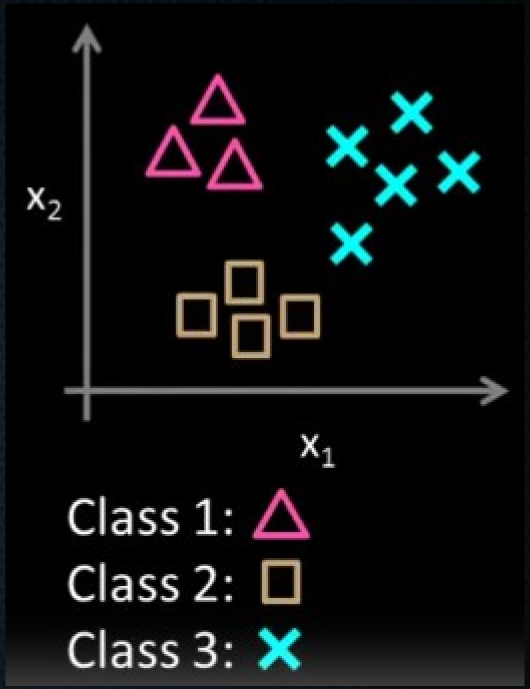
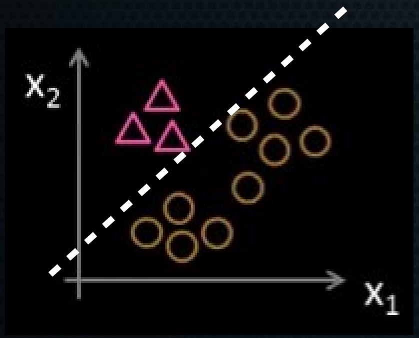
What if we have more than one class?

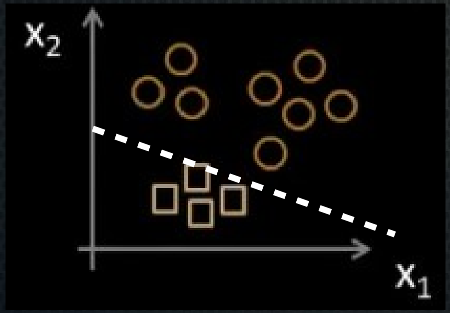
- What do you think we could do?

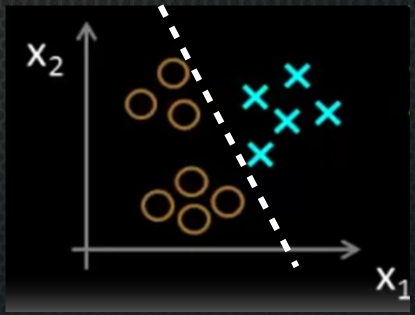


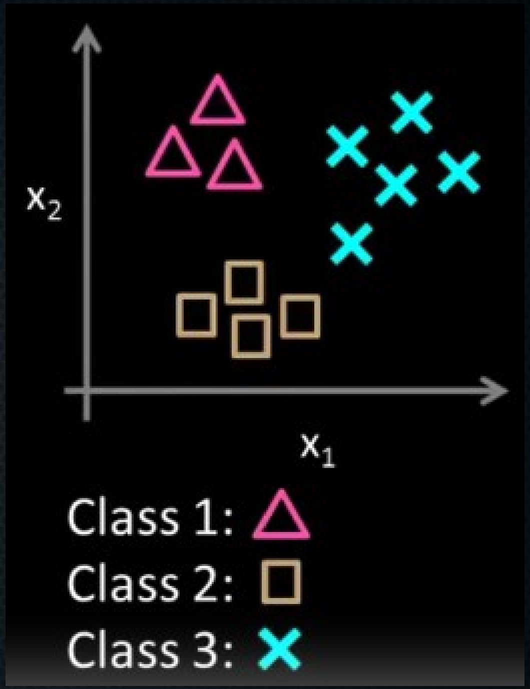
What if we have more than one class?

- What do you think we could do?
- Train a separate binary classifier for each instance:



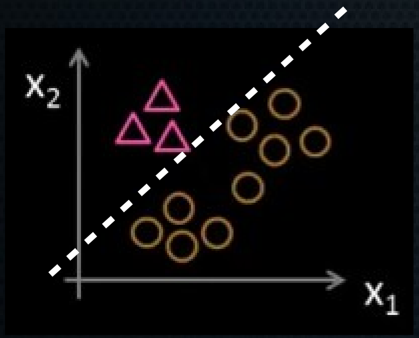


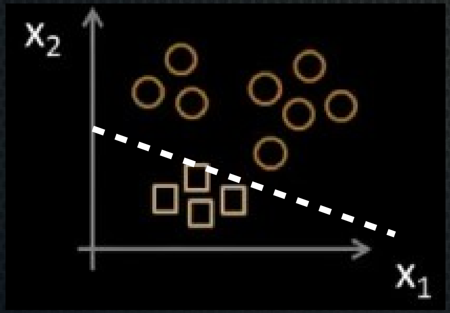


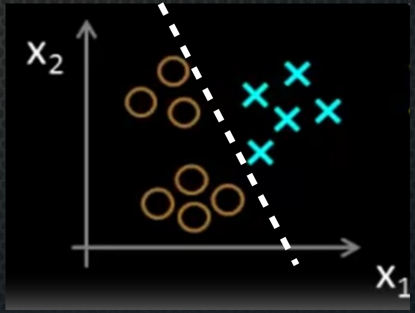


What if we have more than one class?

- What do you think we could do?
- Train a separate binary classifier for each instance:



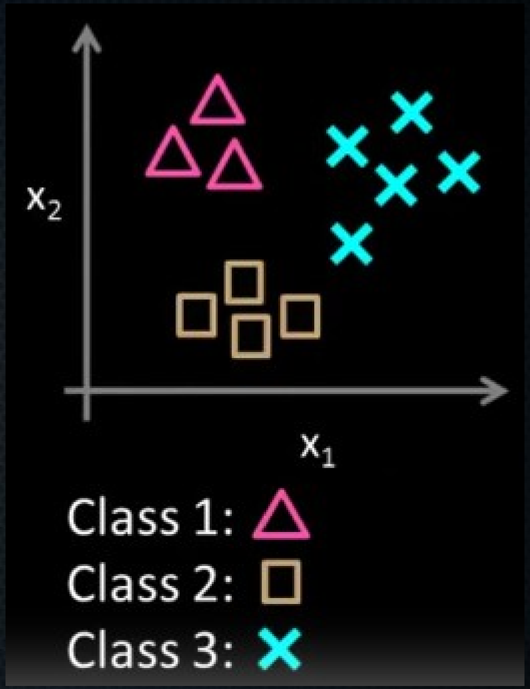




$h_\theta^{(1)}(x)$ $h_\theta^{(2)}(x)$ $h_\theta^{(3)}(x)$

$P(y=i|x;\theta)$

For *i* classes, train *i* binary classifiers to predict that the point is class *i* given the data



How well are we doing?

