736}$$

Macro Precision, Recall, and F1

- We can further calculate the performance as macro average precision, recall, and f1-score
- Macro precision and recall are defined as average of each class precision and recall
- To do that, it's easier to see if we split the confusion matrix

cat		actual		
		+	-	
pred	+	82	8	90
	-	18	142	160
		100	150	

dog		actual		
		+	-	
pred	+	91	29	120
	-	29	101	130
		120	130	

horse		actual		
		+	-	
pred	+	11	29	40
	-	19	191	210
		30	220	

Macro Precision and Recall

$$
\begin{aligned}
\text{Prec}_{cat} &= \frac{82}{82+8} = 0.91 \\
\text{Prec}_{dog} &= \frac{91}{91+29} = 0.76 \\
\text{Prec}_{horse} &= \frac{11}{11+29} = 0.28 \\
\text{macro}-\text{Prec} &= 0.65
\end{aligned}
$$

$$
\begin{aligned}
\text{Rec}_{cat} &= \frac{82}{82+18} = 0.82 \\
\text{Rec}_{dog} &= \frac{91}{91+29} = 0.76 \\
\text{Rec}_{horse} &= \frac{11}{11+19} = 0.37 \\
\text{macro}-\text{Rec} &= 0.65
\end{aligned}
$$

cat		actual		
		+	-	
pred	+	82	8	90
	-	18	142	160
		100	150	

dog		actual		
		+	-	
pred	+	91	29	120
	-	29	101	130
		120	130	

horse		actual		
		+	-	
pred	+	11	29	40
	-	19	191	210
		30	220	

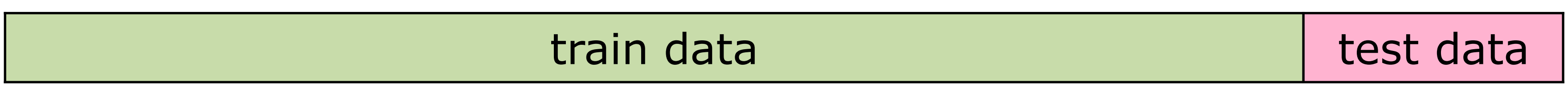
Model complexity, overfitting, and model selection

A Good Classifier

- Training accuracy is only useful for checking whether the learning process is running
- During the learning process, training accuracy is expected to increase during the iteration, until learning converges at one point
- High Training accuracy **DOES NOT** mean you achieve a good model

A Good Classifier

- The goal of classification is to perform well **on new (unseen) data.**
- That's why you split your dataset into two parts: **train data** and **test data**



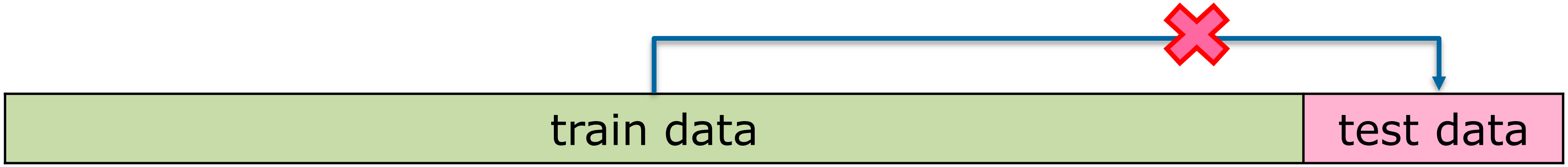
A Good Classifier

- Estimate the generalization error by using only part of the available data for **'training'** and leaving the rest for **'testing'.**
- The test data is now **'new data'**,
  so we can with this approach get unbiased estimates of the generalization error

train data	test data

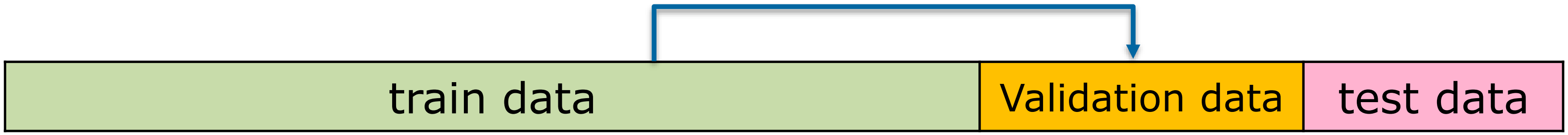
A Good Classifier

- However, it is **NOT RIGHT** for you to use test data during the hyperparameter observation
- Because then the test data is no longer "unseen"
- It might skew the result



