COMPSCI 5100
ML & AI for Data Science

Ali Gooya

ali.gooya@glasgow.ac.uk

Detour...

Classification: Part III

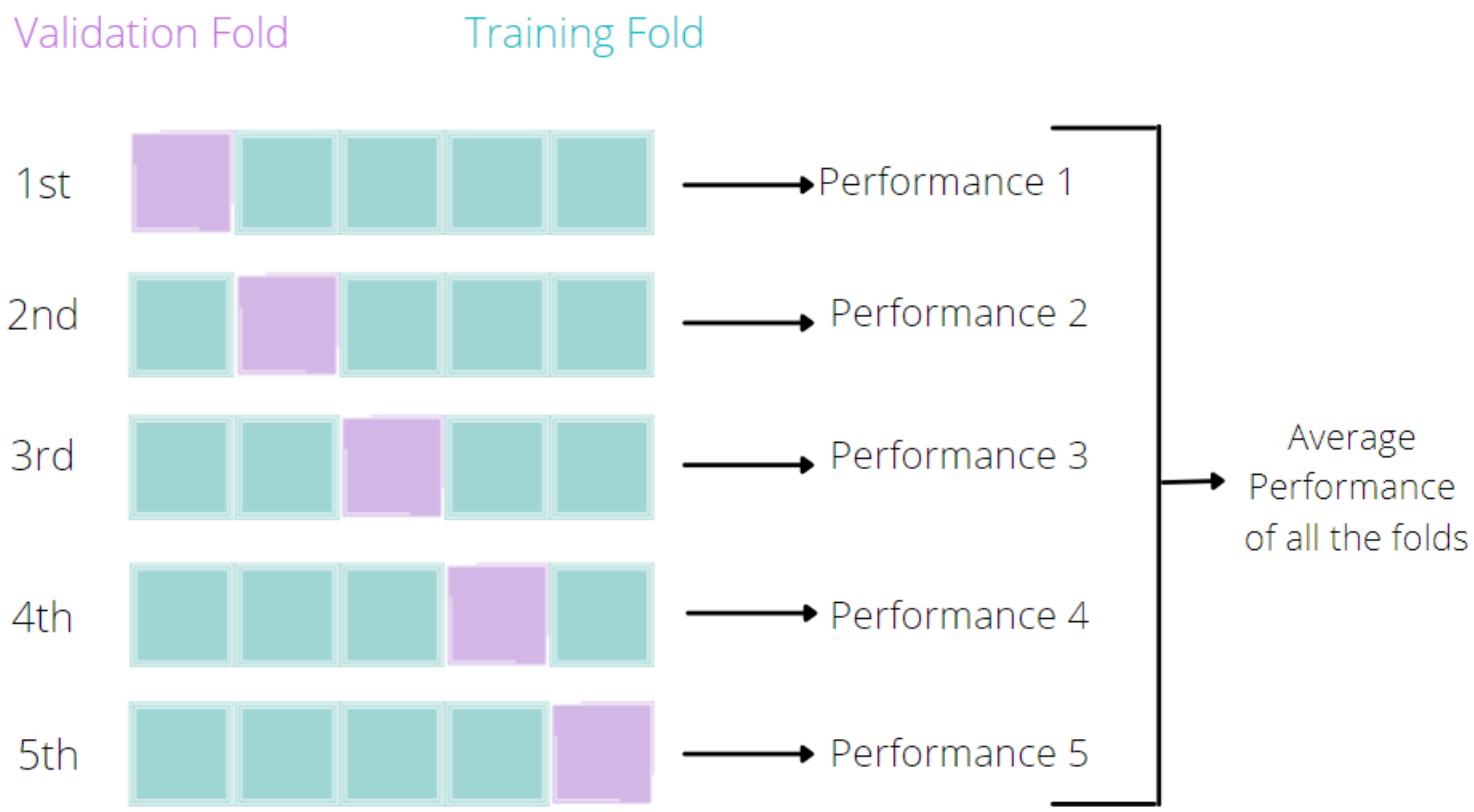
Performance evaluation

- **Evaluation strategy**
  - How to split data for training and testing
- **Evaluation metrics**
  - How to measure performance accurately
- **Benchmarking**
  - Compare results against other ‘known’ results

image from: pub.towardsai.net

Evaluation strategy

- **Cross validation**



Evaluation strategy

- **Leave one subject out**
  - Particularly useful for classification tasks involving human-centric data