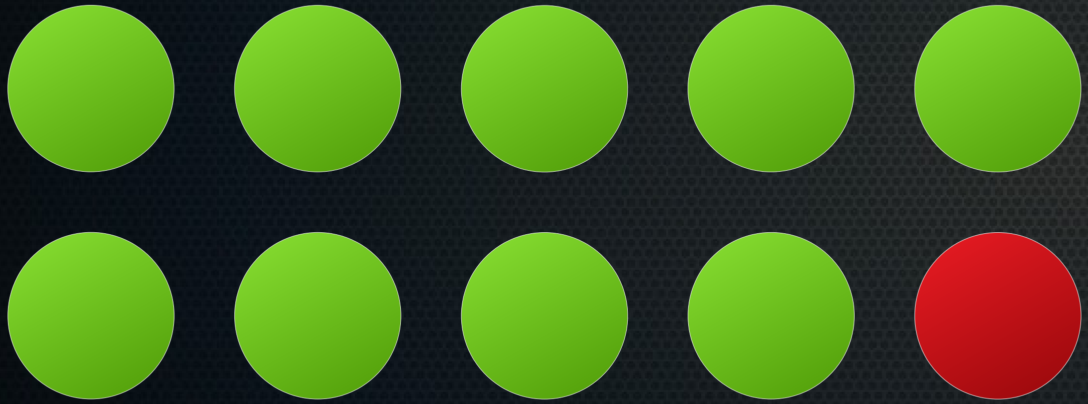
- We can‘t use an R-squared here. So what can we use?
- Idea: accuracy.

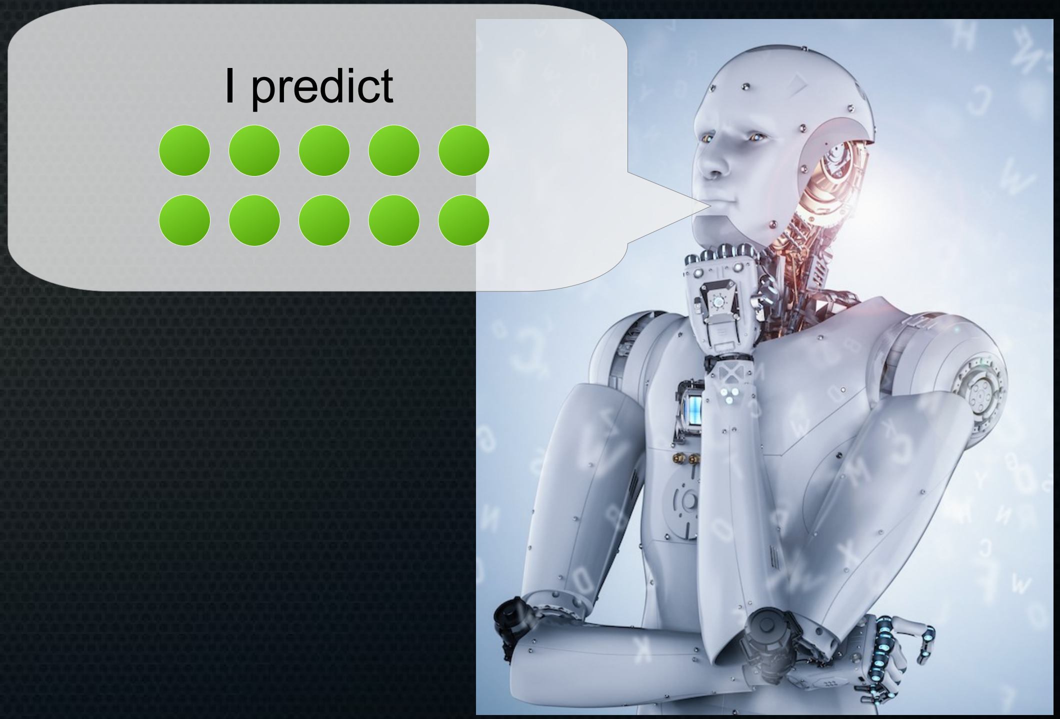
How well are we doing?

- We can‘t use an R-squared here. So what can we use?
- Idea: accuracy.

How well are we doing?

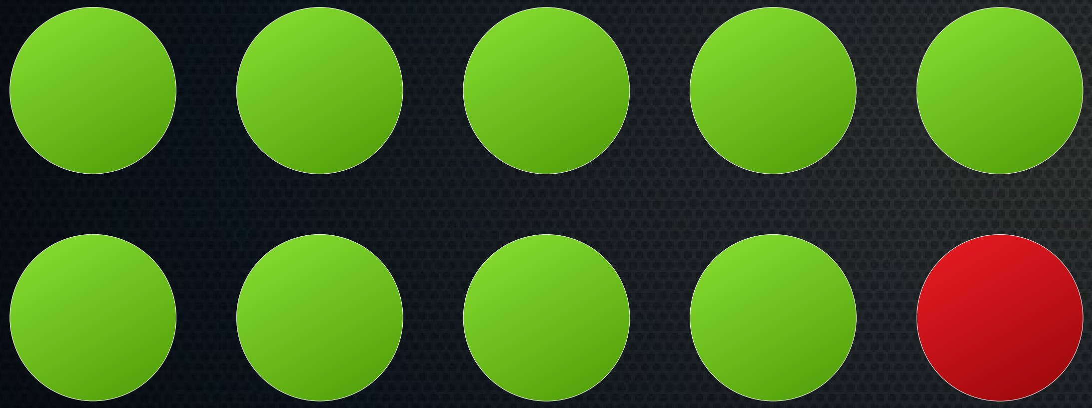
- We can‘t use an R-squared here. So what can we use?
- Idea: accuracy.



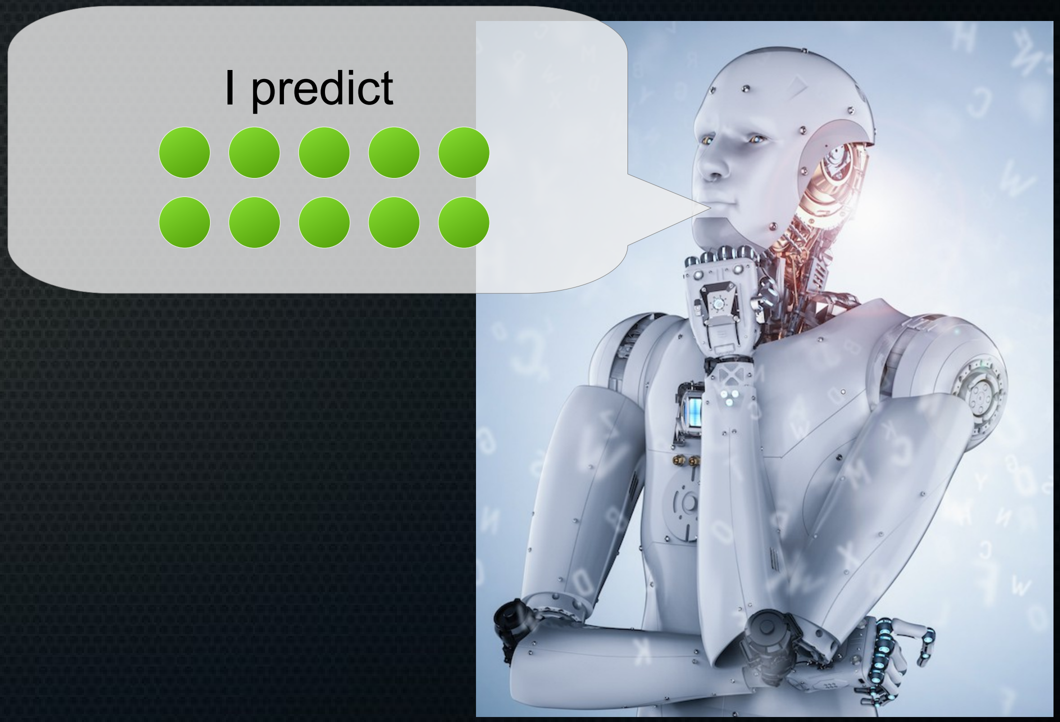


How well are we doing?

