Evaluation Metrics

Evaluation Metrics

Thus far, for classification problems we've been primarily concerned with the misclassification error rate and the standard definition of accuracy that comes with it

$$\mathrm{Err} = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i \neq y_i)$$

And for many classification tasks, this makes perfect sense

In fact, many classification techniques are designed specifically to minimize this error rate

But there are many scenarios in which the misclassification error rate can be misleading

Evaluation Metrics

Consider the case when your training set is heavily skewed towards a particular class

- If 98% of training data is from the negative class, should you feel good about a model with a 98.5% classification accuracy?

What about when there are different consequences for false positives vs a false negatives?

Can you think of specific examples of this case?

(FP)

SELF DRIVING CAR → PREDICT PERSON WHEN NOT

SECURITY → ALLOW NEFARIOUS PERSON log-on (FP)

PACEMAKERS → JOLT WHEN COUCH (FP)

NO JOLT WHEN H.A. (FN)

Evaluation Metrics

Consider the case when your training set is heavily skewed towards a particular class

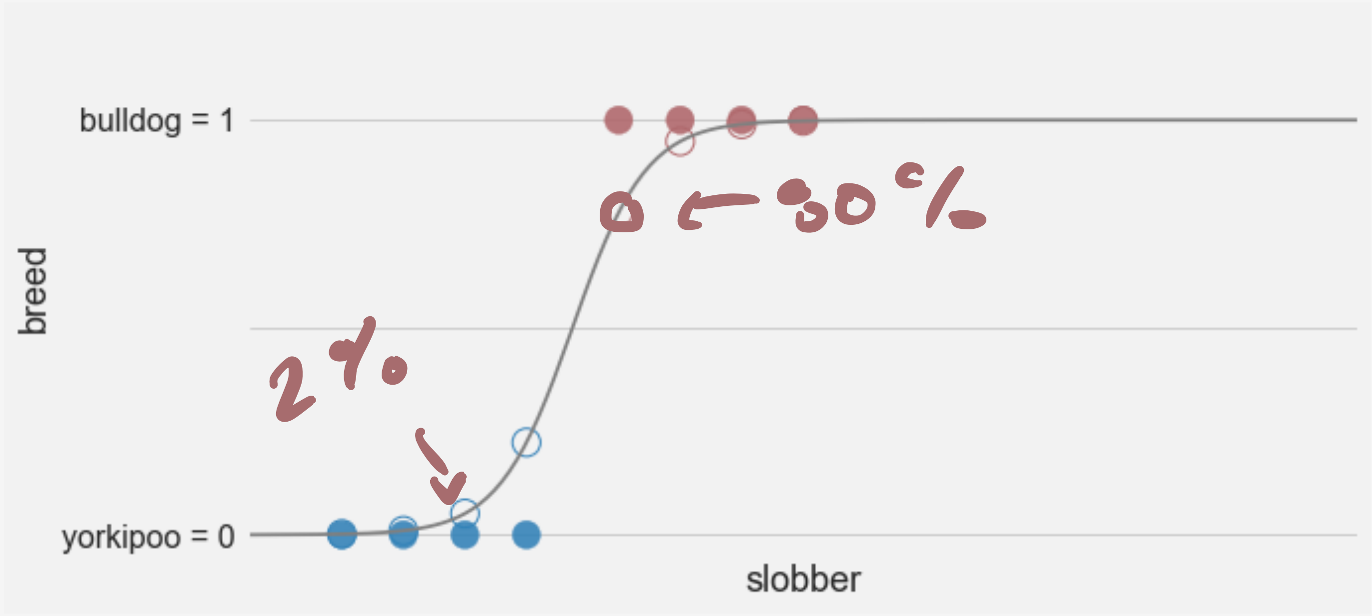
Consider the case that there are different consequences for false positives vs a false negatives?

