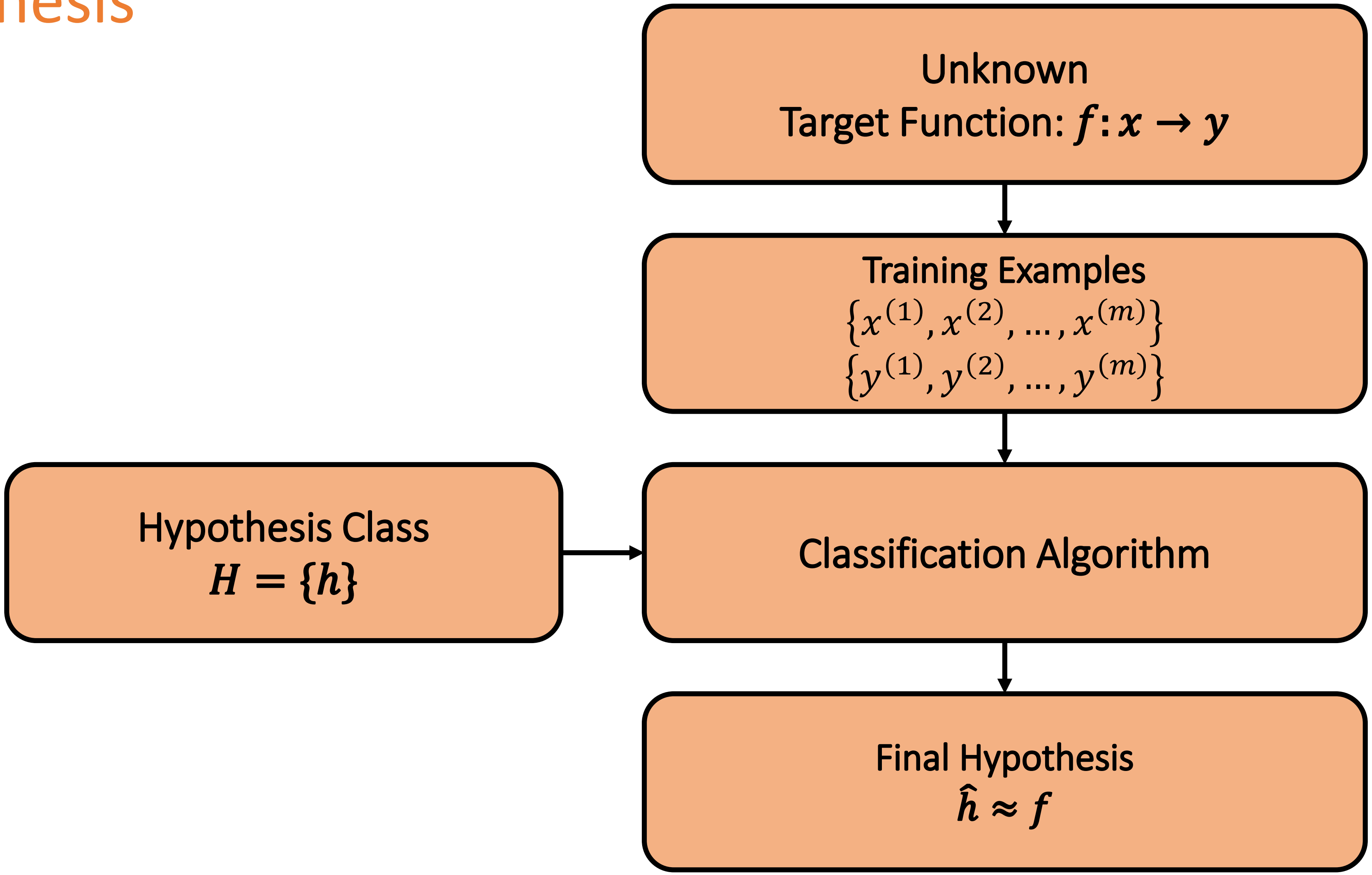
Machine Learning in Healthcare

#L08-Performance statistics

Technion-IIT, Haifa, Israel

Asst. Prof. Joachim Behar
Biomedical Engineering Faculty, Technion-IIT
Artificial intelligence in medicine laboratory (AIMLab.)
https://aim-lab.github.io/
Twitter: @lab_aim

Hypothesis


Unknown
Target Function: $f: x \to y$
Training Examples
$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
$\{y^{(1)}, y^{(2)}, \dots, y^{(m)}\}$
Hypothesis Class
$H = \{h\}$
Classification Algorithm
Final Hypothesis
$\hat{h} \approx f$

Cost function

Cross entropy cost function

- Binary LR:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)} log\left(h_w\left(x^{(i)}\right)\right) - (1 - y^{(i)})\log(1 - h_w\left(x^{(i)}\right))\right].$
- Multinomial LR:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{n_y}\left[1\{y^{(i)} = k\}\log(\frac{\exp\left(w^{(k)T}x^{(i)}\right)}{\sum_{j=1}^{K}\exp\left(w^{(k)T}x^{(i)}\right)})\right].$
- Cross-entropy takes the output probability and measures its distance from the target class. The cross-entropy loss increases as the predicted probability diverges from the true label (i.e. target class).

Cross entropy cost function

- This is what we need for our gradient descent optimization task but what about interpretability of the model performance?
- Say $J(w) = 2$ then what does it means in term of the number of patients correctly or misdiagnosed? How can we appreciate if "$J(w) = 2$" is good enough for our specific medical challenge?
- A fortiori for a multiclass classification problem.
- This is where we need **performance statistics**.

Performance statistics

Confusion matrix

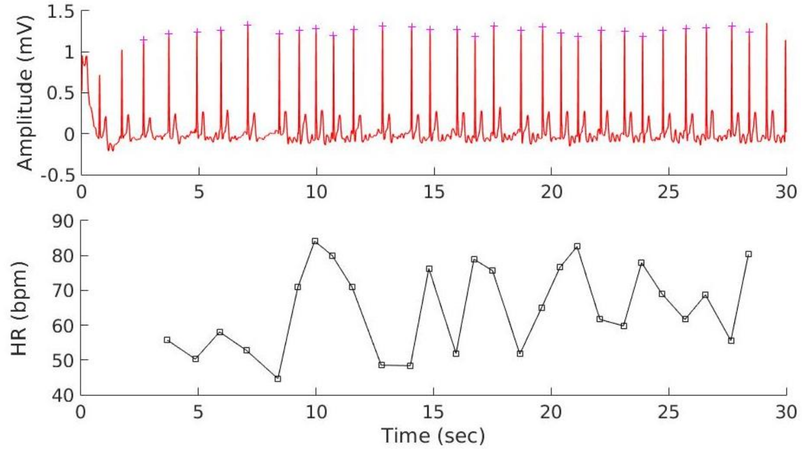
- Confusion matrix:

	Predicted No	Predicted Yes	
True label No	TN	FP	Sp = TN/(TN+FP)
True label Yes	FN	TP	Se = TP/(TP/FN)
	NPV = TN/(FN+TN)	PPV = TP/(TP+FP)	

AIMLab.