- We can‘t use an R-squared here. So what can we use?
- Idea: accuracy.

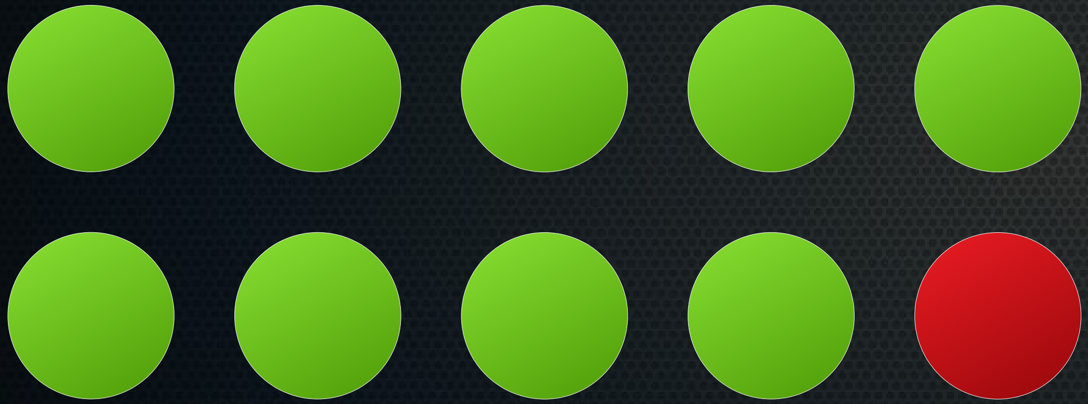


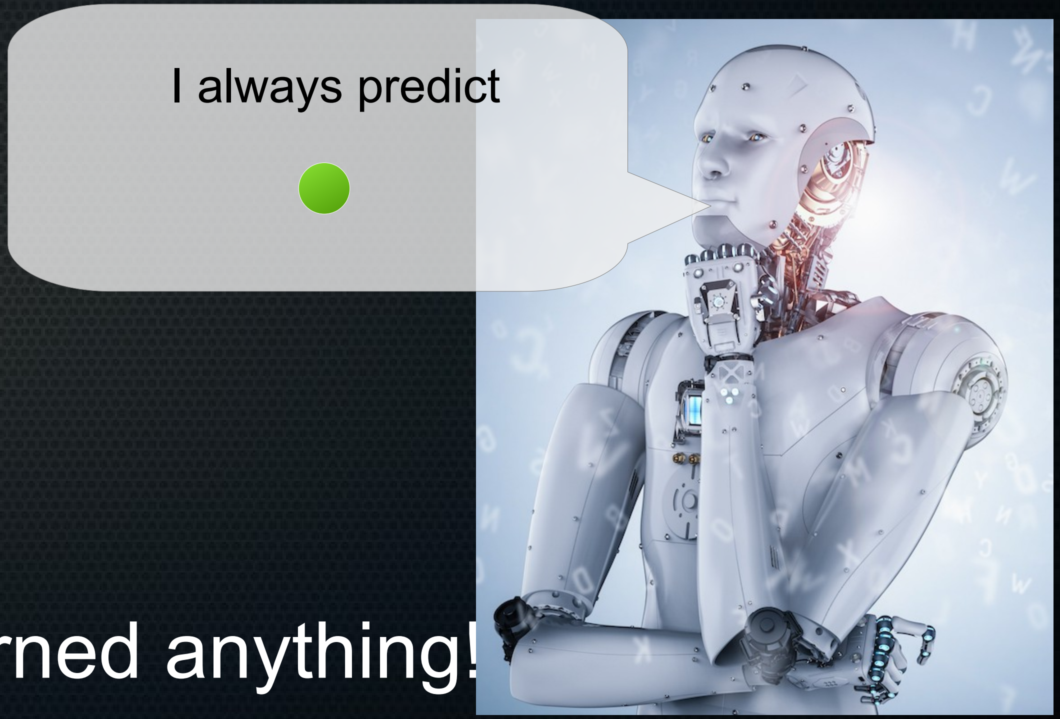
- Accuracy = 90% (9/10 correct).
  → seems pretty good!



How well are we doing?

- We can't use an R-squared here. So what can we use?
- Idea: accuracy.





- Accuracy = 90% (9/10 correct).
  → Lucky break! Classifier hasn't learned anything!

Need something else to measure performance

- Have 4 types of predictions:

		Reality	
		Cancer	Not Cancer
Model Prediction	Cancer	True Positive	False Positive
	Not Cancer	False Negative	True Negative

Need something else to measure performance

- Have 4 types of predictions:
- We want to know both how many true positives we pick out from the test data (sensitivity) and how many true negatives we correctly classify as negative (specificity).

		Reality	
		Cancer	Not Cancer
Model Prediction	Cancer	True Positive	False Positive
	Not Cancer	False Negative	True Negative

Need something else to measure performance

- Have 4 types of predictions:
- We want to know both how many true positives we pick out from the test data (sensitivity) and how many true negatives we correctly classify as negative (specificity).

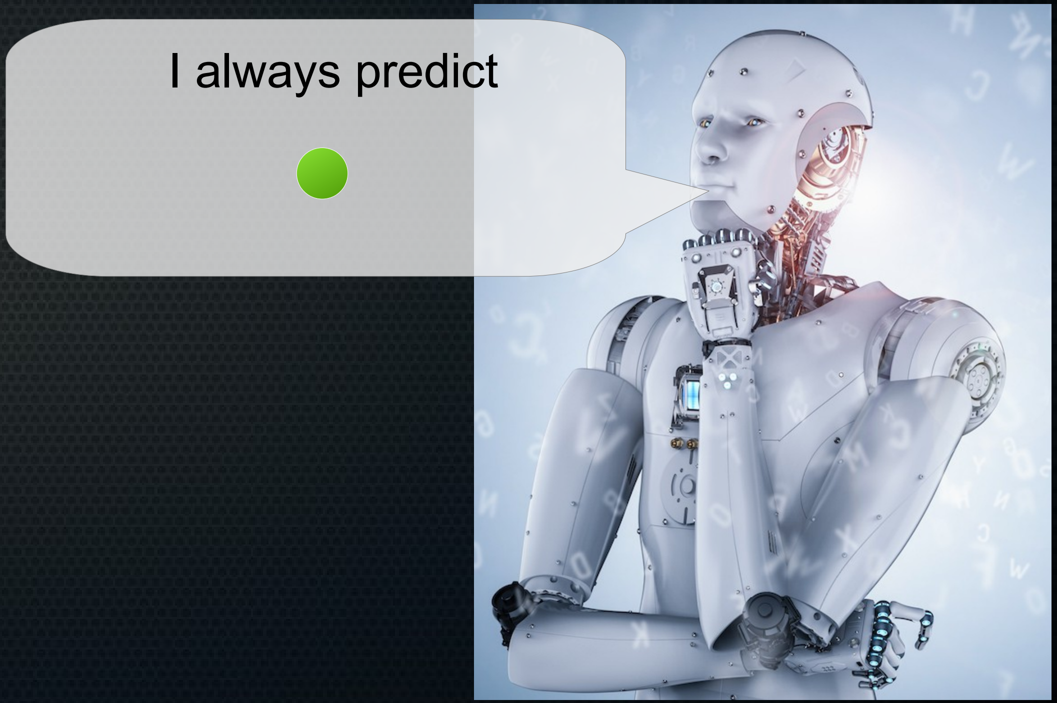
		Reality	
		Cancer	Not Cancer
Model Prediction	Cancer	True Positive	False Positive
	Not Cancer	False Negative	True Negative