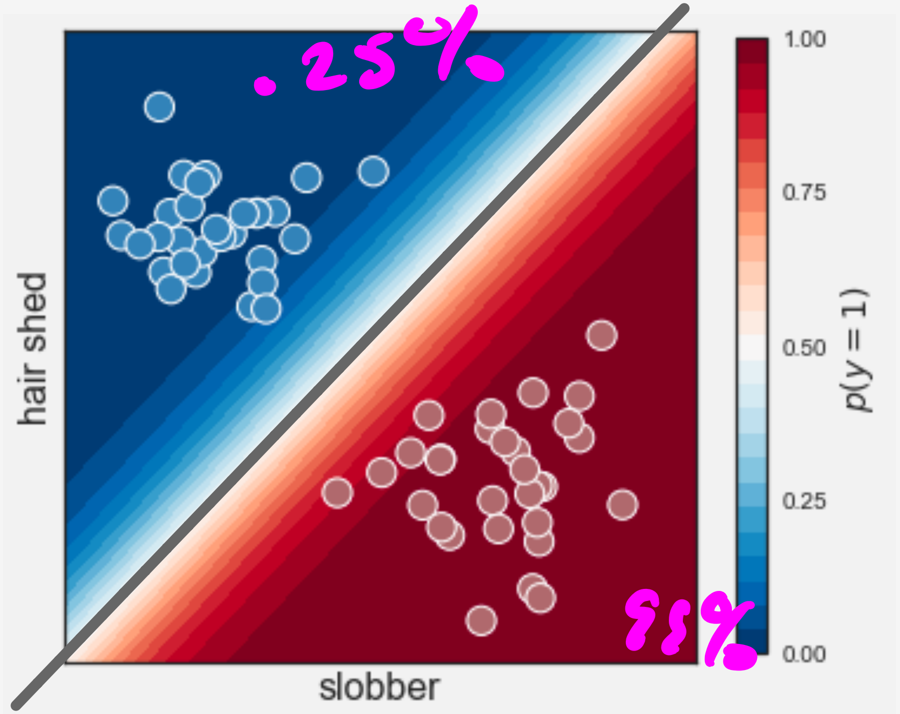
Can you think of ways that we could use our own personal judgement to mitigate these problems when we're using a classifier like **Logistic Regression**?

Logistic Regression Refresher

Decision rule based on a probability: $p(y = 1 \mid \mathbf{x}) = \mathrm{sigm}(\beta^T \mathbf{x})$

Large prob (0.99) implies confidence in Class 1. Low prob (0.01) implies confidence in Class 0