Limitation of accuracy

- Let's assume a population of 900 patients.
- 100 of them have AF and 800 are non-AF.
- This is the confusion matrix we get from our classifier.
- If we compute the accuracy:
  - $Ac = \frac{TP+TN}{TP+TN+FP+FN} = \frac{50+780}{50+780+20+50} = 0.95$
- Should we conclude our classifier is doing a great job?



	Predicted No	Predicted Yes
True label No	780 (TN)	20 (FP)
True label Yes	50 (FN)	50 (TP)

Performance statistics

My test identified 10 patients with the condition correctly. What is the proportion of patients with the condition that were correctly identified out of all with the conditions?

My test identified 20 patients without the condition correctly. What is the proportion of patients without the condition that were correctly identified out of all without the condition?

	Predicted No	Predicted Yes
True label No	TN	FP
True label Yes	FN	TP

- **Sensitivity:** proportion of people with a condition who are correctly identified by a test as indeed having that condition.
  - $Se = \frac{TP}{TP+FN}$
- **Specificity:** proportion of people without a condition who are correctly identified by a test as indeed not having the condition.
  - $Sp = \frac{TN}{TN+FP}$

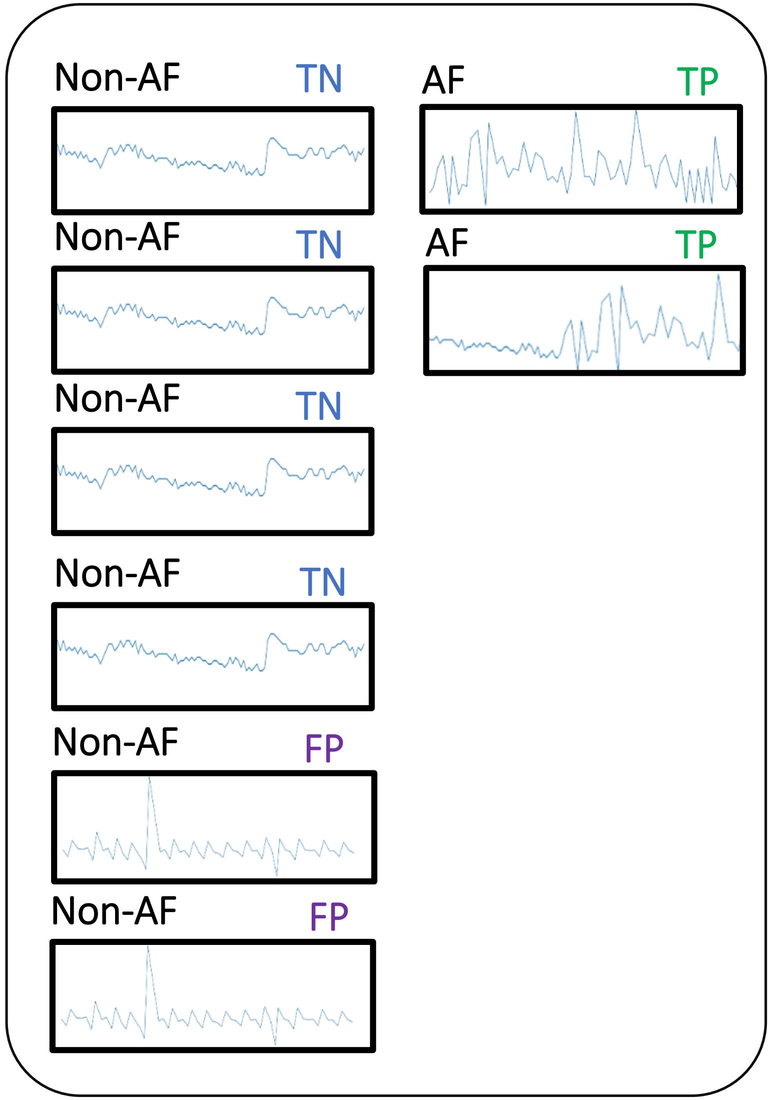
Example: focus on the positive class

Performance statistics

$$Se = \frac{2}{0 + 2} = 1$$

$$Sp = \frac{4}{4 + 2} = 0.67$$

Examples



Performance statistics

Got a patient with a positive test. What is the probability that this patient has indeed the condition?

Got a patient with a negative test. What is the probability that this patient does not indeed have the condition?

	Predicted No	Predicted Yes
True label No	TN	FP
True label Yes	FN	TP

- **Positive predictive value**: is the probability that people with a positive test result indeed do have the condition of interest.
  - $PPV = \frac{TP}{TP+FP}$
- **Negative predictive value**: probability that people with a negative test result indeed do not have the condition of interest.
  - $NPV = \frac{TN}{TN+FN}$.

AIMLab.

Example: focus on the positive class

Performance statistics

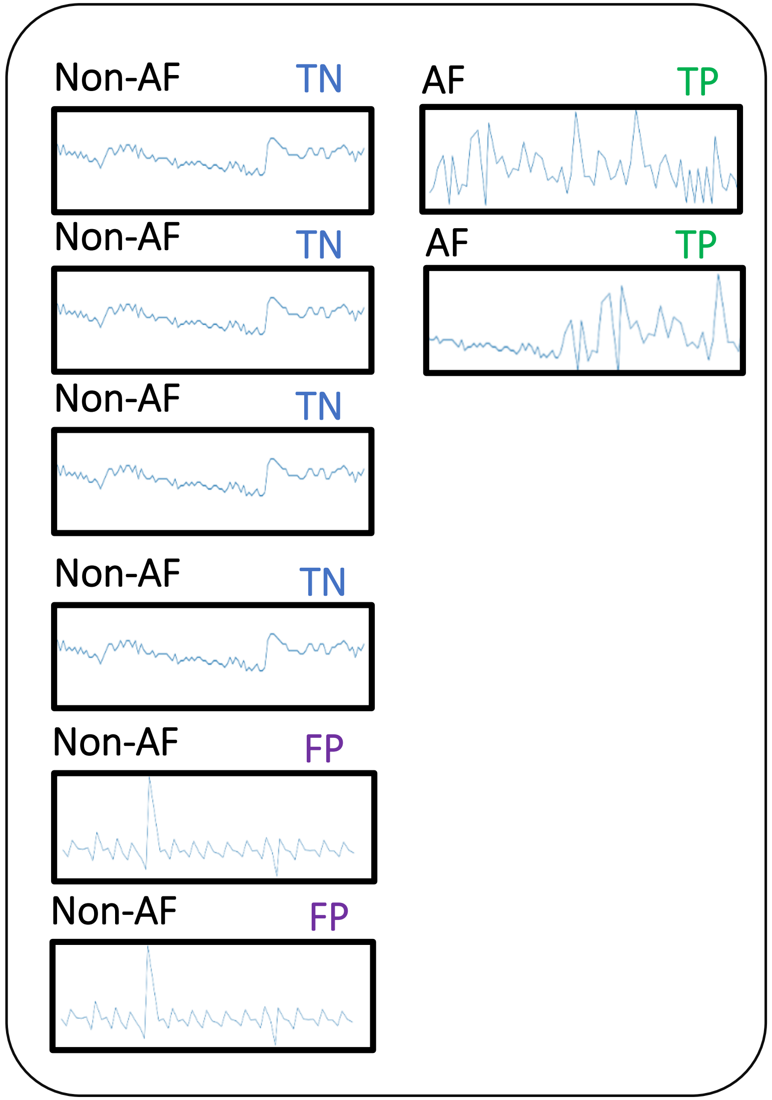
$$Se = \frac{2}{0 + 2} = 1$$

$$PPV = \frac{2}{2 + 2} = 0.5$$

$$NPV = \frac{4}{0 + 4} = 1$$

$$Sp = \frac{4}{4 + 2} = 0.67$$

Examples



Performance statistics

I want to optimize my classifier. How can I quantify its “accuracy” with one single statistical measure?