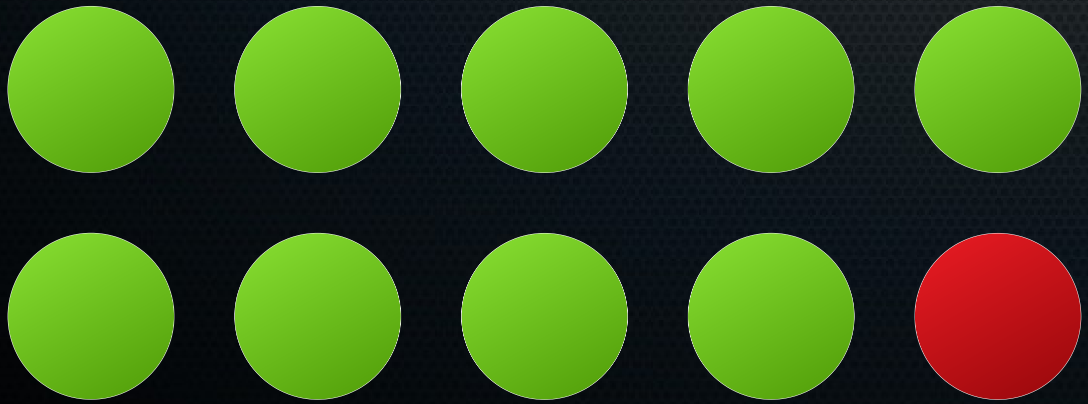
Sensitivity (true positive rate) $\dfrac{True\ Positives}{True\ Positives + False\ Negatives}$ → What proportion of positives in the data do we correctly predict?

Specificity (true negative rate) $\dfrac{True\ Negatives}{True\ Negatives + False\ Positives}$ → What proportion of negatives in the data do we correctly predict?

How well are we doing?

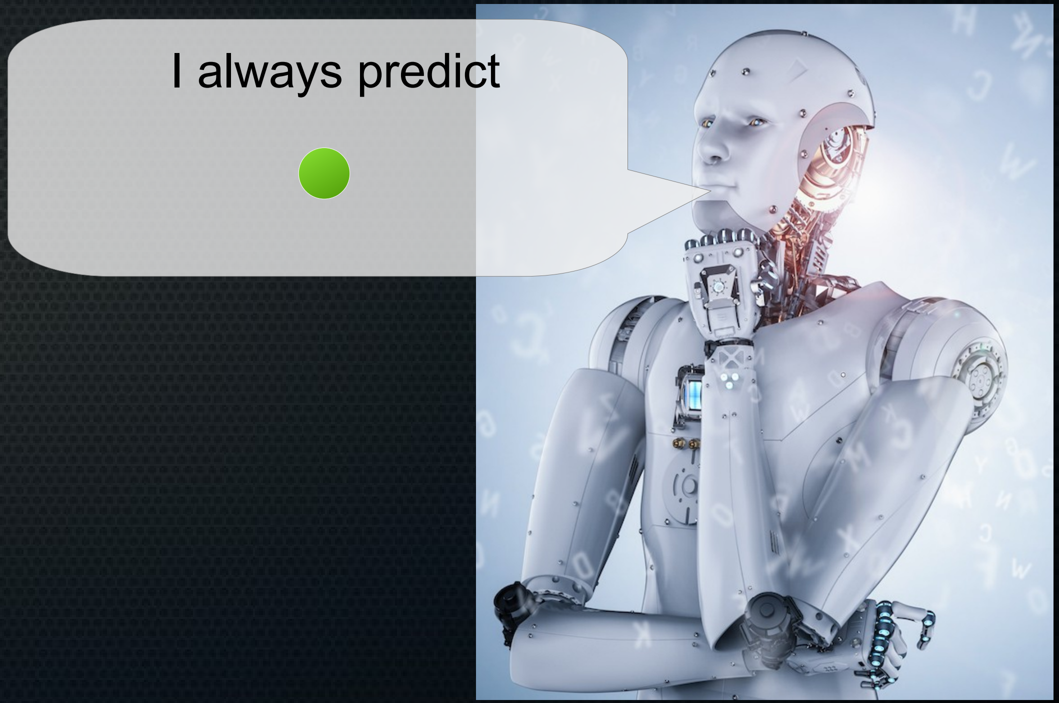
- Accuracy = 90% (9/10 correct).
  → Lucky break! Classifier hasn‘t learned anything!
- Sensitivity = 100%
- Specificity = 0%

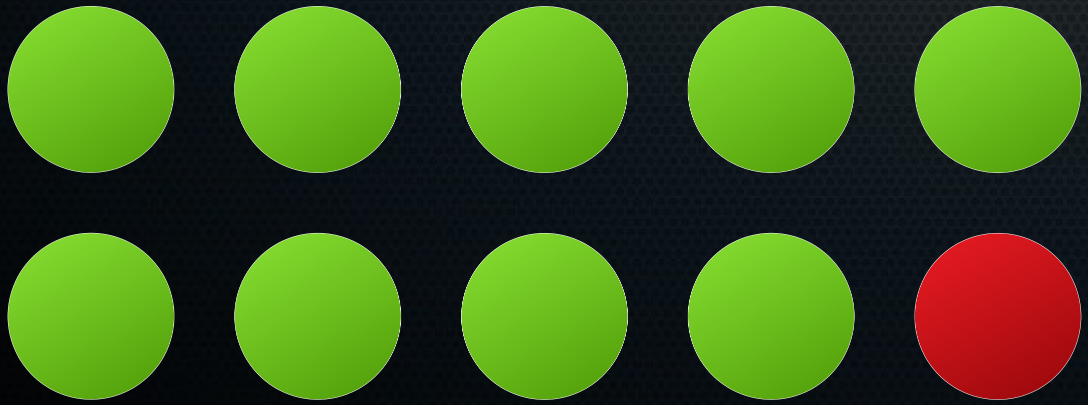




How well are we doing?

- Accuracy = 90% (9/10 correct).
  → Lucky break! Classifier hasn‘t learned anything!
- Sensitivity = 100% } Found all positives
- Specificity = 0% } By having 0 discerning ability





ROC curve

- What is the best balance between sensitivity and specificity?
  - Depends on your application:

Sensitivity (true positive rate) $\frac{True\ Positives}{True\ Positives + False\ Negatives}$

Specificity (true negative rate) $\frac{True\ Negatives}{True\ Negatives + False\ Positives}$

Patient data

No chemo

Chemotherapy

What do you care about most in each case?

Patient data

No follow-up screening

Follow-up diabetes

ROC curve

- What is the best balance between sensitivity and specificity?
  - Depends on your application:

Sensitivity (true positive rate) $\frac{True\ Positives}{True\ Positives + False\ Negatives}$

Specificity (true negative rate) $\frac{True\ Negatives}{True\ Negatives + False\ Positives}$

Patient data

Machine Learning Model

No chemo | Chemotherapy

Don‘t want to give devastating chemo treatments unnecessarily: care most about specificity!