A Good Classifier

- For that you need to further split your data set into train data, validation data, and test data
- Perform hyperparameter observation by evaluating the model on validation data



Types of Dataset

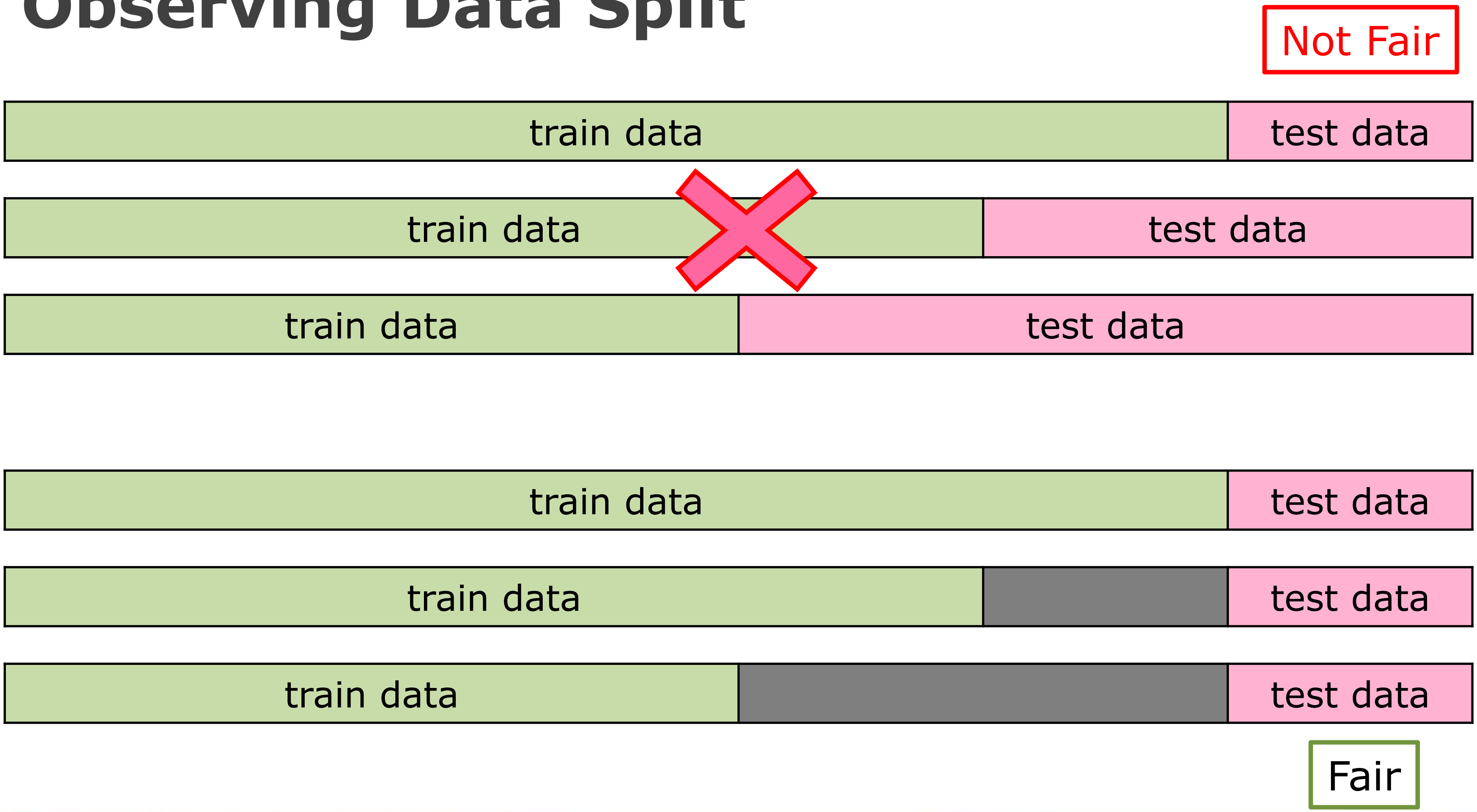
- Train set
  - Known Data Points (data label known)
  - Use to train the model

- Validation set (dev set)
  - Known data points treated as Unseen (data label known)
  - Use to tune (find the best) Hyperparameters
  - Use to check the model performance