- Say TN are not what we care about (as much).
- We look for a measure that “average” Se and PPV.
- F_1: harmonic mean between Se and PPV.
- Measure of the tradeoff between Se and PPV.

	Predicted No	Predicted Yes
True label No	TN	FP
True label Yes	FN	TP

$$F_1 = 2 \cdot \frac{PPV \cdot Se}{PPV + Se} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

$$F_\beta = (1+\beta^2) \cdot \frac{PPV \cdot Se}{\beta^2 \cdot PPV + Se} = \frac{(1+\beta^2) \cdot TP}{(1+\beta^2) \cdot TP + \beta^2 \cdot FP + FN}$$

Se is β times as important as PPV.

AIMLab.

Example: focus on the positive class

Performance statistics

$$Se = \frac{2}{0 + 2} = 1$$

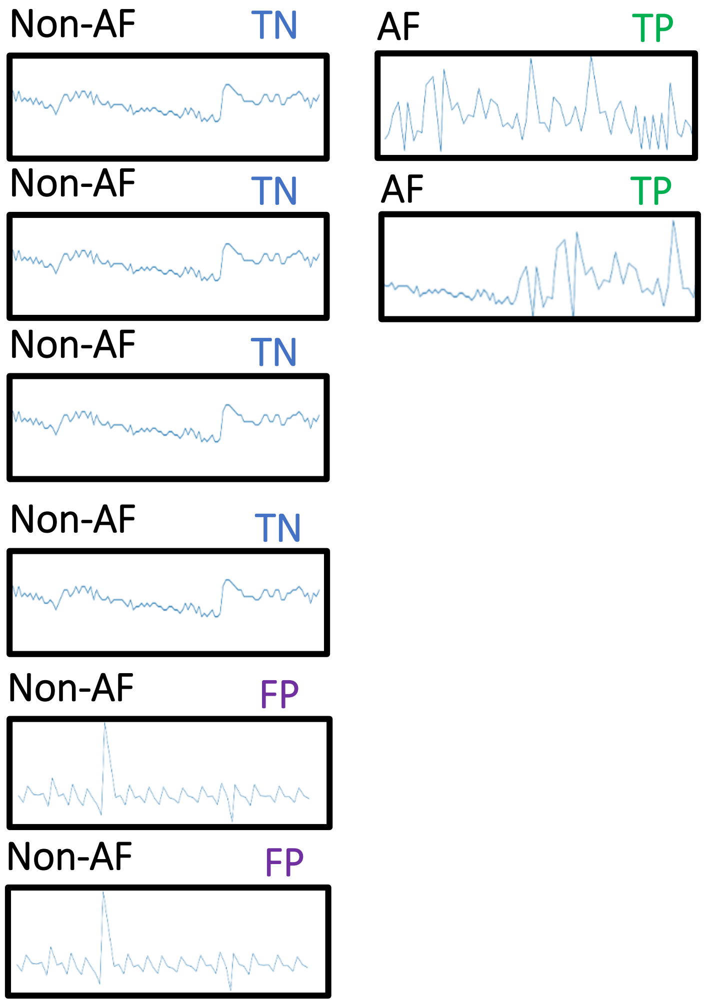
$$PPV = \frac{2}{2 + 2} = 0.5$$

$$NPV = \frac{4}{0 + 4} = 1$$

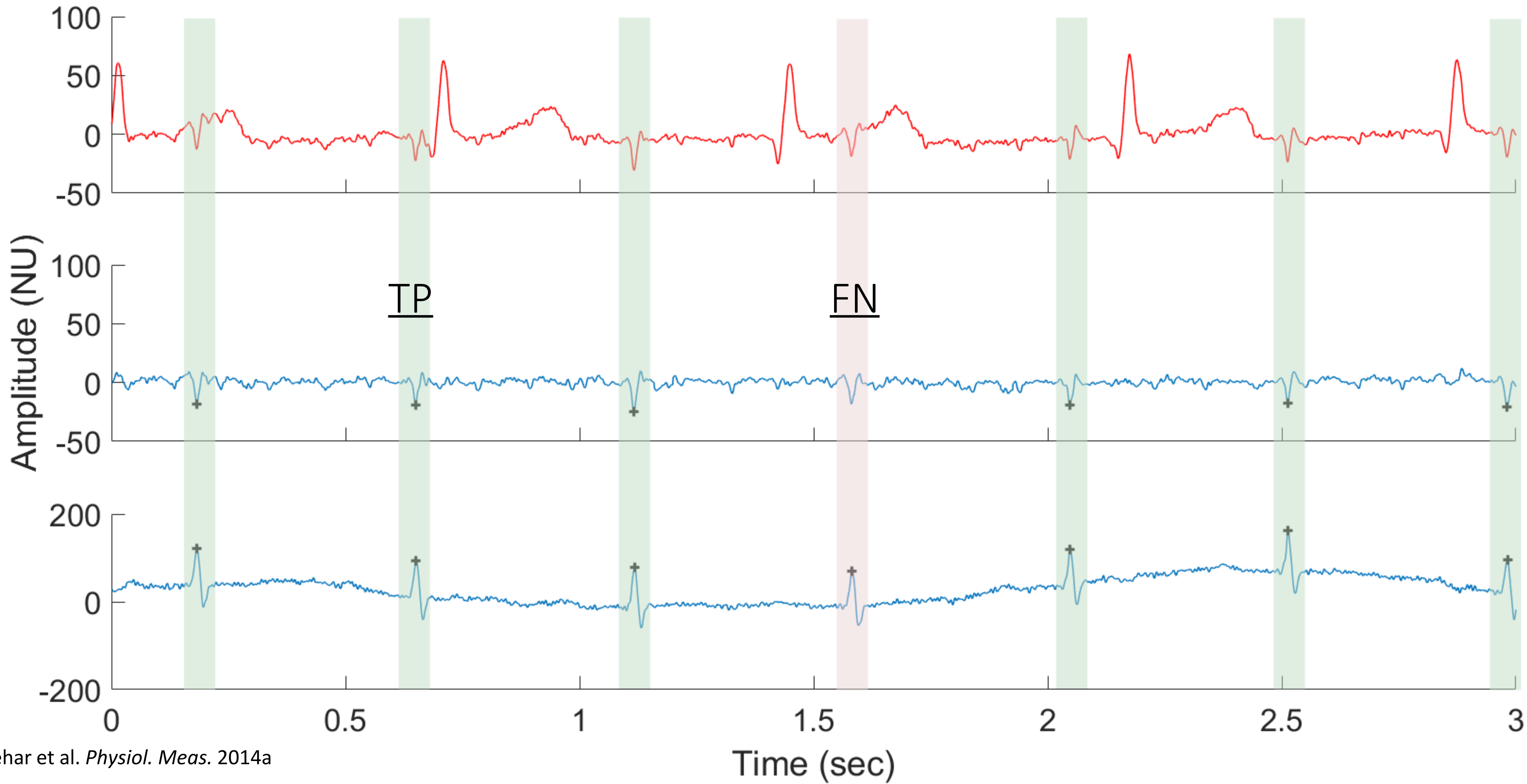
$$Sp = \frac{4}{4 + 2} = 0.67$$

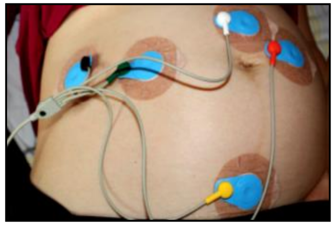
$$\boldsymbol{F1 = \frac{2 * Se * PPV}{Se + PPV} = 0.67}$$

