Evaluation Measures & Dataset

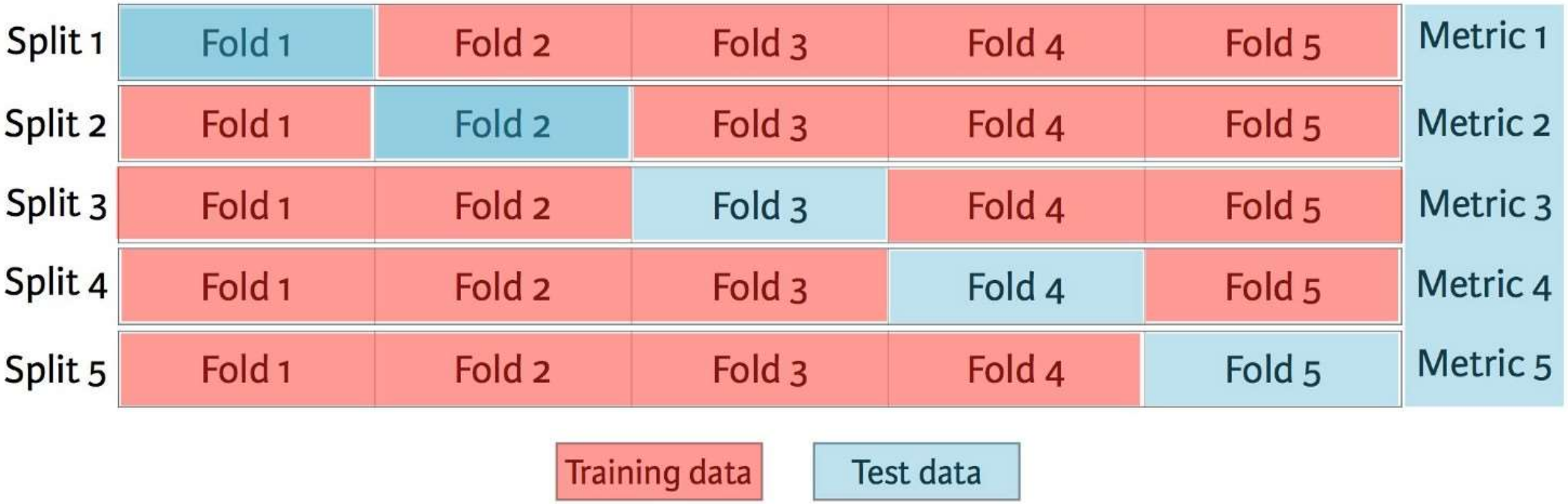
How to improve the model

More data may be required
Data needs to have more diversity
Algorithm needs longer training
More hidden layers or hidden units are required
Add Regularization
Change the Neural network architecture like activation function etc.
There are many other considerations you can think of..

Distribution of data for improving the accuracy of the model

Training set	Which you run your learning algorithm on.
Dev (development) set	Which you use to tune parameters, select features, and make other decisions regarding the learning algorithm. Sometimes also called the **hold-out cross validation set** .
Test set	which you use to evaluate the performance of the algorithm, but not to make any decisions regarding what learning algorithm or parameters to use.

K-Fold Cross Validation



- Not much used for Deep Learning.. Why?

Partition your data in different categories

Training Set	Dev Test or Hold Out Cross Validation Set	Test Set

Traditional Style partitioning 70/30 or 60/20/20

But in the era of Deep Learning may even go down to 99 0.5 0.5

If the data size is 1,00,0000 then 5000 5000 data size will still be there in dev and test sets

Importance of Choosing dev and test sets wisely

The purpose of the dev and test sets are to direct your team toward the most important changes to make to the machine learning system

Very important that dev and test set reflect data you expect to get in the future and want to do well on.

Bad distribution will severely restrict analysis to guess that why test data is not giving good results

Data Distribution Mismatch

It is naturally good to have the data in all the sets from the same distribution.

For example Housing data coming from Mumbai and we are trying to find the house prices in Chandigarh.