Types of Dataset

- Test set
  - Proxy for generalization performance
  - **Should not use** while observation
  - **Forget you even have it**
  - Use only **very sparingly** at the end
    - To check how well the model in generalizing the unseen data point
    - Evaluate on the test set only a single time, at the very end

Observing Data Split



Using Validation

- Holdout Validation (Single Split)
- Cross Validation
- K-Fold Cross Validation

Single Validation Split (Holdout Validation)

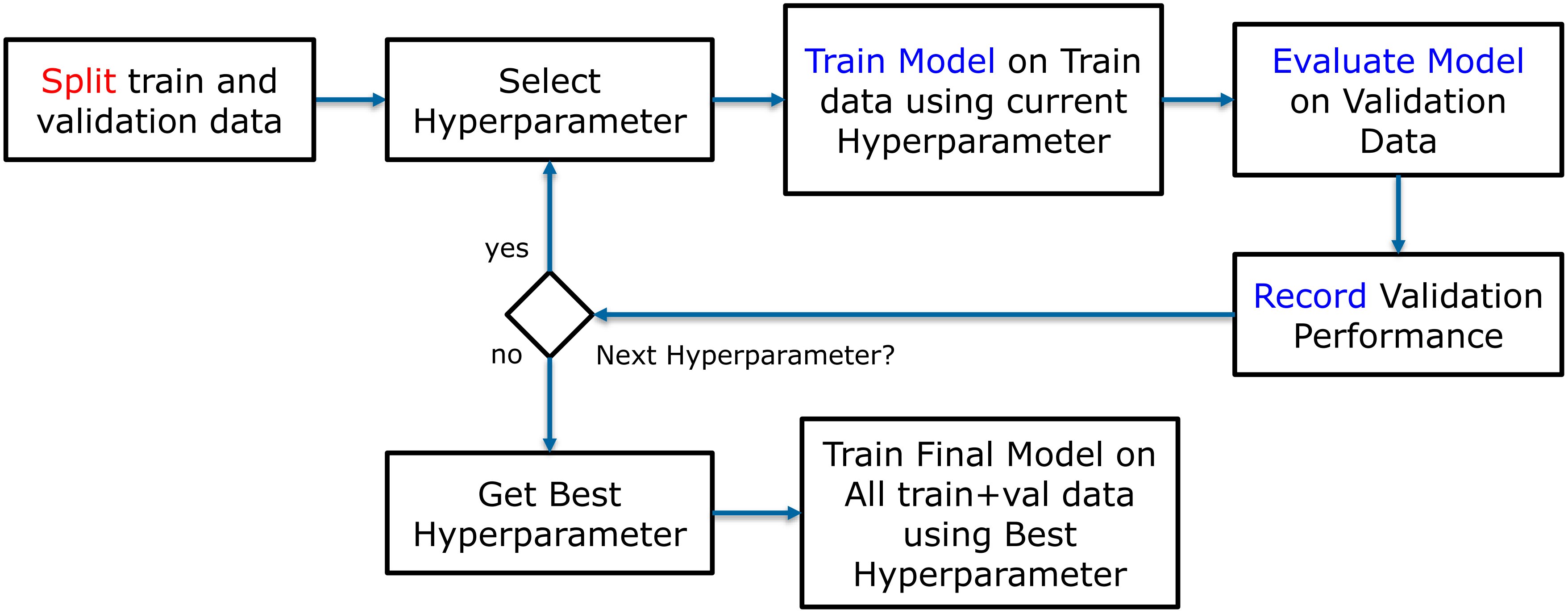
Holdout Validation

- Single Validation
  - Divide data into two : train set and validation set
  - Handpicked
  - Use to the rest of observation
- Data proportion
  - No definitive proportion between train and validation
  - It is usually that train set has bigger proportion, but it’s not necessarily has to be like that
  - Just make sure that the number is not too small or too big

Holdout Validation

- Class proportion
  - The important note is that the validation set should have similar class proportion to training set
  - Different distribution on class may skew the observation
  - Either that, or set that all classes in validation set has the same number of data points