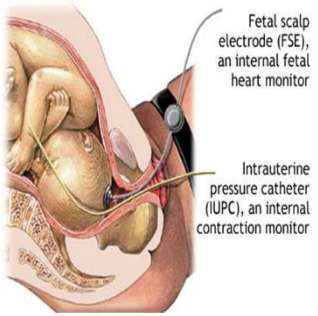
Examples



Example: focus on the positive class







Behar et al. *Physiol. Meas.* 2014a

Class imbalance and measures interpretation

- This is the confusion matrix we get from our classifier:
- If we compute the performance statics:
  - $Ac = \frac{TP+TN}{TP+TN+FP+FN} = \frac{50+780}{50+780+20+50} = 0.95$
  - $Se = \frac{TP}{TP+FN} = \frac{50}{50+50} = 0.5$
  - $Sp = \frac{TN}{TN+FP} = \frac{780}{780+20} = 0.98$
  - $PPV = \frac{TP}{TP+FP} = \frac{50}{50+20} = 0.71$
  - $NPV = \frac{TN}{FN+TN} = \frac{780}{780+50} = 0.94$
  - $F_1 = 2 \cdot \frac{PPV \cdot Se}{PPV+Se} = 0.59$
- So other stats than Ac provide better insights when classes are **skewed**.

	Predicted No	Predicted Yes
True label No	780 (TN)	20 (FP)
True label Yes	50 (FN)	50 (TP)

Performance statistics

- To summarize:
  - Se: proportion of people with a condition who are correctly identified by a test as indeed having that condition.
  - Sp: proportion of people without a condition who are correctly identified by a test as indeed not having the condition
  - PPV: is the probability that people with a positive test result indeed do have the condition of interest.
  - NPV: probability that people with a negative test result indeed do not have the condition of interest.
  - F_1: harmonic average between Se and PPV. Useful as a single measures for classifier optimization. Measure of the tradeoff between Se and PPV.

Performance statistics

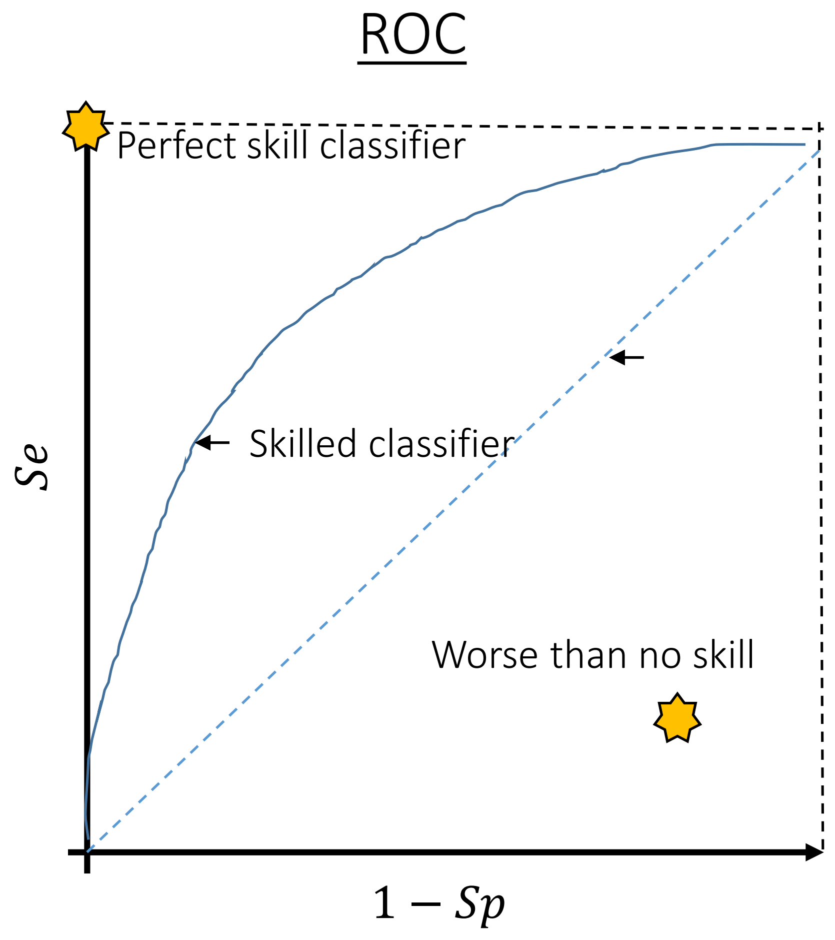
- Difference between Se, Sp and PPV, NPV:
  - Se and Sp quantify how accurate the classifier performs with respect to a reference (“ground truth”).
  - PPV and NPV encapsulate the information about the prevalence of the condition. In other words the imbalance of the classes is taken into account which might be a good thing if our population sample is characteristic of our population of interest.
  - Thus PPV and NPV are particularly appropriate when considering the performance of a medical screening test for example.
  - In practice, report Se, Sp, PPV and NPV and interpret carefully with respect to the research question.

Finding a tradeoff between Se and Sp

- We do not want to miss the AF patients, right?
- $h_w(x) \geq 0.5$ predicts AF.
- $h_w(x) \geq 0.3$ might be better as by being more lenient on the threshold we will increase our Se.
- However, this will lower our Sp.
- As we chose a different threshold we will affect the tradeoff between Se and Sp.
- How do we analyze this Se - Sp relationship and chose a tradeoff that is suitable for our particular application?