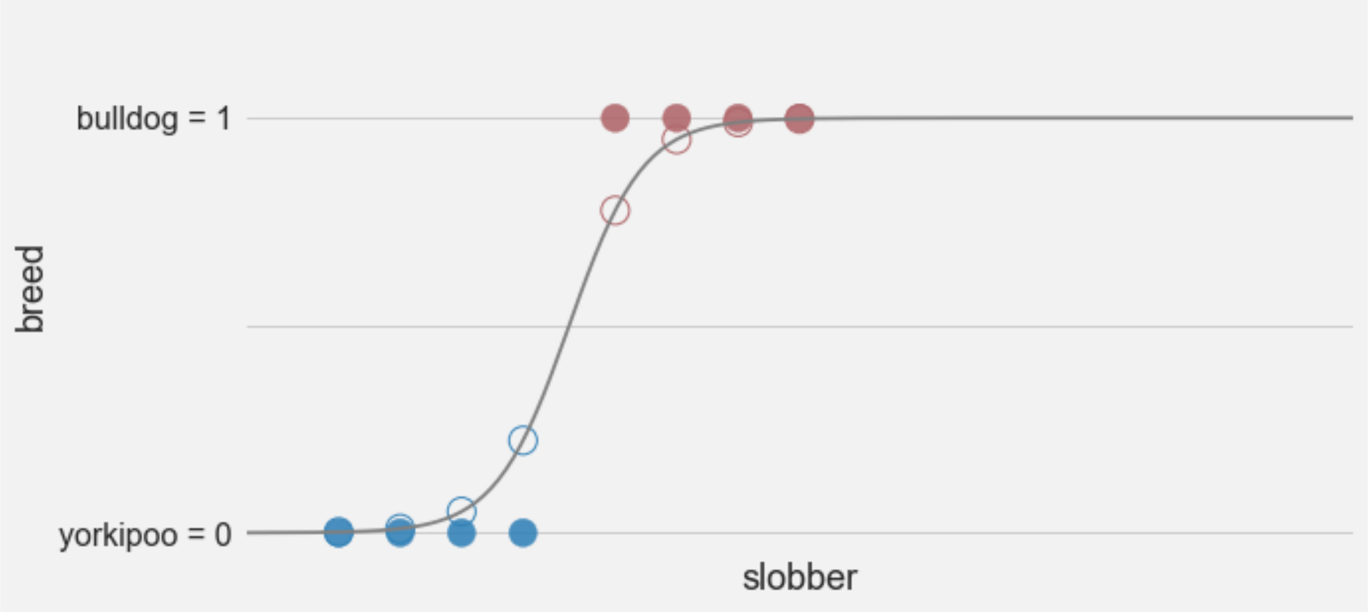


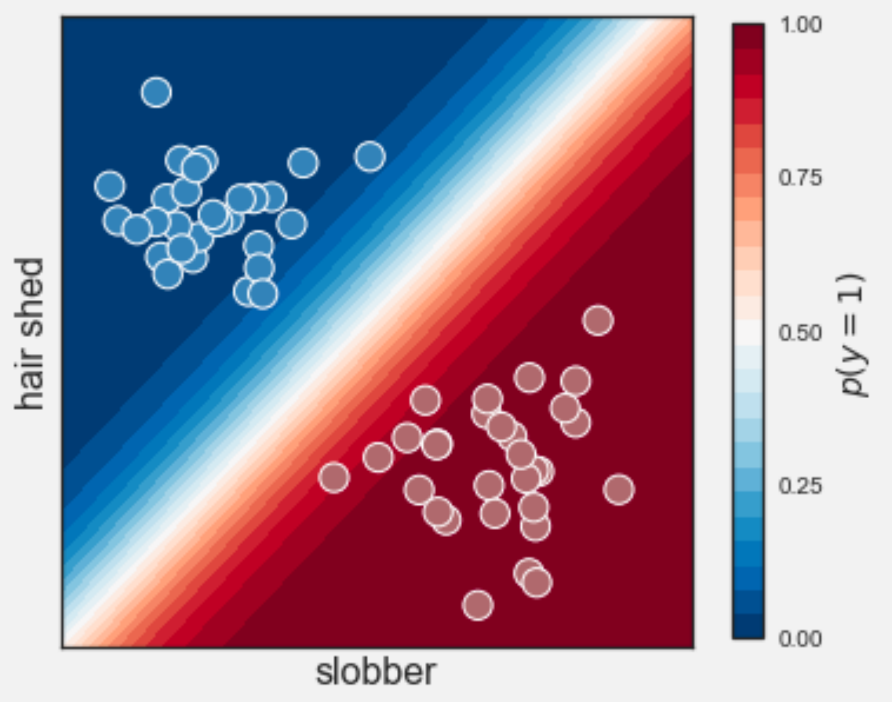
In certain cases we might want to juke the decision threshold from the usual 0.5

Logistic Regression Refresher

Decision rule based on a probability: $p(y = 1 \mid \mathbf{x}) = \text{sigm}(\boldsymbol{\beta}^T \mathbf{x})$

Large prob (0.99) implies confidence in Class 1. Low prob (0.01) implies confidence in Class 0