Holdout: Hyperparameter Observation



Validation Split Proportion Note

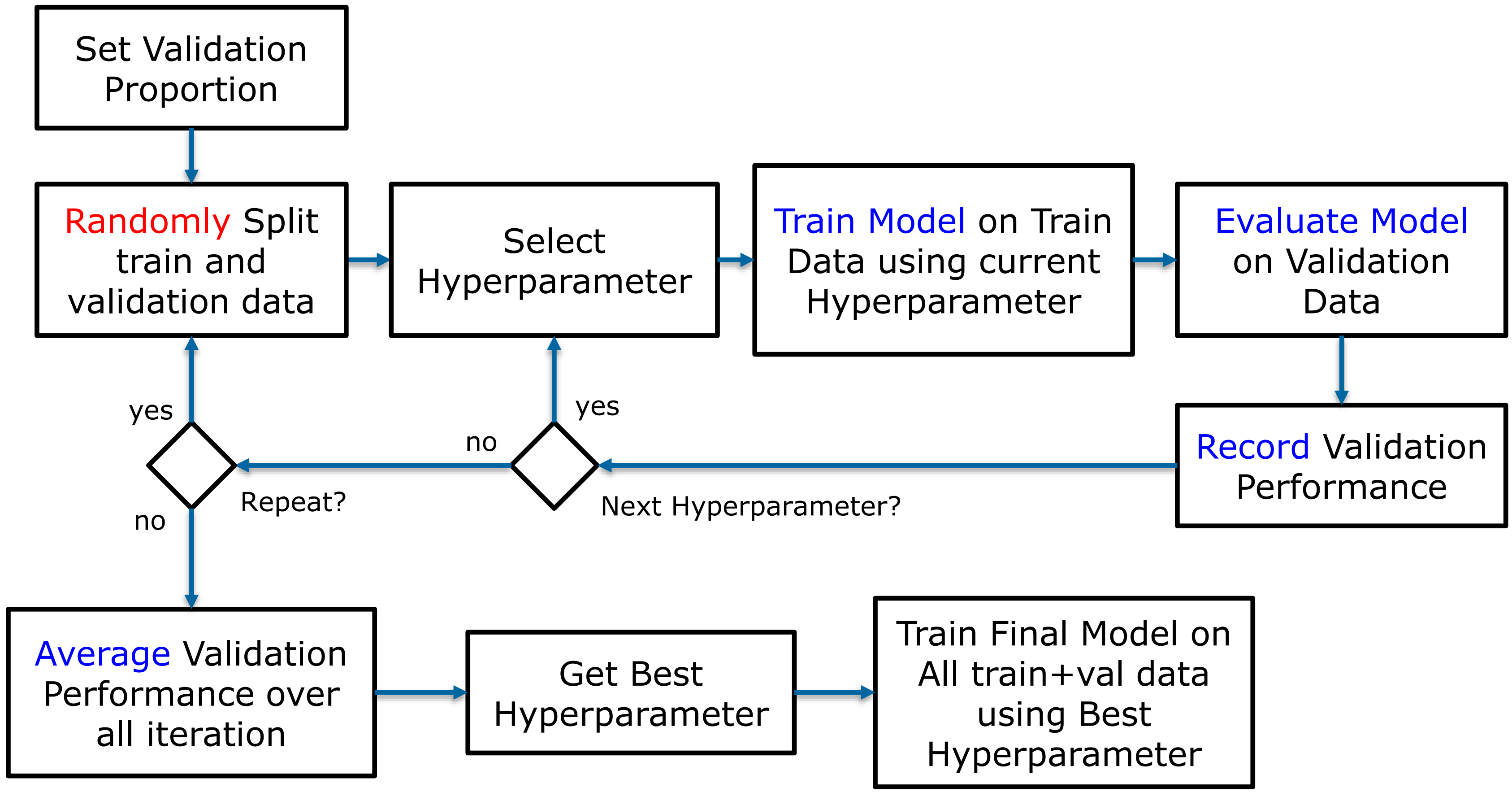
- Since validation accuracy is depend on the training data distribution,
- Too big of a validation split will make the training data distribution too sparse (as the training number is too little) and may decrease the accuracy
- Too small of a validation split may result the validation data does not represent the whole data point

Cross Validation

Random Cross Validation

- Problem with handpicked validation split is that the validation set may not represent the whole data point distribution
- Solution:
  Choose the validation data point randomly
- But adding stochasticity to the observation means that the process must be done multiple time
  - ~10-30 times