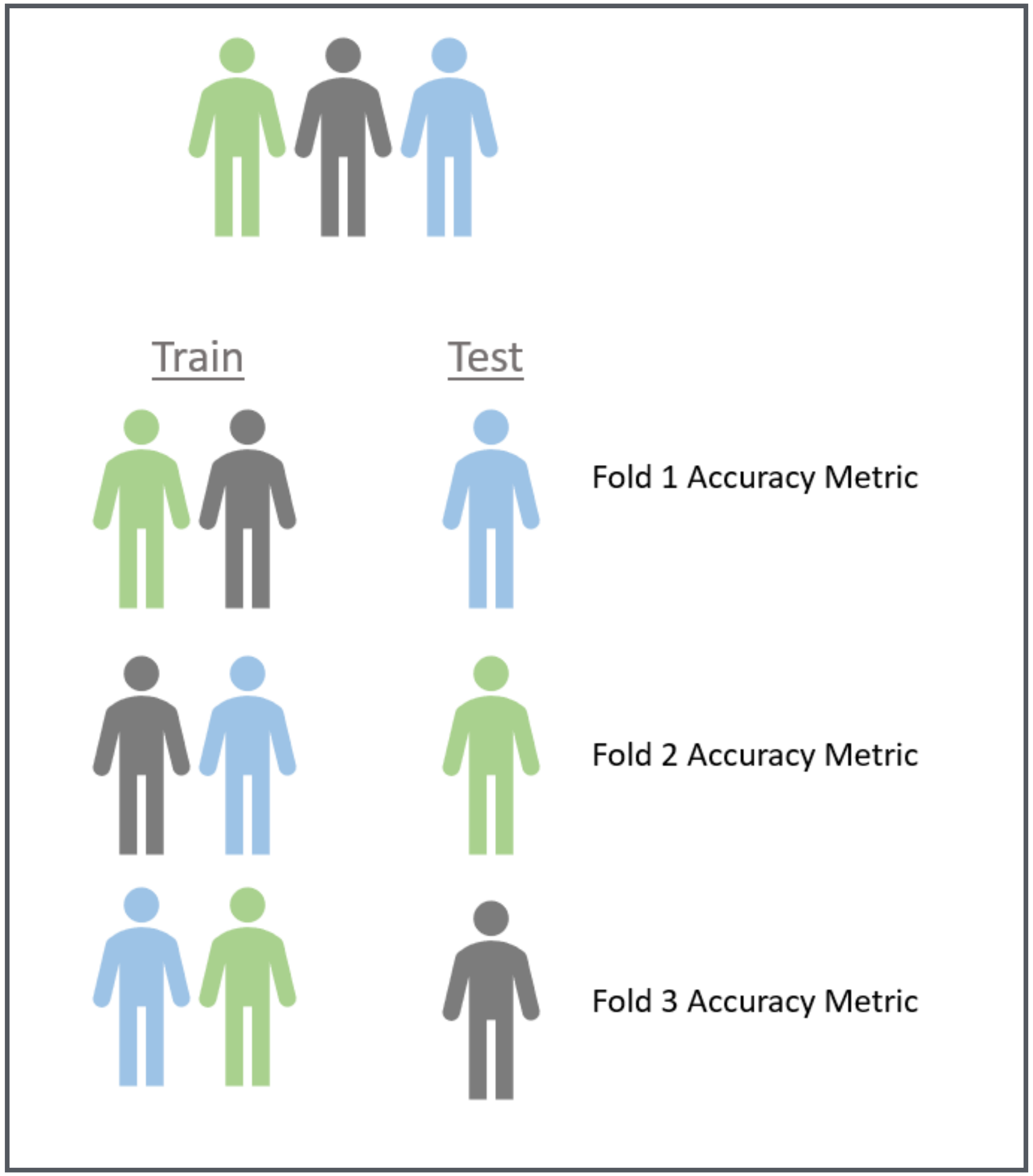


image from: medium.com

Evaluation strategy

- **Random train:test splits**
  - Particularly useful for very large datasets
  - CV may be difficult
  - Randomly choose 70-80% data for training and rest for testing