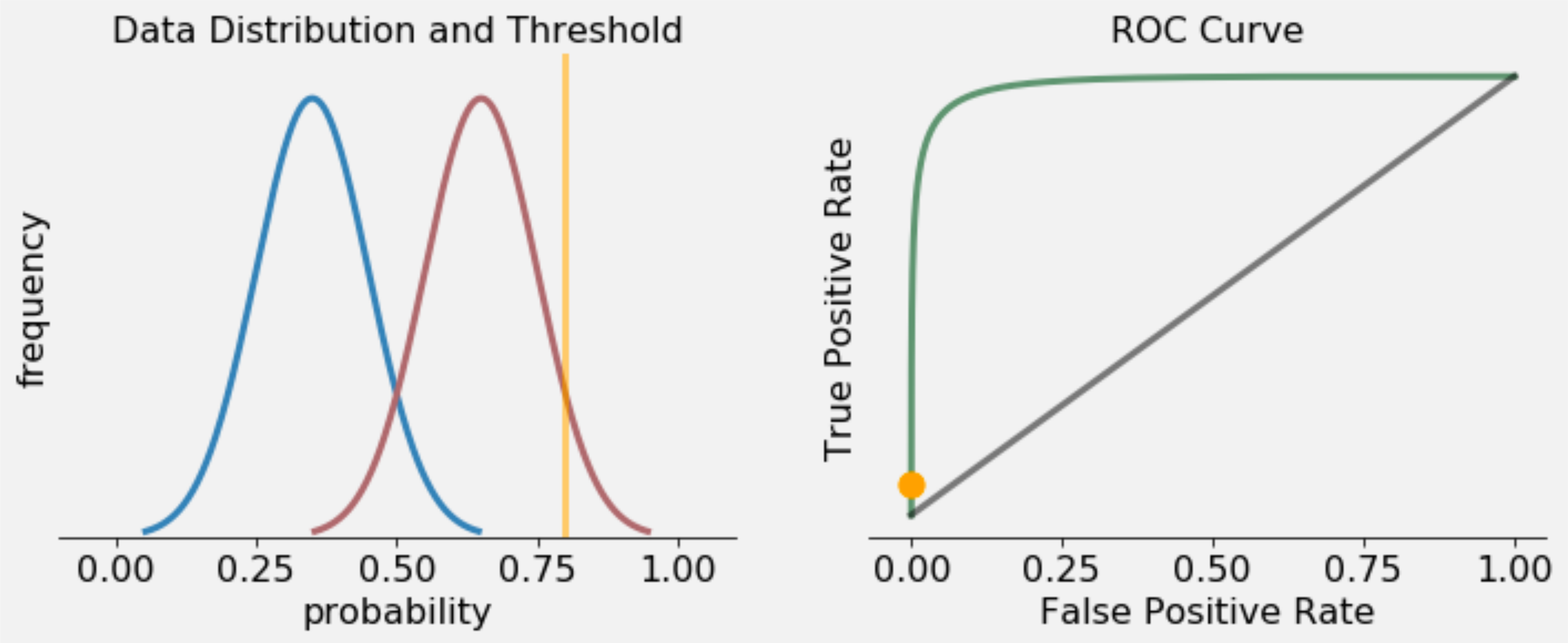


But which threshold is best? Tricky if you have class imbalances or consequences to consider.

The ROC Curve

The Receiver Operating Characteristic Curve gives us convenient way to evaluate thresholds

Requires a binary classifier (kinda) that ranks predictions in terms of confidence (probability)



The ROC Curve

Consider the case when your training set is heavily skewed towards a particular class

Everything starts from the confusion matrix:

	Predicted Positive	Predicted Negative
Actually Positive	true positive (TP)	false negative (FN)
Actually Negative	false positive (FP)	true negative (TN)

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

The ROC Curve

Everything starts from the confusion matrix:

	Predicted Positive	Predicted Negative
Actually Positive	true positive (TP)	false negative (FN)
Actually Negative	false positive (FP)	true negative (TN)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{TP + FP + FN + TN}$$

The ROC Curve

Everything starts from the confusion matrix:

	Predicted Positive	Predicted Negative
Actually Positive	true positive (TP)	false negative (FN)
Actually Negative	false positive (FP)	true negative (TN)

$$\text{True Positive Rate} = p(\text{predict Pos} \mid \text{is Pos}) = \frac{TP}{FN + TP}$$

$$\text{False Positive Rate} = p(\text{predict Pos} \mid \text{is Neg}) = \frac{FP}{FP + TN}$$

The ROC Curve

Example: Suppose you build a classifier to predict credit card fraud from recent transactions

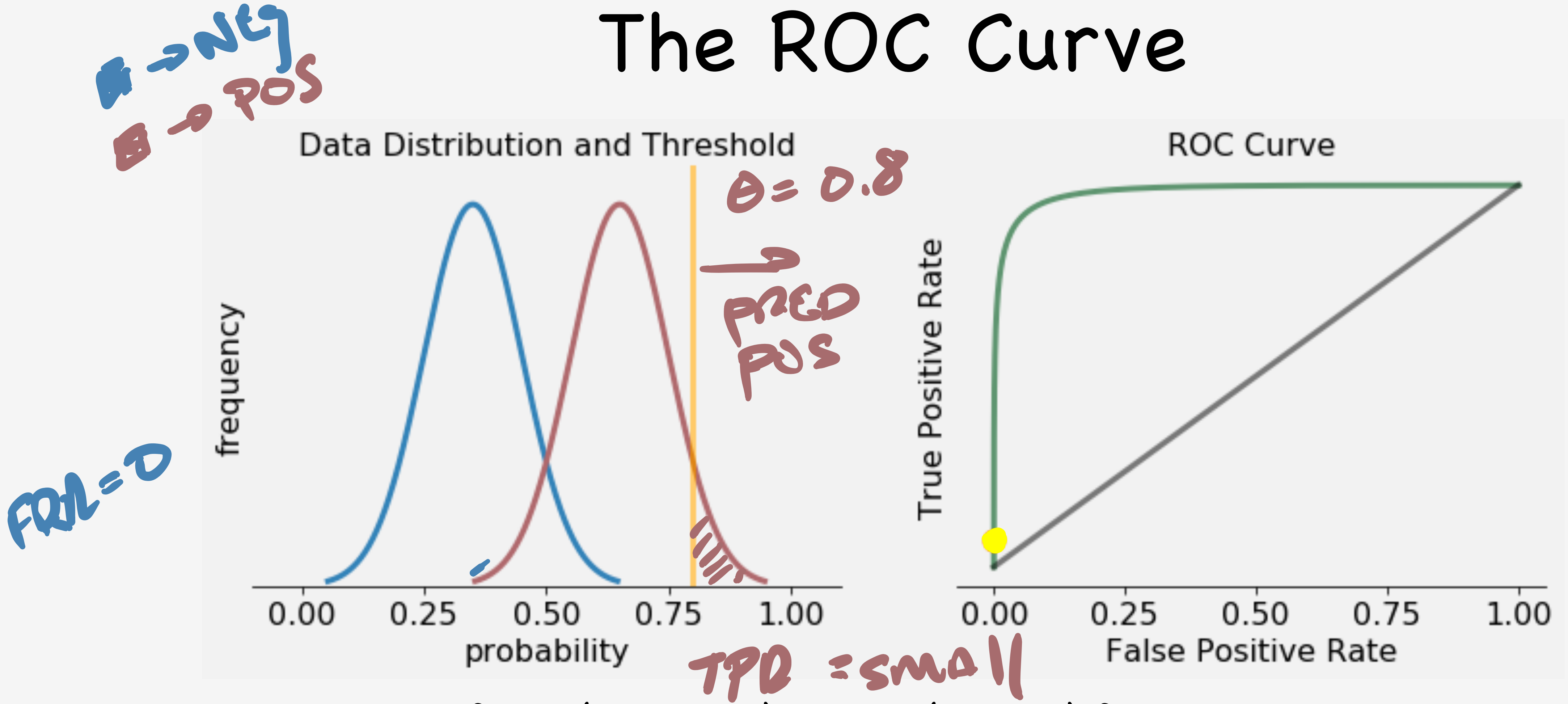
Customers would rather be warned even when things are OK than let actual fraud be missed