Cross Val: Hyperparameter Observation



K-Fold Cross Validation

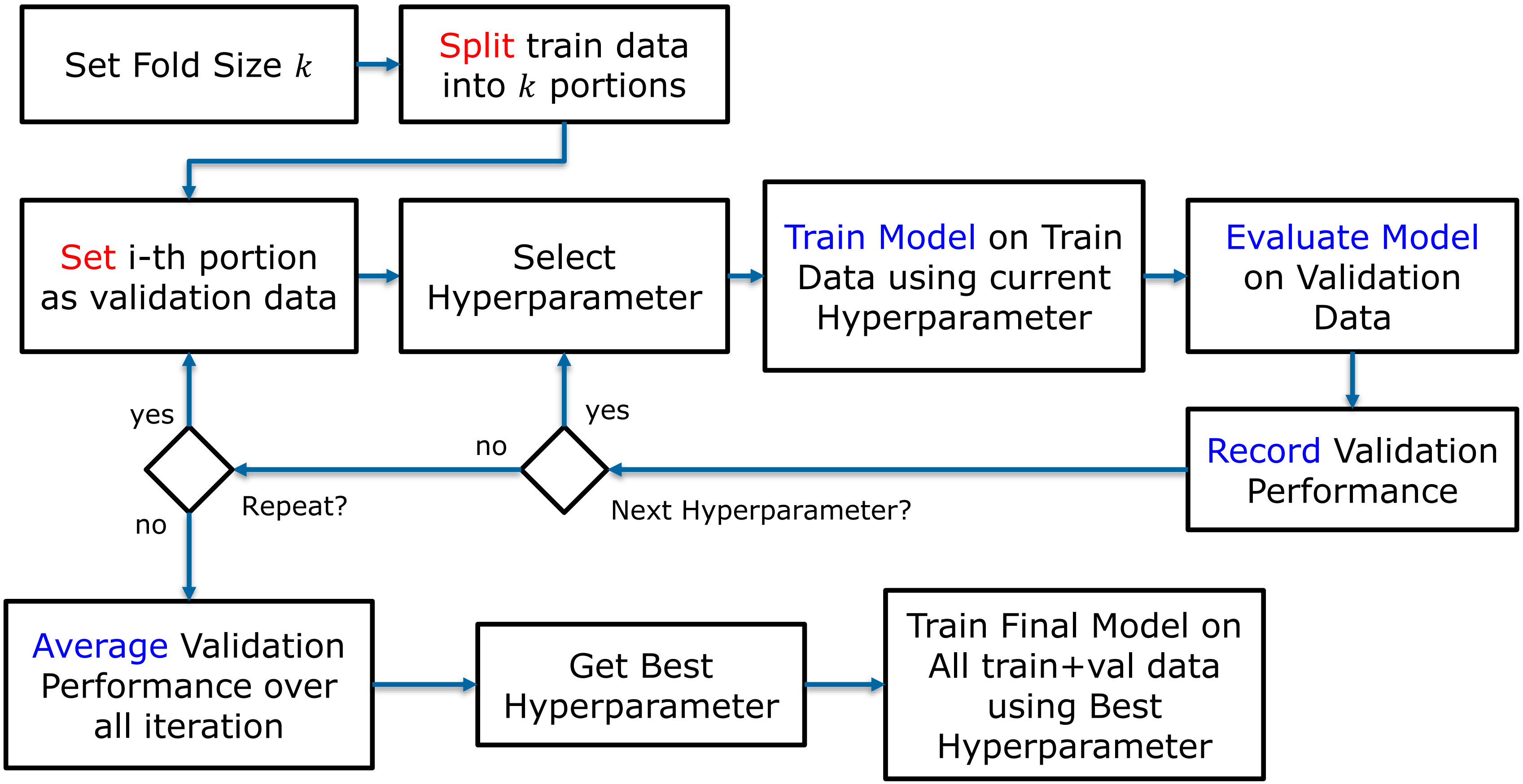
K-Fold Cross Validation

- Adding stochasticity to the observation means that the process must be done multiple time
- how to reduce this iterative process?