Patient data

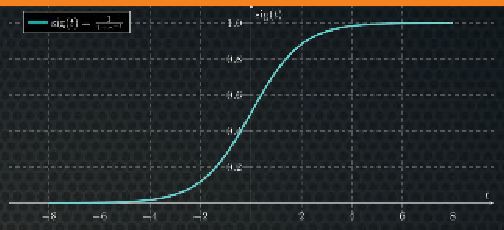
Machine Learning Model

No follow-up screening | Follow-up diabetes

Don‘t want to miss early signs diabetes if follow-up tests will confirm or deny: care most about sensitivity!

ROC curve

- How do you implement a focus on specificity or sensitivity?
  - How do we work with this? $h_\theta(x) = \frac{1}{1+e^{(\theta^T \cdot x)}}$



    - Interpret outcome of $h_\theta(x)$ as probability that class = 1 given the features. Example:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{Tumor size} \\ \text{Neovascularisation level} \end{bmatrix}$$

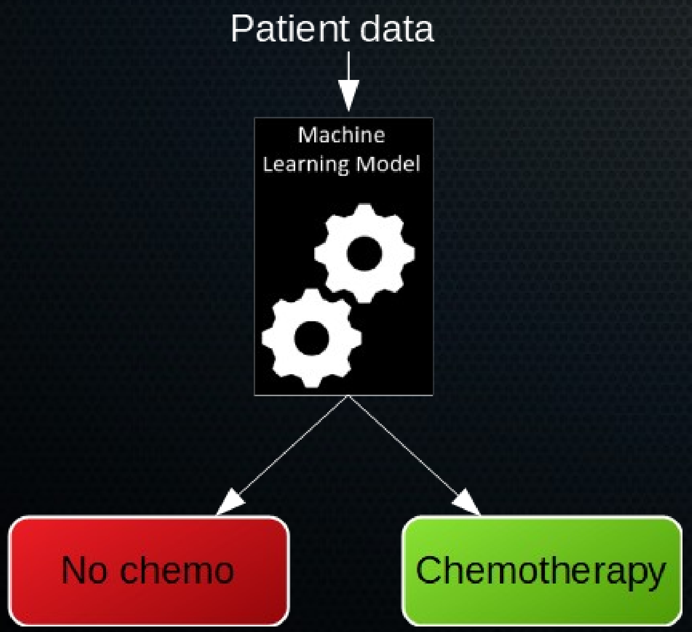
$h_\theta(x) = 0.8 \longrightarrow$ 80% chance of tumor being malignant (class 1)

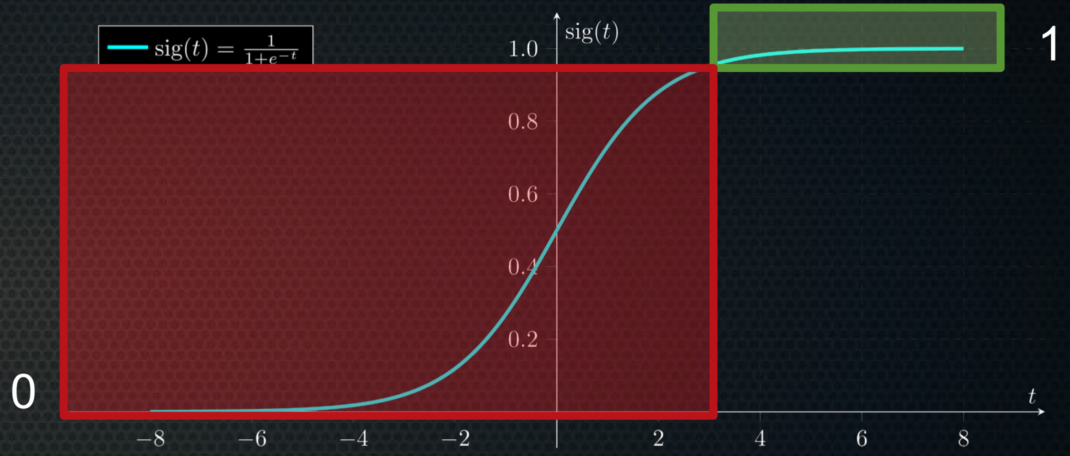
100% - 80% → 20 % chance of being benign (class 0)

ROC curve

- How do you implement a focus on specificity or sensitivity?
- We could say:

$$\text{if } h_\theta(x) \geq 0.95 \text{ classify as positive}$$
$$\text{else classify as negative}$$

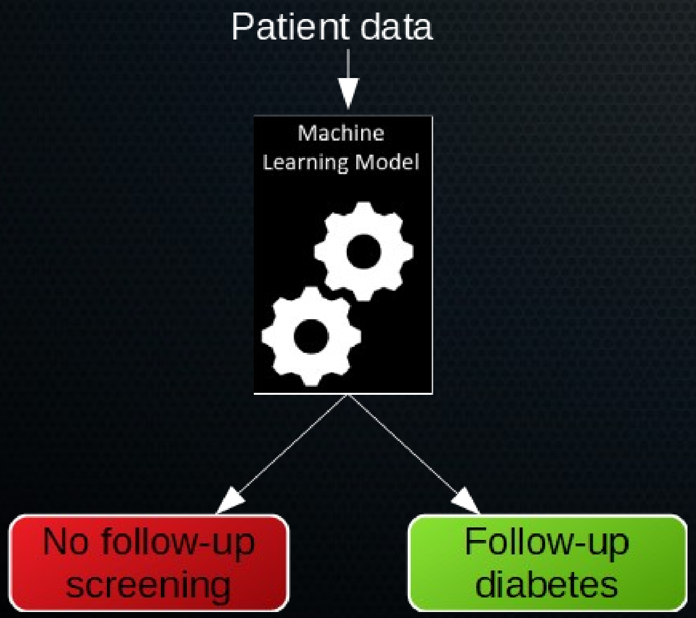


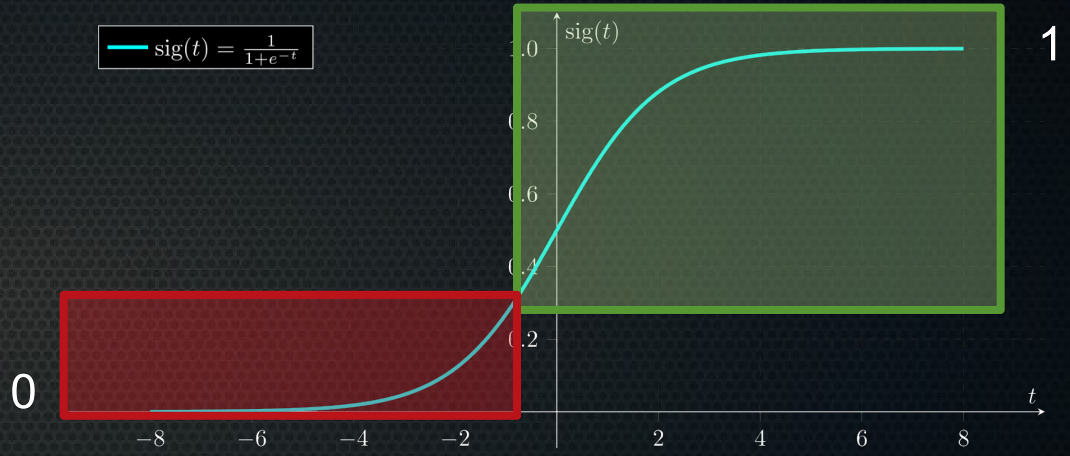


ROC curve

- How do you implement a focus on specificity or sensitivity?
- We could say:

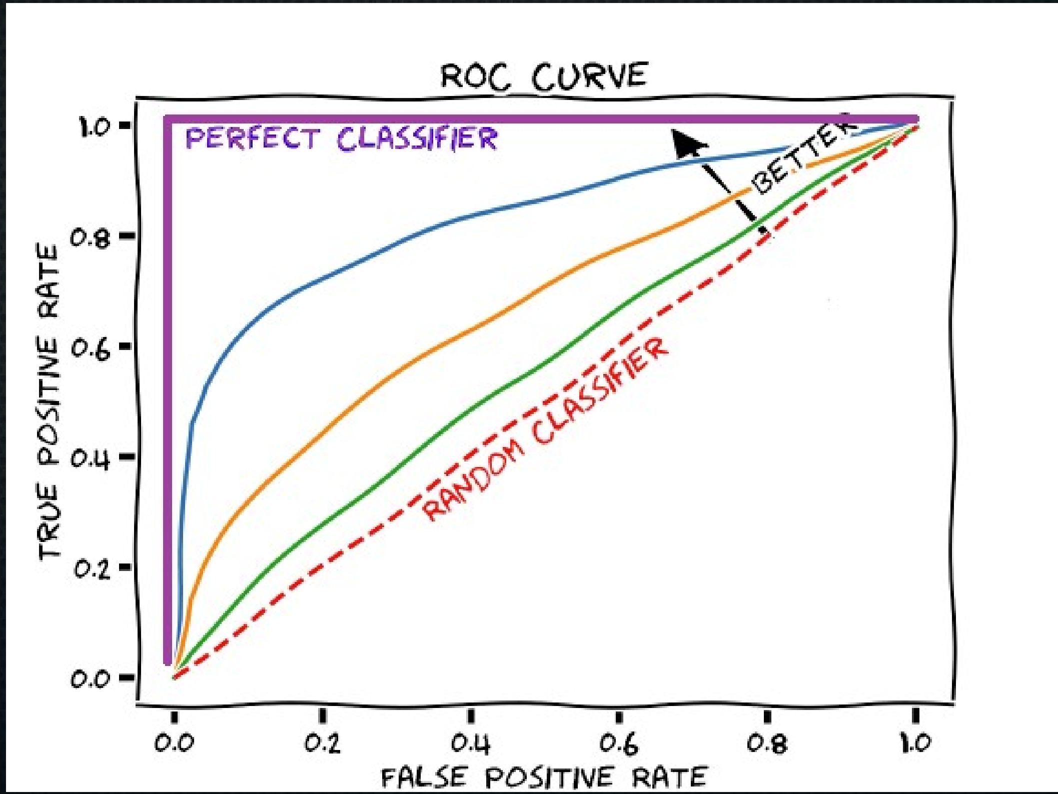
$$\text{if } h_\theta(x) \geq 0.3 \text{ classify as positive}$$
$$\text{else classify as negative}$$





ROC curve

- What if we see how our classifier performs for all possible thresholds?



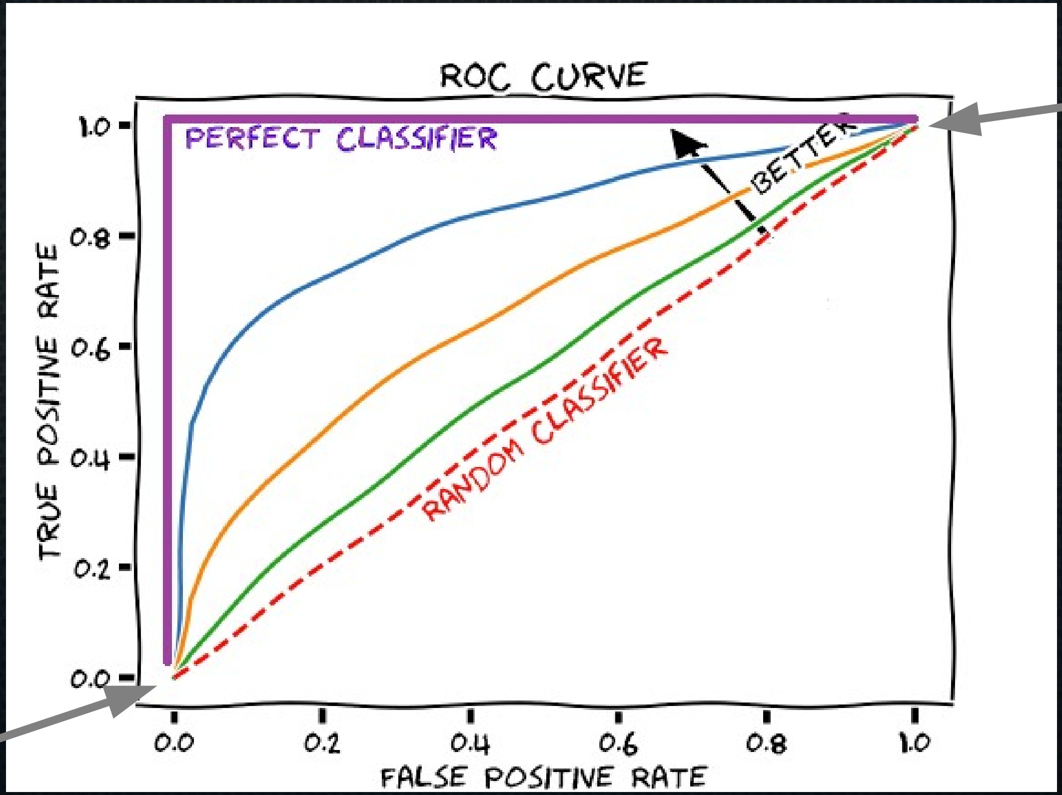
Sensitivity

1-specificity

Source: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

ROC curve

- What if we see how our classifier performs for all possible thresholds?



Sensitivity

1-specificity

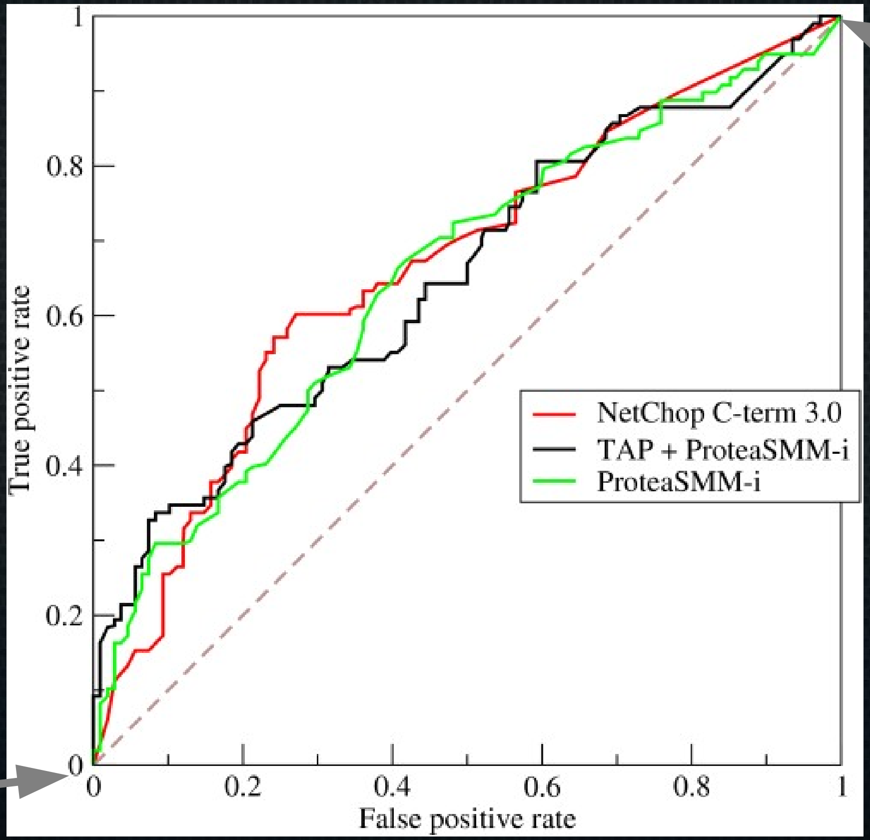
Classify as 1 only if sigmoid >= 0
(i.e. classify everything as positive)