Performance metrics

- Performance **metrics** are important to
  - Compare performances of multiple classifiers
  - Compare performance of the same classifier under different conditions
  - Tune hyperparameters

- No metric is perfect; each gives you some insights

- Practical tip: **Use multiple evaluation metrics**

Accuracy

- $$\textbf{Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Total number of test samples}}$$
- Often expressed in %
- Simple, intuitive, widely used

Disadvantage: Doesn't take into account class imbalance:

- We're building a classifier to detect a rare disease.
- Assume only 1% of population is diseased.
- Diseased: $t = 1$
- Healthy: $t = 0$
- What if we always predict healthy? $(t = 0)$
- Accuracy 99%
- But classifier is rubbish!

[Content from Dr. Ke Yuan's slide]

Weighted accuracy (WA): Accuracies computed per class, averaged across all classes

Confusion matrix

		True	
		1	0
Predicted	1	True Positives (TP)	False Positives (FP)
	0	False Negatives (FN)	True Negatives (TN)

True class

Predicted class	...	10	11	12	13	14	15	16	18	18	19	20
1	...	4	2	0	2	10	4	7	1	12	7	47
2	...	0	0	4	18	7	8	2	0	1	1	3
3	...	0	0	1	0	1	0	1	0	0	0	0
4	...	1	0	1	28	3	0	0	0	0	0	0
⋮						⋮						
16	...	3	2	2	5	17	4	376	3	7	2	**68**
17	...	1	0	9	0	3	1	3	325	3	**95**	19
18	...	2	1	0	2	6	2	1	2	325	4	5
19	...	8	4	8	0	10	21	1	16	19	**185**	7
20	...	0	0	1	0	1	1	2	4	0	1	**92**

- Algorithm is getting ‘confused’ between classes 20 and 16, and 19 and 17.
  - 17: talk.politics.guns
  - 19: talk.politics.misc
  - 16: talk.religion.misc
  - 20: soc.religion.christian
- Maybe these should be just one class?
- Maybe we need more data in these classes?
- Confusion matrix helps us direct our efforts to improving the classifier.

[Content from Dr. Ke Yuan’s slide]

Precision

Predicted \ True	1	0
1	True Positives (TP)	False Positives (FP)
0	False Negatives (FN)	True Negatives (TN)

Example:

1: Diseased 0: Healthy

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP+FP}}$$

- Among all people classified as ‘diseased’, how many are actually diseased
- Perfect precision = no FP
- Higher the better

Recall or Sensitivity

Predicted \ True	1	0
1	True Positives (TP)	False Positives (FP)
0	False Negatives (FN)	True Negatives (TN)