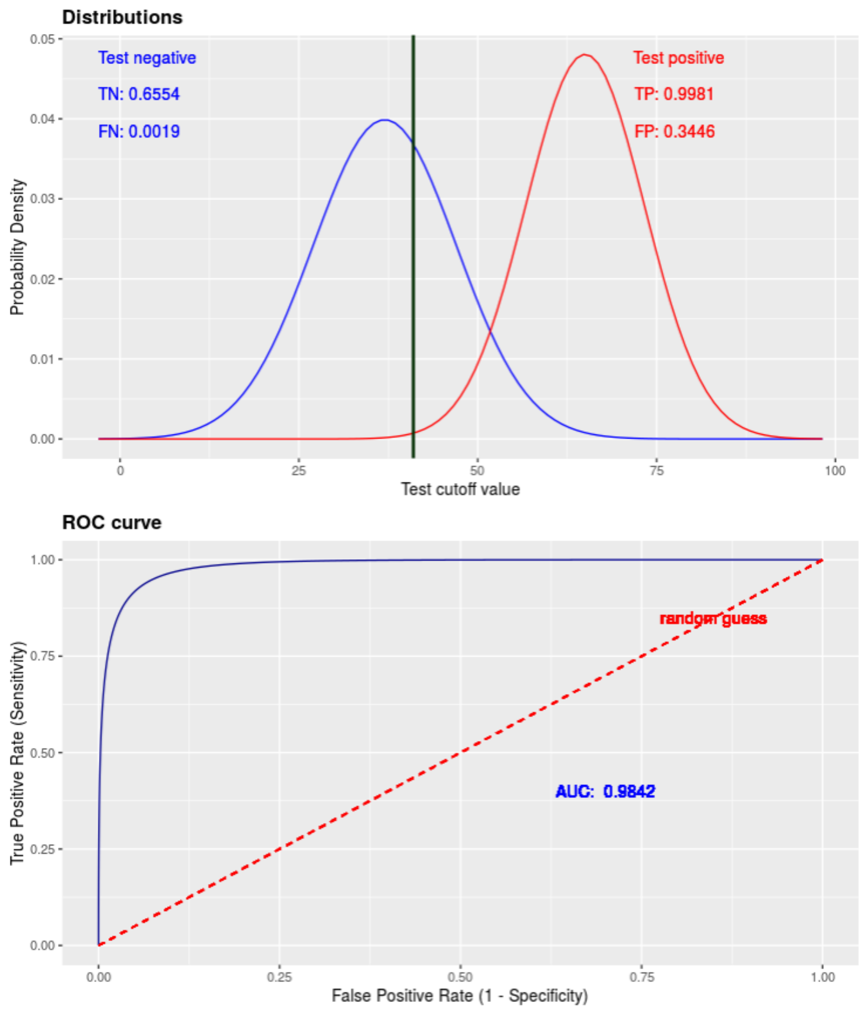
Finding a tradeoff between Se and Sp

- Receiver Operating Characteristic (ROC)
  - Diagnostic ability of a binary classifier.
  - Se is plotted against $1 - Sp$.
  - Performance measurement at different threshold values.
- The Area Under the ROC (AUROC)
  - Quantify the separability of the classes.
  - The closer from one the better.
- Extension to multiple classes:
  - Hand, David J., and Robert J. Till. "A simple generalisation of the area under the ROC curve for multiple class classification problems." Machine learning 45.2 (2001): 171-186.



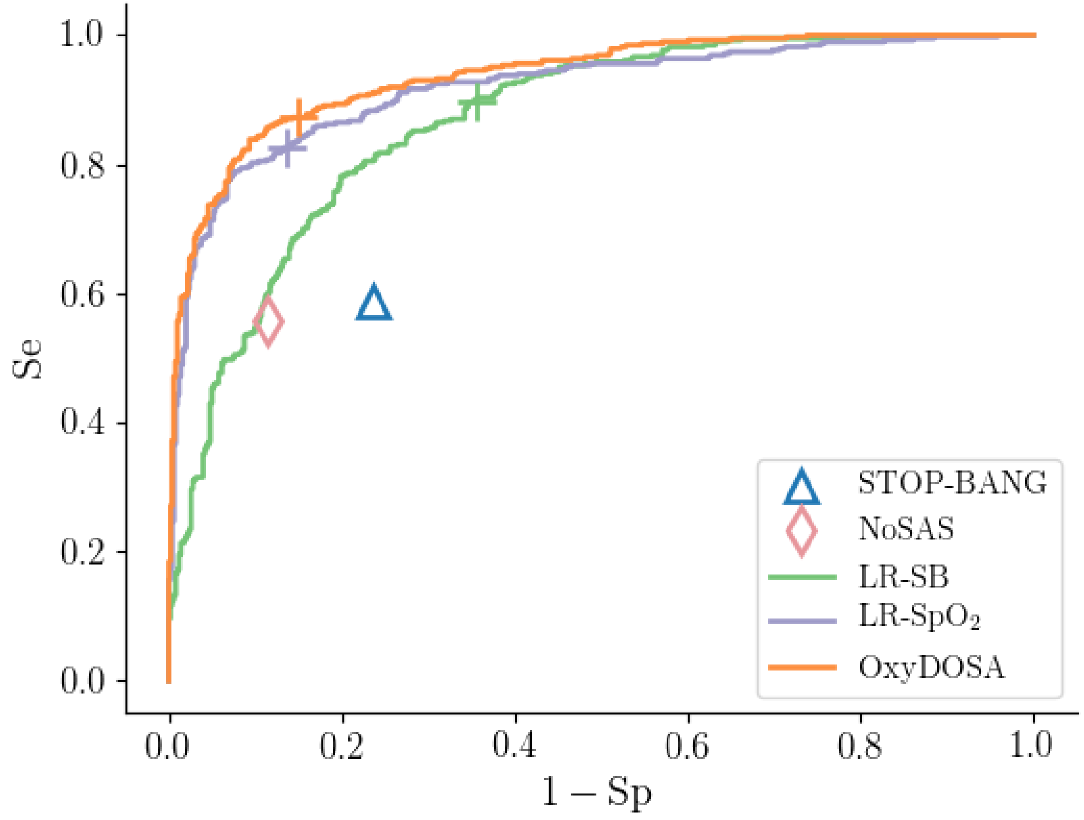
Receiver operating curve

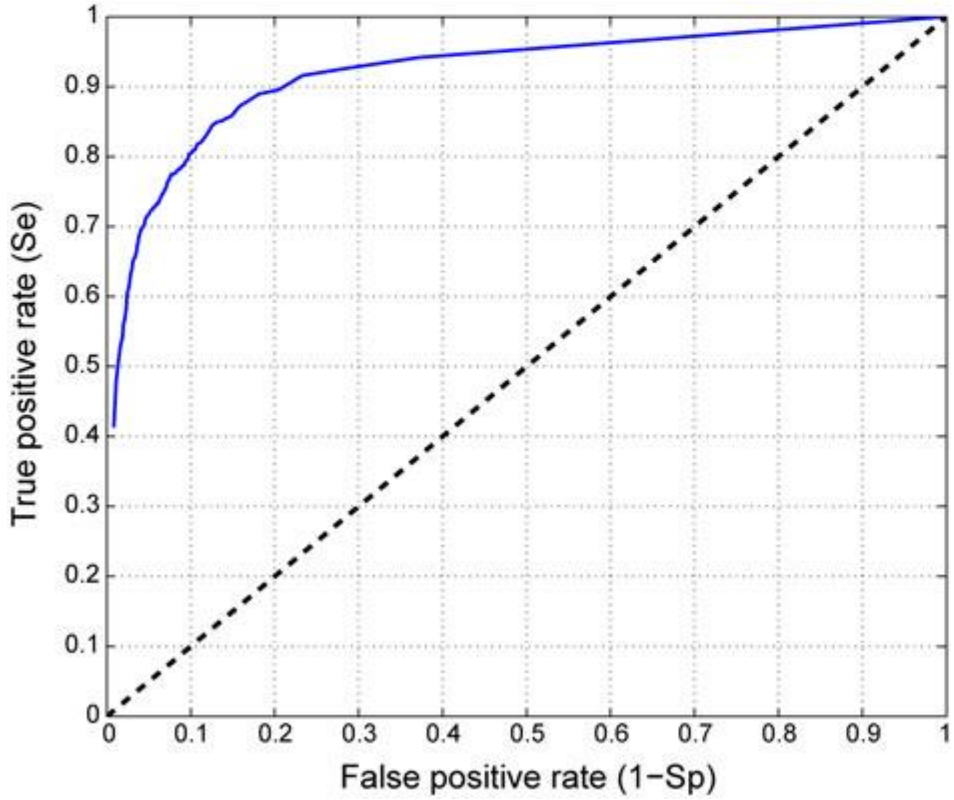
- Evaluate how well the model separates between classes for all decision threshold.
- ROC curve can help choose the threshold that balance between Se and Sp in a suitable way for your application.