Classify as 1 only if sigmoid > 1
(i.e. classify everything as negative)

Source: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

ROC curve

- What if we see how our classifier performs for all possible thresholds?



Sensitivity

1-specificity

Classify as 1 only if sigmoid >= 0
(i.e. classify everything as positive)

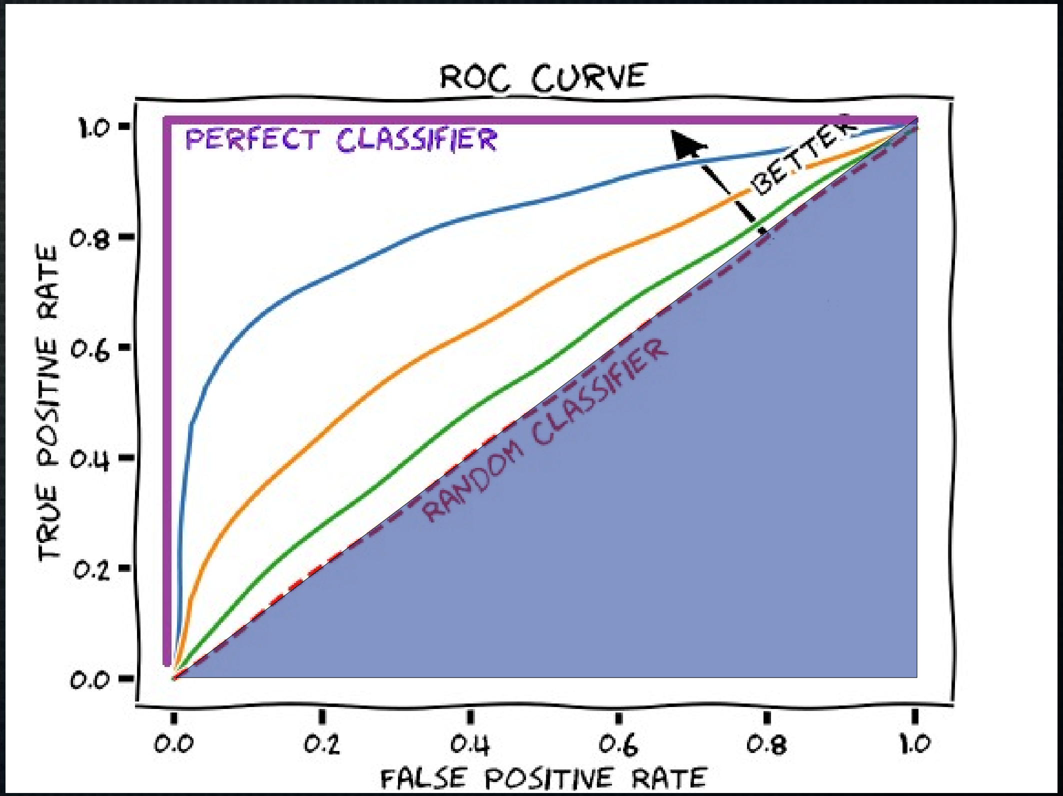
Classify as 1 only if sigmoid > 1
(i.e. classify everything as negative)

Source: https://en.wikipedia.org/wiki/Receiver_operating_characteristic#/media/File:Roccurves.png

Area under the ROC curve (AUC)

- How to compare classifiers numerically?
- Coin-flip classifier (random guess)
  AUC = 0.5

Sensitivity

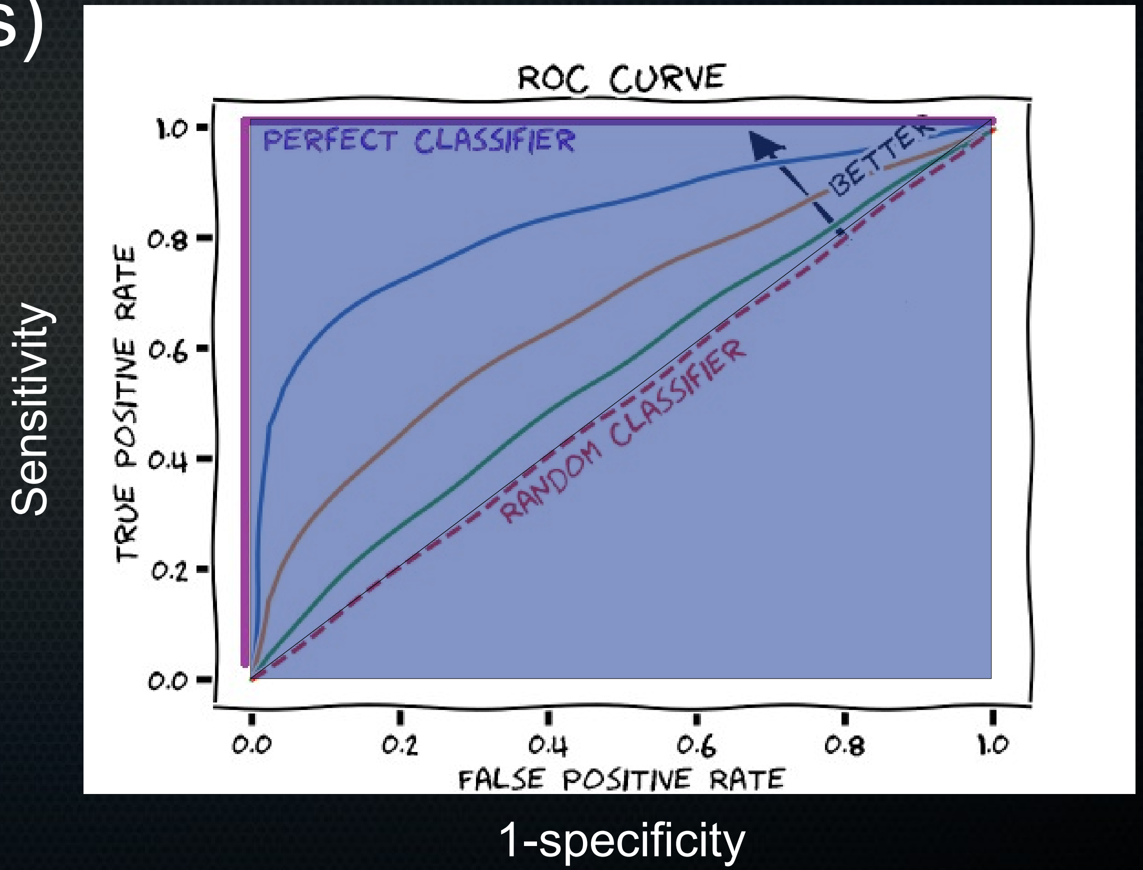


1-specificity

Source: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

Area under the ROC curve (AUC)