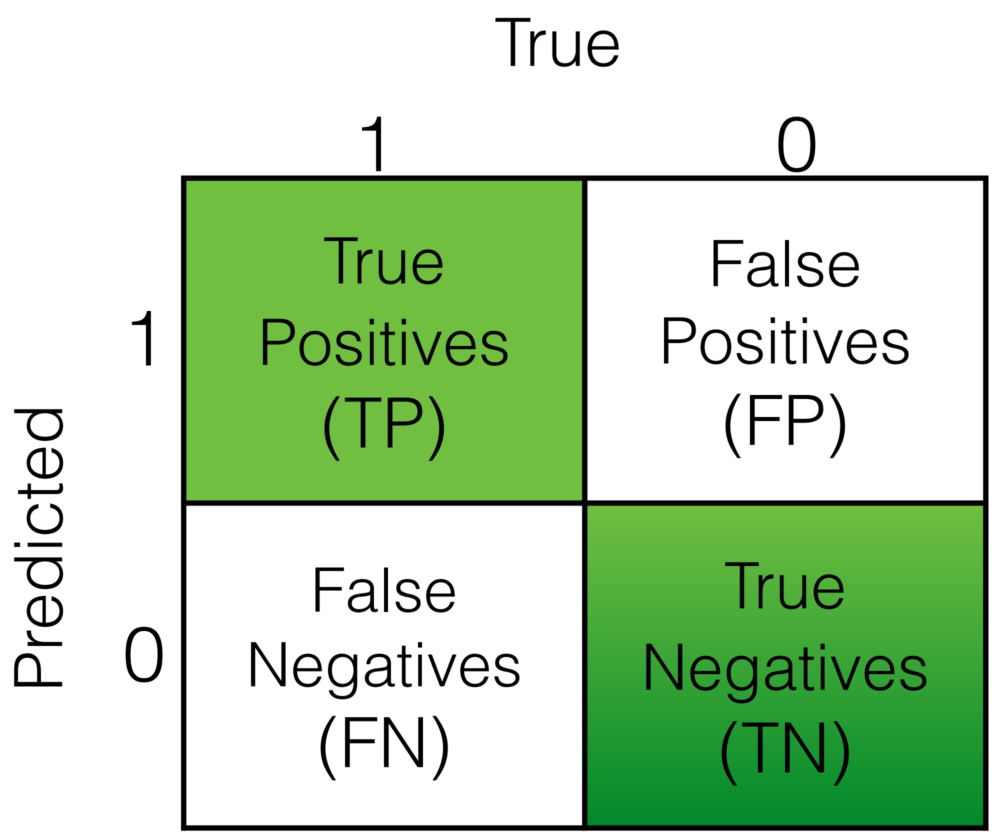
Example:

1: Diseased   0: Healthy

$$\textbf{Sensitivity } S_e = \frac{\textbf{TP}}{\textbf{TP+FN}}$$

- Among all diseased people, how many are correctly identified
- Sensitivity = recall
- Perfect recall = no FN
- Higher the better

Specificity



Example:

1: Diseased   0: Healthy

Specificity $S_p = \frac{\textbf{TN}}{\textbf{TN+FP}}$

- Among all healthy people, how many are classified as healthy.
- Higher the better

Optimizing sensitivity and specificity

- We would like both to be as high as possible.
- Often increasing one will decrease the other.
- Balance will depend on application:
- e.g. diagnosis:
  - We can probably tolerate a decrease in specificity (healthy people diagnosed as diseased)....
  - ...if it gives us an increase in sensitivity (getting diseased people right).

[Slide courtesy: Dr. Ke Yuan]

ROC

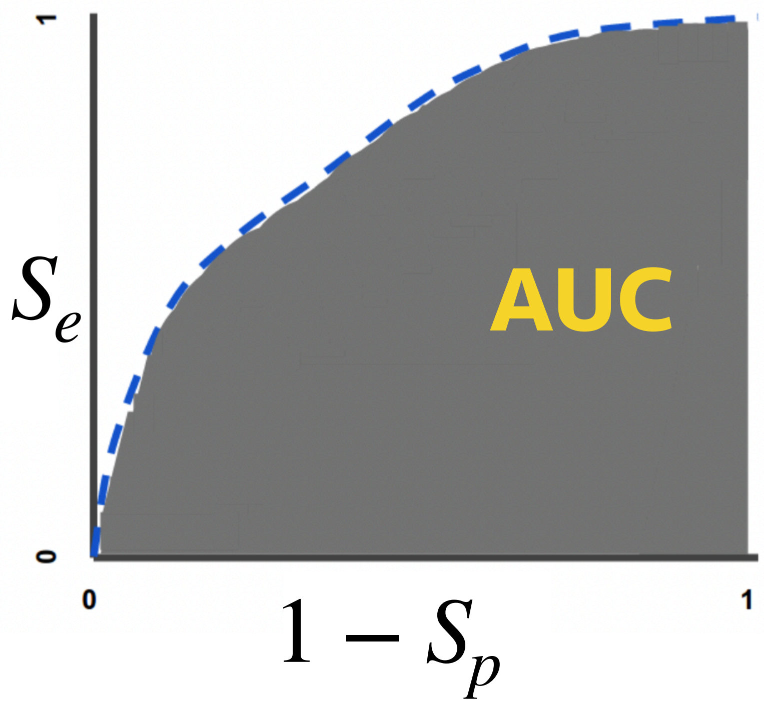
- Many classification algorithms involve setting a threshold.
- e.g. Logistic Regression:

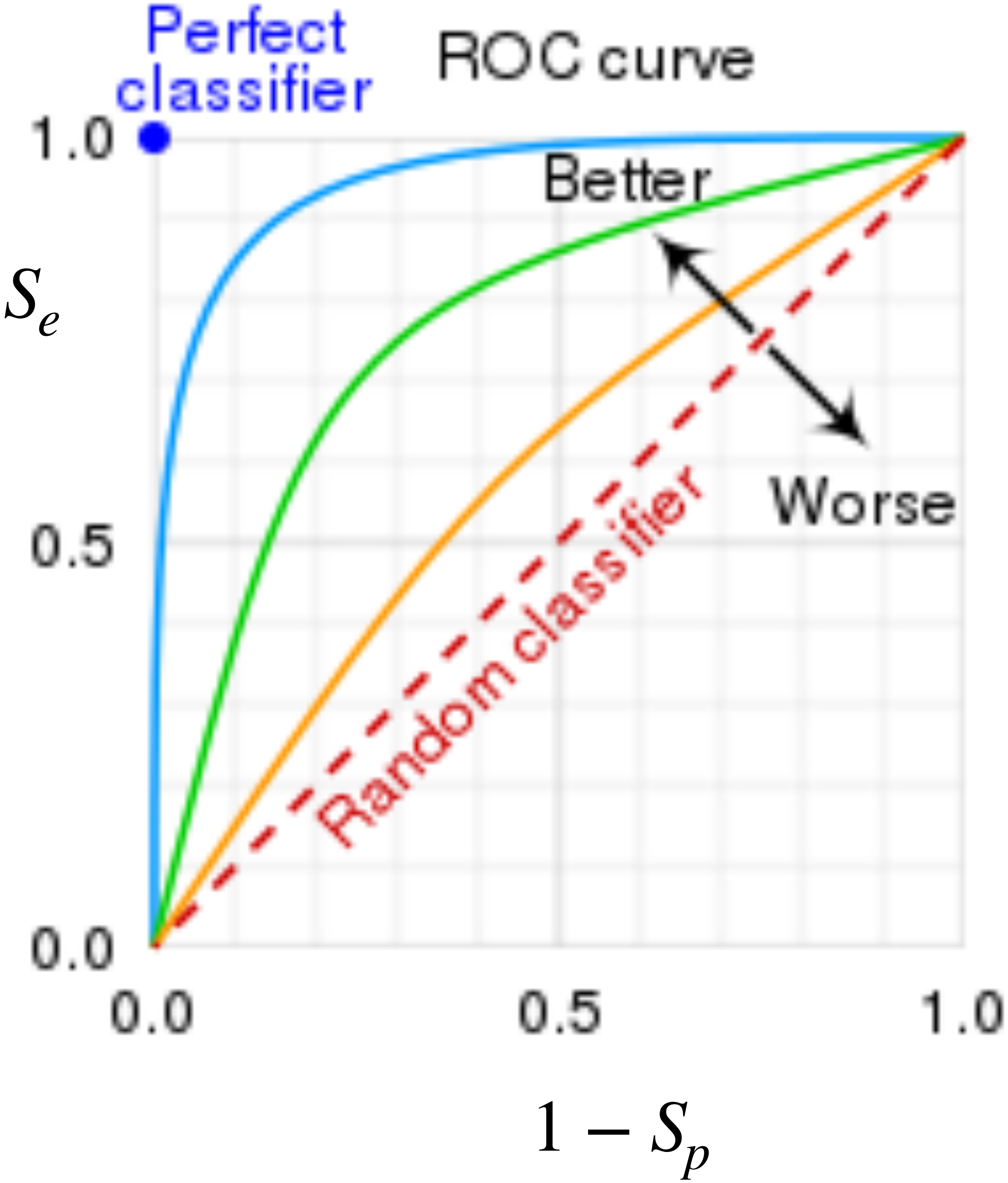
$$p(t_{new} = 1 | \mathbf{x}_{new}, \mathbf{w}) > 0.5$$