https://kennis-research.shinyapps.io/ROC-Curves/

Receiver operating curve

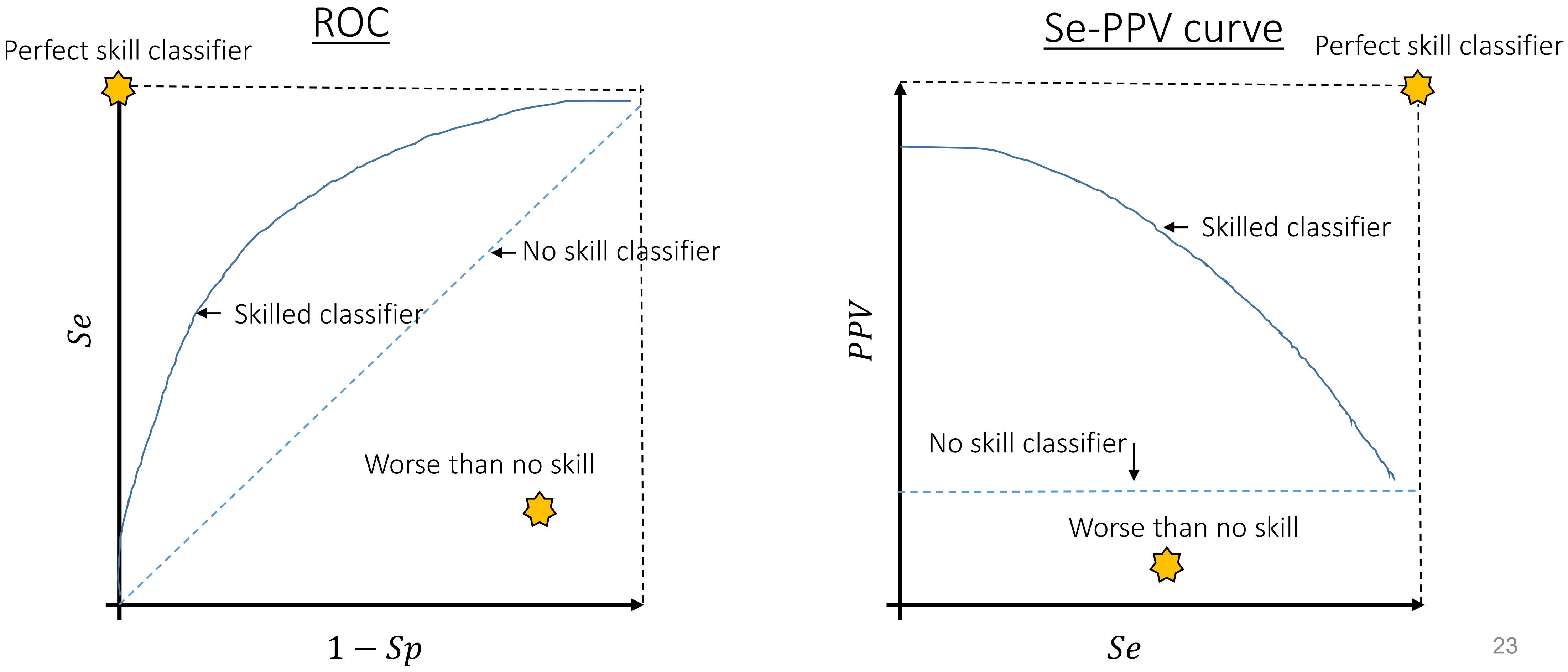




Behar, Joachim, et al. EClinicalMedicine 11 (2019).
Behar, Joachim, et al. IEEE TBME 60.6 (2013).

Finding a tradeoff between Se and PPV: focus on the positive class

- We might want to look at the Se-PPV curve.



Cohen's kappa

- The Cohen's kappa statistics, denoted κ, is a measure of agreement that adjusts the observed proportional agreement to take into account the amount of agreement which would be expected by chance. This is achieved by:

$$\kappa = \frac{p - p_e}{1 - p_e}$$

- p: the proportion of examples where there is agreement.
- p_e: the proportion of examples which are expected to be agreed on by chance.

Cohen's kappa

- $p_e = \frac{1}{m^2} \sum_{k=1}^{n_y} n_{k1} n_{k2}$
  - n_{k1}: the number of examples that rater 1 classified as belonging to k
  - n_{k2}: the number of examples that rater 1 classified as belonging to k
- $p_e = \frac{1}{m^2}(n_{11}n_{12} + n_{21}n_{22}) = \underbrace{\frac{n_{11}}{m}\frac{n_{12}}{m}} + \underbrace{\frac{n_{21}}{m}\frac{n_{22}}{m}}$

Probability that the two raters agree on class 1

Probability that the two raters agree on class 2

Cohen's kappa

- $\kappa \in \left[\frac{-p_e}{1-p_e} : 1\right]$ so the range is not necessarily -1:1 like a correlation coefficient.
- κ is specific for a given population meaning its value will depend on the class imbalance.