K-Fold Cross Validation

- Use fixed split validation rather than random
  - Divide data into k splits (portions)
  - Use the split alternately as validation set
- K-fold Cross Val will make validation process cover the whole data point with less iteration

Cross Val: Hyperparameter Observation



Example K-Fold Cross Validation

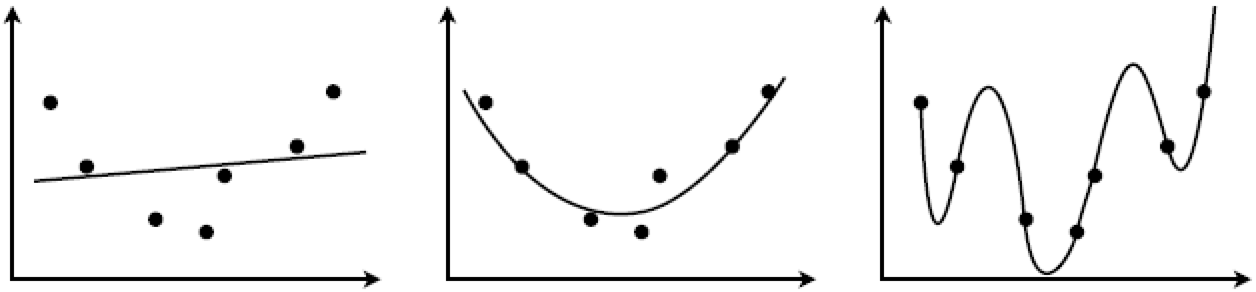
- Example K-Fold Cross Validation using 5 folds
  - Iteration 1
    - 1,2,3,4 train
    - Fold 5 validation
  - Iteration 2
    - 1,2,3,5 train
    - Fold 4 validation
  - Iteration 3
    - 1,2,4,5 train
    - Fold 3 validation
  - ...

fold 1	fold 2	fold 3	fold 4	fold 5
fold 1	fold 2	fold 3	fold 4	fold 5
fold 1	fold 2	fold 3	fold 4	fold 5

⋮

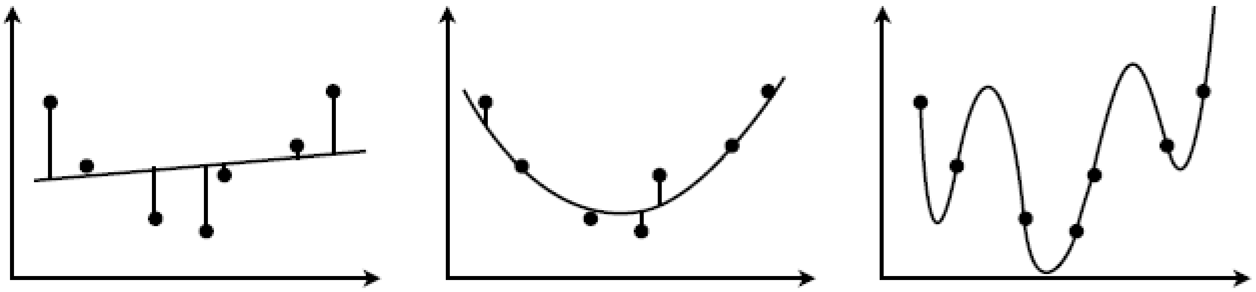
Error vs Flexibility (train and test)

- Which of the following curves 'fits best' to the data?



Error vs Flexibility (train and test)

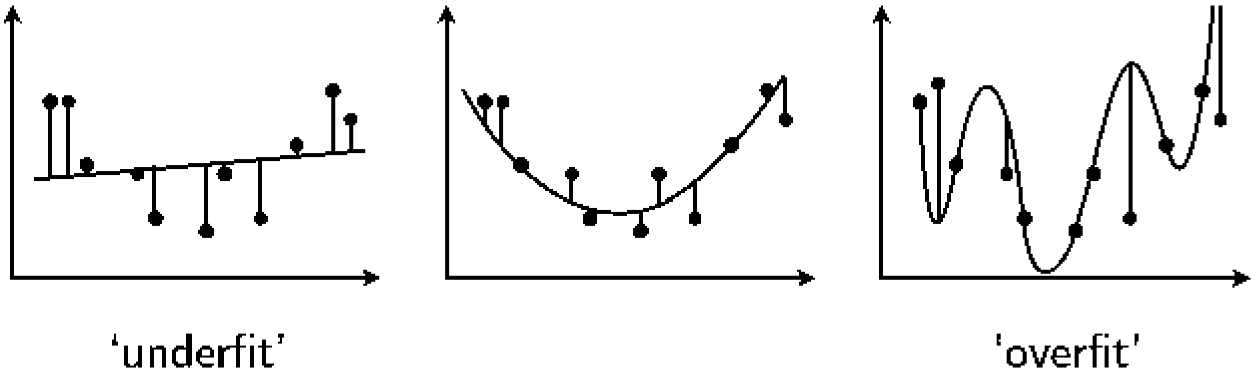
- Which of the following curves 'fits best' to the data?