- However, we could use any threshold we like....
- The *Receiver Operating Characteristic (ROC) curve* shows how S_e and $1 - S_p$ vary as the threshold changes.

[Slide courtesy: Dr. Ke Yuan]

ROC, AUC





[Image from Wikipedia]

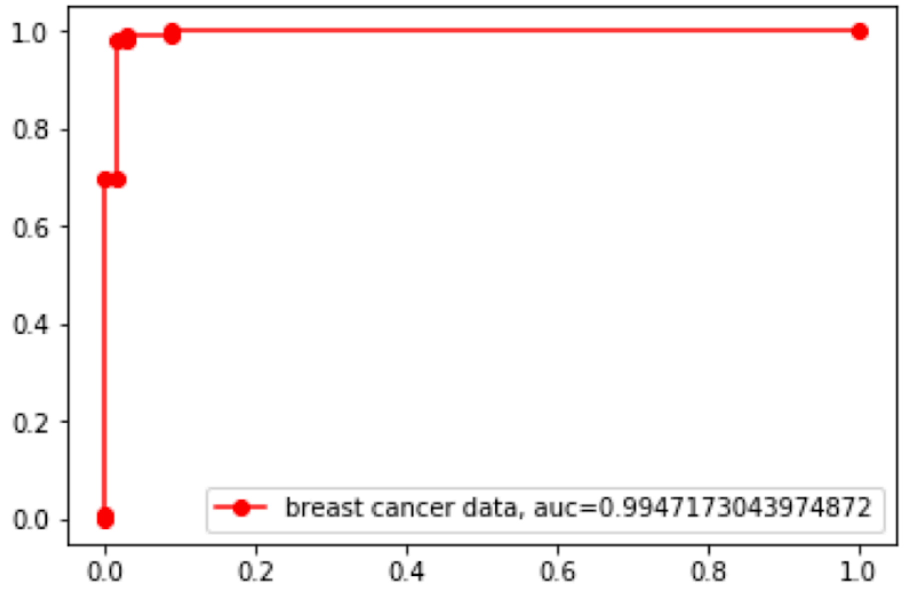
[Slide courtesy: Dr. Ke Yuan]

```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.datasets import load_breast_cancer

breast_cancer = load_breast_cancer()
X = breast_cancer.data
t = breast_cancer.target


X_train, X_test, y_train, y_test = train_test_split(X,t,test_size=0.30, random_sta
te=123)
clf1 = LogisticRegression().fit(X_train, y_train)

y_pred1 = clf1.predict(X_test)
y_pred_proba1 = clf1.predict_proba(X_test)[:,1]
fpr1, tpr1, _ = metrics.roc_curve(y_test,  y_pred_proba1)
auc1 = metrics.roc_auc_score(y_test, y_pred_proba1)
plt.plot(fpr1,tpr1,'ro-',label="breast cancer data, auc="+str(auc1))
plt.legend(loc=4)
plt.show()
```



Try it on a breast cancer dataset

Plot ROC of a Logistic Regression model

F1

- Metric combining Precision and Recall
- **F1 =** $\frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}} = \frac{2TP}{2TP + FP + FN}$
- Bounded between 0 to 1
- Higher the better

Summary

- Evaluation protocol and metrics are equally important
- Should be chosen based on data and application