Else wasting a lot of time in improving the performance of dev set and then finding out that it is not working well for the test set.

Sometime we have only two partitioning of the data in that case they are called Train/dev or train/test set.

Using a single Evaluation Metric

You should be clear about what you are trying to achieve and what you are trying to tune

Classifier	Precision	Recall
A	95	90
B	98	85

Precision – of examples recognized as true how many are true

Recall – of total true examples how many have been correctly extracted

Remember: We are calculating these figures from the dev set

Precision Recall an Example

- Suppose a Cancer hospital has **200** patients.
- Out of them **100** have cancer.
- Our model retrieves **80** people and says that they have cancer
- Out of the **80** which model retrieved only 40 people have cancer

- Precision = 40/80 = 50%
- Recall = 40/100 = 40%

Using a single Evaluation Metric

You should be clear about what you are trying to achieve and what you are trying to tune

Classifier	Precision	Recall	F1 Score
A	95	90	92.4
B	98	85	91

F1 Score = Harmonic mean of Precision and Recall $\frac{2}{\frac{1}{P}+\frac{1}{R}}$

Optimize one parameter and satisfy others

Classifier	Accuracy	Running Time	Safety	False Positive	…	..
A	90	20ms	No			
B	92	80ms	Yes			
C	96	2000ms	Yes			

Maximize ????? Subject to ????? And ????? And…

So few of them can be satisficing metric.
e.g if we say that running time needs to be minimum 100ms that running time is satisficing metric and accuracy can be optimizing metric

Confusion Matrix

	Disease or Condition	No Disease or Condition
Test Positive	A True Positive	B False Positive
Test Negative	C False Negative	D True Negative

False Positive and False Negative

-Amazon Echo listening for “Alexa”; Apple Siri listening for “Hey Siri”; Android listening for “Okay Google”.

-False positive rate—the frequency with which the system wakes up even when no one said the wakeword—as well as the false negative rate—how often it fails to wake up when someone says the wakeword.

-One goal is to minimize the false negative rate (optimizing metric), subject to there being no more than one false positive every 24 hours of operation (satisficing metric).

Importance of Dev Set and Single Evaluation Metric

- If your team improves the classifier's accuracy from 95.0% to 95.1%, you might not be able to detect that 0.1% improvement from playing with the app.
- Having a dev set and metric allows you to very quickly detect which ideas are successfully giving you small (or large) improvements, and therefore lets you quickly decide what ideas to keep refining, and which ones to discard.

When Metric Evaluation may fail

- Suppose that for your cat application, your metric is classification accuracy. This metric currently ranks classifier A as superior to classifier B.
- But suppose you try out both algorithms, and find classifier A is allowing occasional pornographic images to slip through.
- Even though classifier A is more accurate, the bad impression left by the occasional pornographic image means its performance is unacceptable.
- Change the metric to heavily penalize letting through pornographic images.

Start Quickly Then Optimize

- ✓Don’t start off trying to design and build the perfect system.
- ✓Build and train a basic system quickly
- ✓It is valuable to examine how the basic system functions
- ✓Find clues that show you the most promising directions in which to invest your time.

Found few dogs as cats by classifier

Should you try to make your classifier work better on dogs?

- Depends
- Accuracy is 90%
- Error Analysis: Get 100 random mislabeled dev set examples
- Count how many dogs are there
- 5 -that means 5% and will only affect 0.5% of the error
- Try other things. Make a table. How many are blurry, big cats or others

Error Analysis

Image		Dog	Great cat		Blurry	Comments
	1	✔				Usual pitbull color
	2				✔	
	3		✔		✔	Lion; picture taken at zoo on rainy day
	4		✔			Panther behind tree
	...	...	...	...		...
% of total		8%	43%		61%	

Few more points

Whatever process you apply to fixing dev set labels, remember to apply it to the test set labels too so that your dev and test sets continue to be drawn from the same distribution

If you decide to improve the label quality, consider double-checking both the labels of examples that your system misclassified as well as labels of examples it correctly classified.

It is possible that both the original label and your learning algorithm were wrong on an example