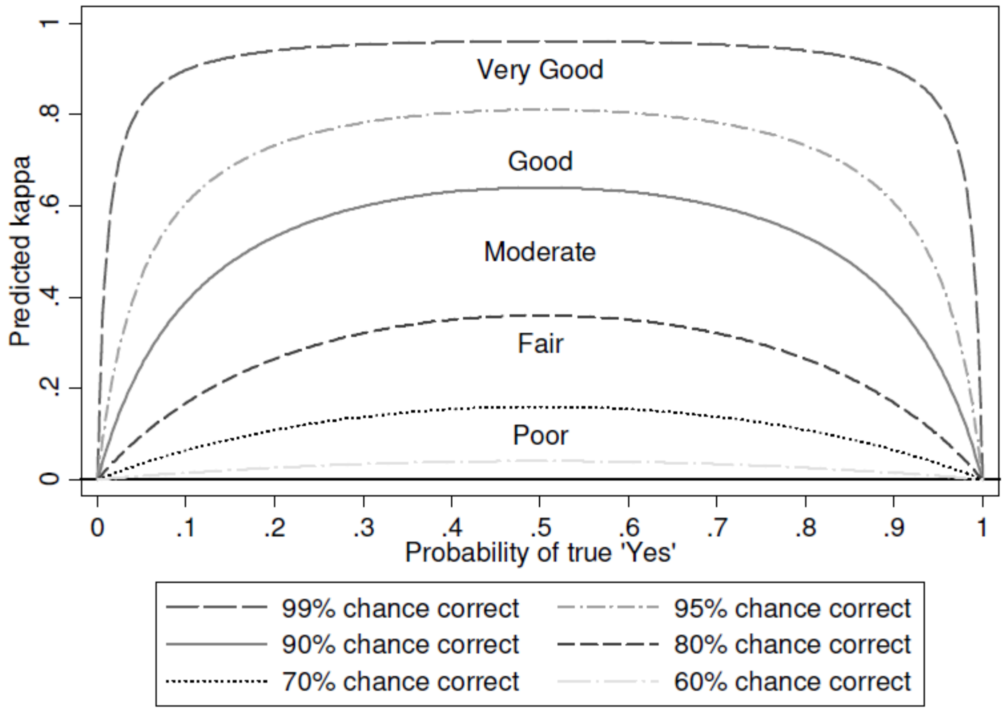
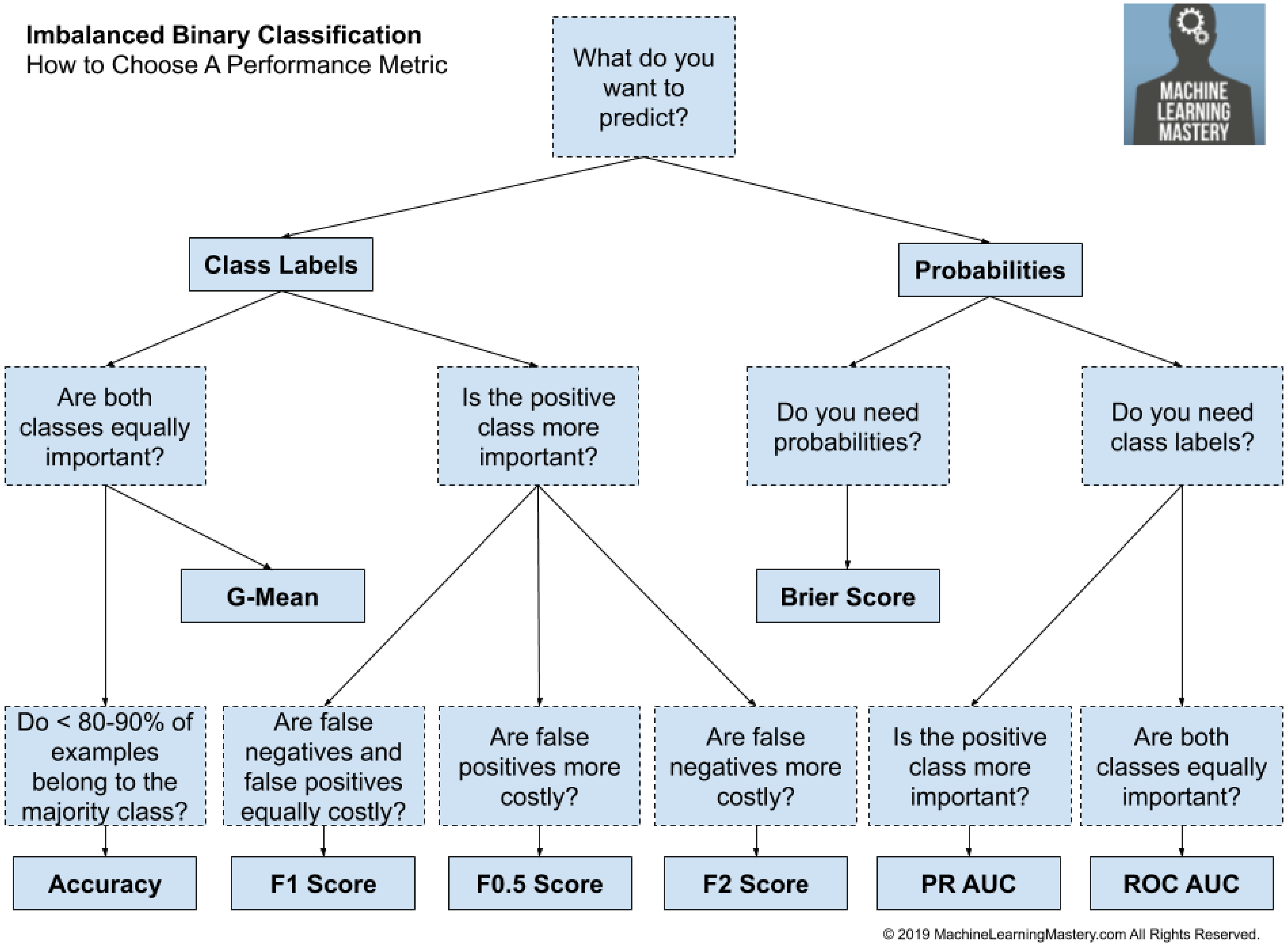


Figure 1. Predicted kappa for two categories, 'yes' and 'no', by probability of a 'yes' and probability observer will be correct. The verbal categories of Landis and Koch are shown.

Image: Bland, J. M. "Measurement in health and disease." *Cohen's Kappa. Department of Health Sciences, University of York, New YorK, UK* (2008).
URL: https://www-users.york.ac.uk/~mb55/msc/clinimet/week4/kappa_text.pdf

AIMLab.

How to choose what measure is suitable to your problem?



https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

Multiclass classification

Multiclass classification performance measures

- The measures we have seen are for binary classification.
- What if we deal with a multiclass classification problem?
- Let's consider three classes.
- We can compute the accuracy:
  - $Ac = \frac{20+50+100}{20+50+100+1+2+20+20} = 0.798$
- What about other statistics?

	Predicted C0	Predicted C1	Predicted C2
True C0	20	1	2
True C1	0	50	15
True C2	20	20	100

Multiclass classification performance measures

- We break our confusion matrix into binary ones:

	Predicted C0	Predicted C1	Predicted C2
True C0	20	1	2
True C1	0	50	15
True C2	20	20	100

	Predicted C0	Predicted Non-C0
True C0	20	3
True Non-C0	20	185

	Predicted C1	Predicted Non-C1
True C1	50	15
True Non-C1	21	142

	Predicted C2	Predicted Non-C2
True C2	100	40
True Non-C2	17	71

Multiclass classification performance meas

- We can compute the TP_k, TN_k, FP_k and FN_k for $k \in [\![1,2,3]\!]$.
- Average accuracy:

$$Ac = \frac{1}{n_y}\sum_{k=1}^{n_y}\frac{TP_k + TN_k}{TP_k + TN_k + TF_k + FN_k}$$

- Will give equal contribution to each of the three classes independent of their number of examples.