- The more flexible the curve...
  - the better you can make it fit your data...

Error vs Flexibility (train and test)

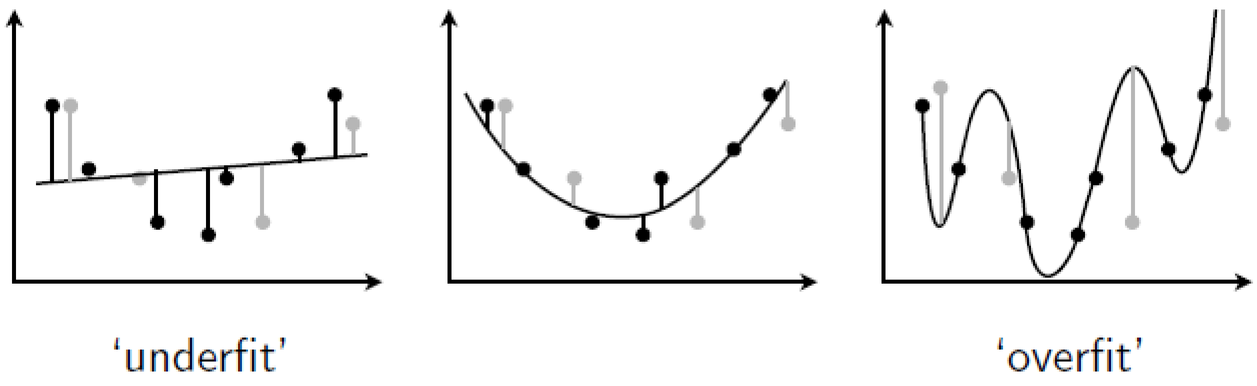
- Which of the following curves 'fits best' to the data?



- The more flexible the curve...
  - the better you can make it fit your data...
  - but the more likely it is to overfit

Error vs Flexibility (train and test)

- Which of the following curves 'fits best' to the data?



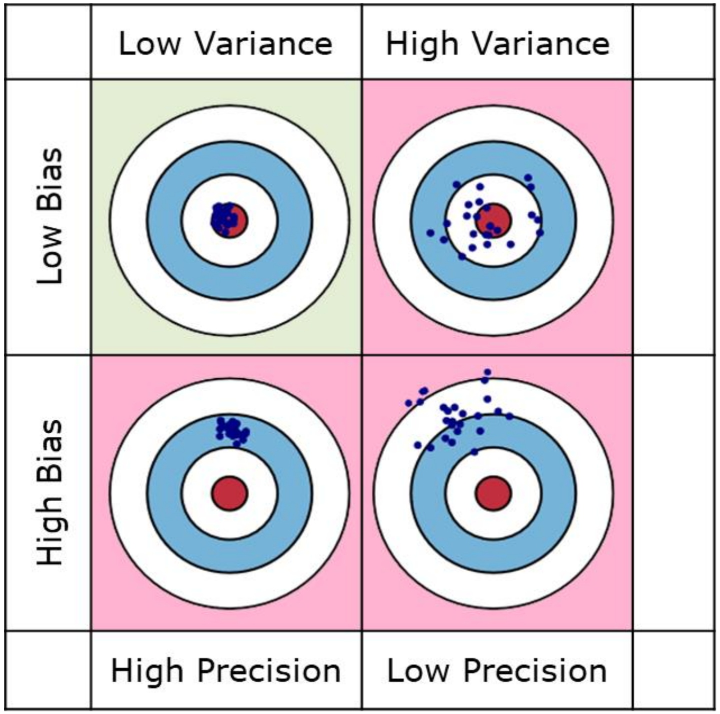
- So you need to be careful to strive for both model simplicity and for good fit to data!

Bias-variance tradeoff

- Based on N training datapoints from the distribution, how close is the learned classifier to the optimal classifier?
- Consider multiple trials: repeatedly and independently drawing N training points from the underlying distribution.
  - Bias: Difference between the optimal classifier and the average (over all trials) of the learned classifiers
  - Variance: Average squared difference from the (single-trial) learned classifier from the average (over all trials) of the learned classifiers

Bias-variance tradeoff

- Goal: Low bias and low variance.
- High model complexity
  ➔ low bias and high variance
- Low model complexity
  ➔ high bias and low variance



Question?

THANK YOU