- How to compare classifiers numerically?
- Coin-flip classifier (random guess)
  AUC = 0.5
- Best possible classifier (positive cases all predicted 1)
  AUC = 1

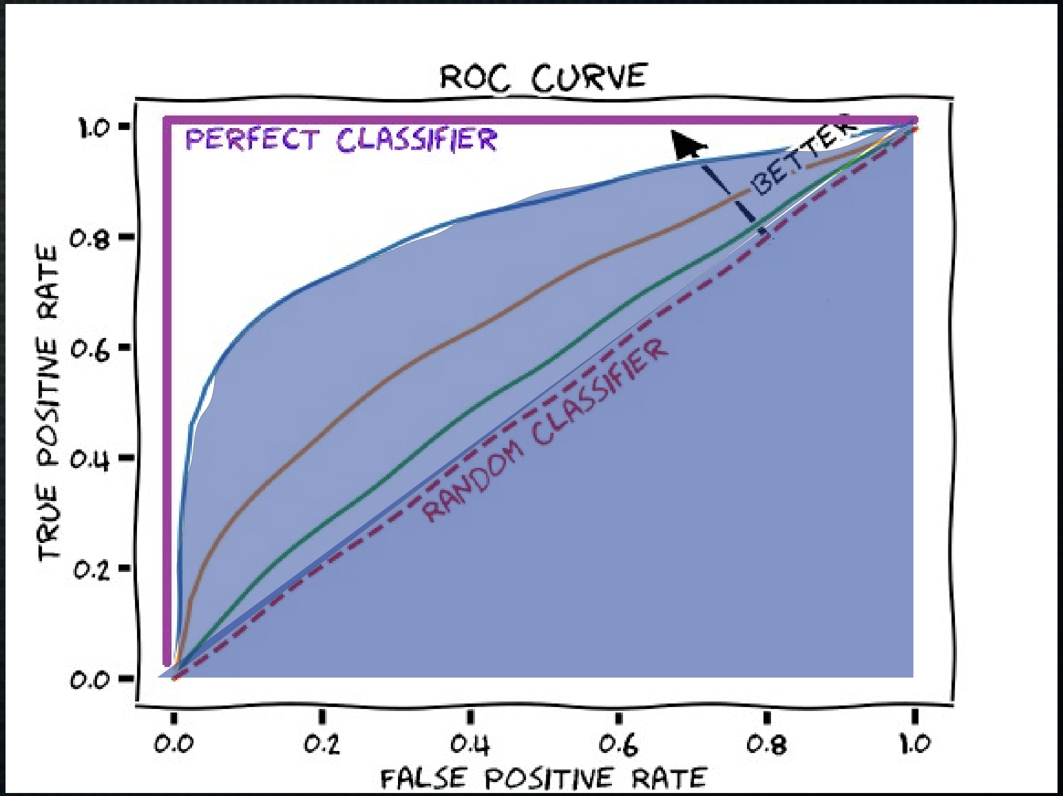


Source: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

Area under the ROC curve (AUC)

- How to compare classifiers numerically?
- Coin-flip classifier (random guess)
  AUC = 0.5
- Best possible classifier (positive cases all predicted 1)
  AUC = 1
- In-between: ~0.8, for instance

Sensitivity



1-specificity

Source: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

Pitfall AUC (of ROC curve)

- All these values are determined by ratios
- If data is sampled from general population: $N_{negative} >> N_{positive}$ for a disease
- For specificity, because true negatives is a *huge number*, more false positives matter much less. Specificity becomes overly optimistic (especially if, later, you run your classifier in a clinical setting where $N_{negative}$ is much smaller!)

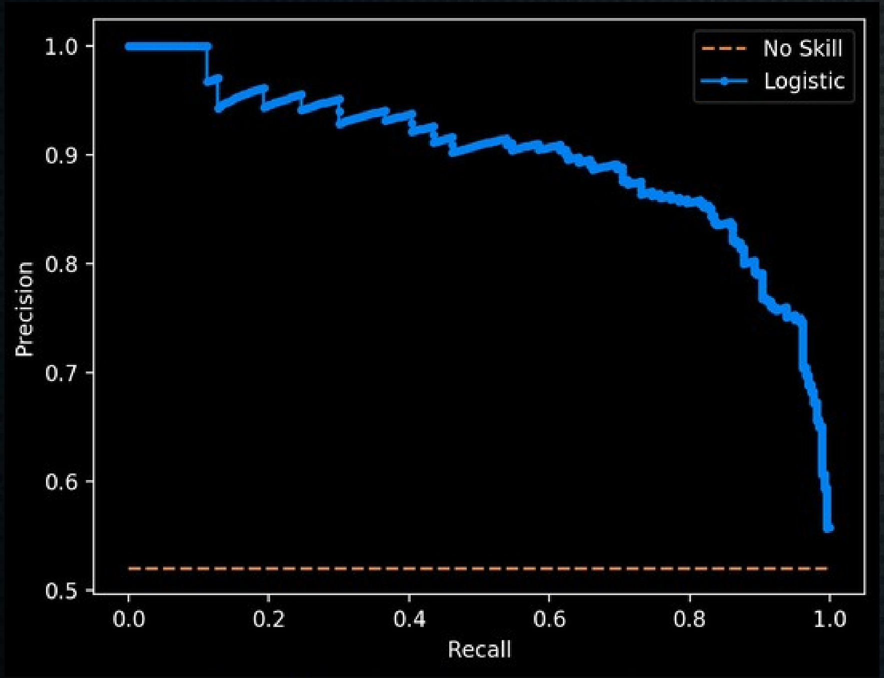
		Reality	
		Cancer	**Not Cancer**
Model Prediction	**Cancer**	True Positive	False Positive
	Not Cancer	False Negative	True Negative

Sensitivity (true positive rate) $\dfrac{True\ Positives}{True\ Positives + False\ Negatives}$

Specificity (true negative rate) $\dfrac{True\ Negatives}{True\ Negatives + False\ Positives}$

Pitfall AUC (of ROC curve)

- Instead, should then look at precision-recall curve.



Precision-Recall Plot for a No Skill Classifier and a Logistic Regression Model

Source: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

		Reality	
		Cancer	Not Cancer
Model Prediction	Cancer	True Positive	False Positive
	Not Cancer	False Negative	True Negative

Recall = Sensitivity (true positive rate) $\frac{True\ Positives}{True\ Positives + False\ Negatives}$

Specificity (true negative rate) $\frac{True\ Negatives}{True\ Negatives + False\ Positives}$

$$\underbrace{\frac{True\ Positives}{(True\ Positives + False\ Positives)}}_{\text{Precision}}$$

- Or make sure to train on balanced data!

ADD REGULARISATION HERE

Summary

- Multi-class classification: simply train *n* independent binary logistical regressors for your *n* classes, run them all on the data, pick for each sample the class with the highest probability over the regressors
- Performance metrics: Accuracy, sensitivity and specificity, ROC curve and ROC AUC (or PRC AUC for imbalanced data)
- Regularisation: add a cost to making parameters too large (i.e. fitting them too precisely to the data). Forces the model to only increase those parameters that really improve the fit (less fine-tuning exactly to the training data)!

BREAK FOR PRACTICAL