	Predicted C0	Predicted Non-C0
True C0	20	3
True Non-C0	20	185

	Predicted C1	Predicted Non-C1
True C1	50	15
True Non-C1	21	142

	Predicted C2	Predicted Non-C2
True C2	100	40
True Non-C2	17	71

Multiclass classification performance meas

- We can define **micro averaged** performance statistics:
  - $Se_\mu = \frac{\sum_{k=1}^{n_y} TP_k}{\sum_{k=1}^{n_y}(TP_k+FN_k)}$
  - $PPV_\mu = \frac{\sum_{k=1}^{n_y} TP_k}{\sum_{k=1}^{n_y}(TP_k+FP_k)}$
  - $F_{1,\mu} = 2 \cdot \frac{PPV_\mu \cdot Se_\mu}{PPV_\mu + Se_\mu}$

	Predicted C0	Predicted Non-C0
True C0	20	3
True Non-C0	20	185

	Predicted C1	Predicted Non-C1
True C1	50	15
True Non-C1	21	142

	Predicted C2	Predicted Non-C2
True C2	100	40
True Non-C2	17	71

Multiclass classification performance meas

- We can define **macro averaged** performance statistics:
  - $Se_M = \frac{1}{n_y} \sum_{k=1}^{n_y} \frac{TP_k}{TP_k + FN_k}$
  - $PPV_M = \frac{1}{n_y} \sum_{k=1}^{n_y} \frac{TP_k}{TP_k + FP_k}$
  - $F_{1,M} = 2 \cdot \frac{PPV_M \cdot Se_M}{PPV_M + Se_M}$

	Predicted C0	Predicted Non-C0
True C0	20	3
True Non-C0	20	185

	Predicted C1	Predicted Non-C1
True C1	50	15
True Non-C1	21	142

	Predicted C2	Predicted Non-C2
True C2	100	40
True Non-C2	17	71

Training the final ML model

Training the final ML model

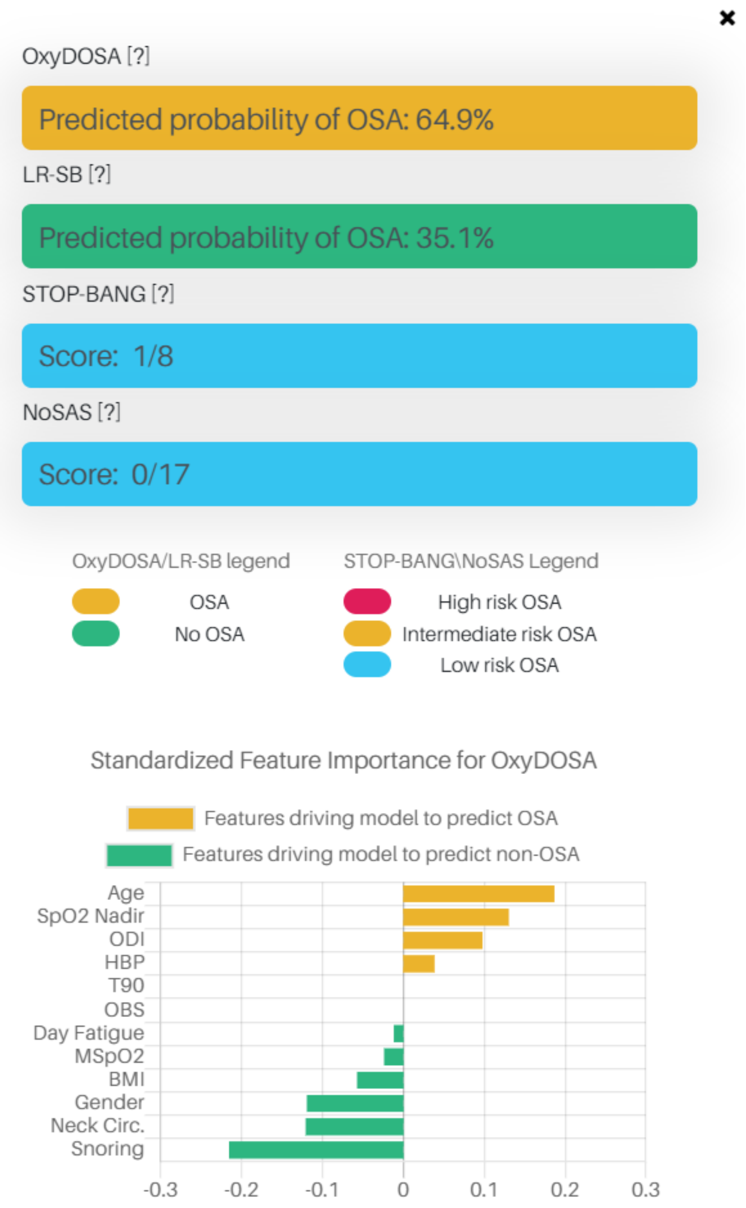
- You have performed cross-validation and you are satisfied with the performance results you got on the test set using one or a set of the performance statistics. Great!
- Now you need to generate the "**final model**" that is the model you will deploy in the real world and use to make prediction on new examples. How do you do that?
- At this point we have figured out:
  - How to prepare our data (e.g. what scaling approach to use),
  - What algorithm to use and with what complexity (e.g. regression with power d),
  - Found suitable hyperparameters (e.g. regularization parameter λ).

Training the final ML model

- The performance of our model on the test set represents how our algorithm will perform on unseen examples. Thus at this point we have designed well our procedure and found a suitable model.
- The train-validation-test set split/cross validation procedure has served its purpose and we do not need them further.
- We could just take the model trained with the optimized hyperparameters on the training set. However, we would not take in the test set data.
- You can generate the final model by applying the selected ML model (preprocessing, algorithm type, algorithm hyperparameters) on the whole dataset.

Training the final ML model

- Example of such implementation:
  - https://aim-lab.github.io/oxydosa.html



Take Home

- Cost function versus performance statistics.
- **Confusion matrix**. **Performance statistics** and their meaning.
  - Sensitivity Se, specificity Sp, positive predictive value PPV, negative predictive value NPV, F_1measure and $AUROC$.
  - For multiclass classification: micro and macro average statistics.
- Depending on the question you ask you will use one set of statistics or another.
- A practical tip:
  - Start with a simple algorithm, obtain results on cross-validation. Then plot the learning curves and get an idea of where you can improve.
  - Avoid “premature optimization” and let evidence guide your design.

Take Home

- When you are done with the model evaluation and are satisfied with its performance (according to some representative performance statistics that are tailored to your problem) then you can generate the **final model** by training the model you have identified (preprocessing procedure, algorithm, hyperparameters) on the whole dataset.

References

[1] Coursera, Andrew Ng. Advice for applying machine learning.

[2] Machine Learning Basic Concepts. CDT Lectures Notes 2013. Alistair Johnson.

[3] How to Train a Final Machine Learning Model. Jason Brownlee

https://machinelearningmastery.com/train-final-machine-learning-model/

[4] Trevethan, R. (2017). Sensitivity, specificity, and predictive values: foundations, pliabilities, and pitfalls in research and practice. Frontiers in public health, 5, 307.

[5] Sokolova, Marina, and Guy Lapalme. "A systematic analysis of performance measures for classification tasks." Information processing & management 45.4 (2009): 427-437.

[6] Bland, J. M. "Measurement in health and disease." Cohen's Kappa. Department of Health Sciences, University of York, New YorK, UK (2008).
URL: https://www-users.york.ac.uk/~mb55/msc/clinimet/week4/kappa_text.pdf