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

This means we're willing to accept a high FPR in order to secure a high TPR

A ROC Curve gives us a visual way to evaluate suitable thresholds to fit our needs

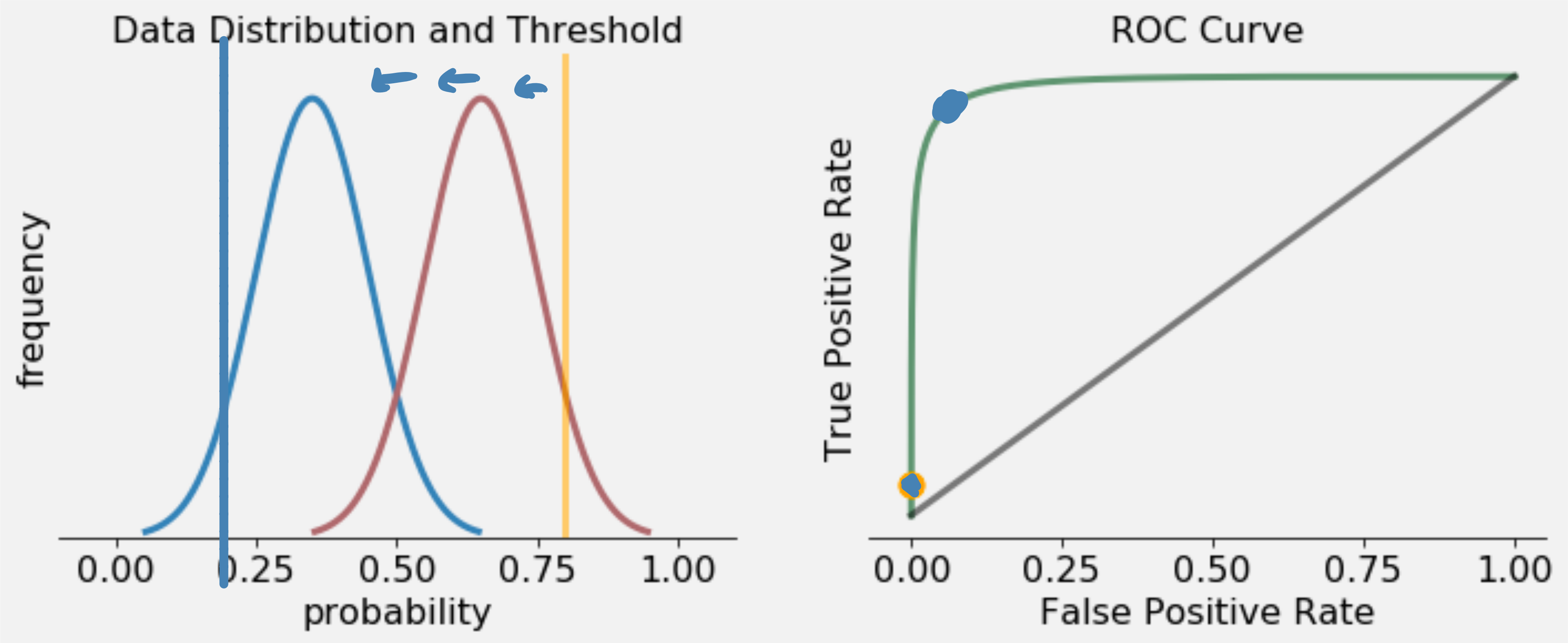
The ROC Curve



A ROC Curve is a plot of FPR (horizontal) vs. TPR (vertical) fall all possible threshold values

Extremely convenient to see how model would perform at all thesholds simultaneously, rather than looking at misclassification rate for thresholds individually

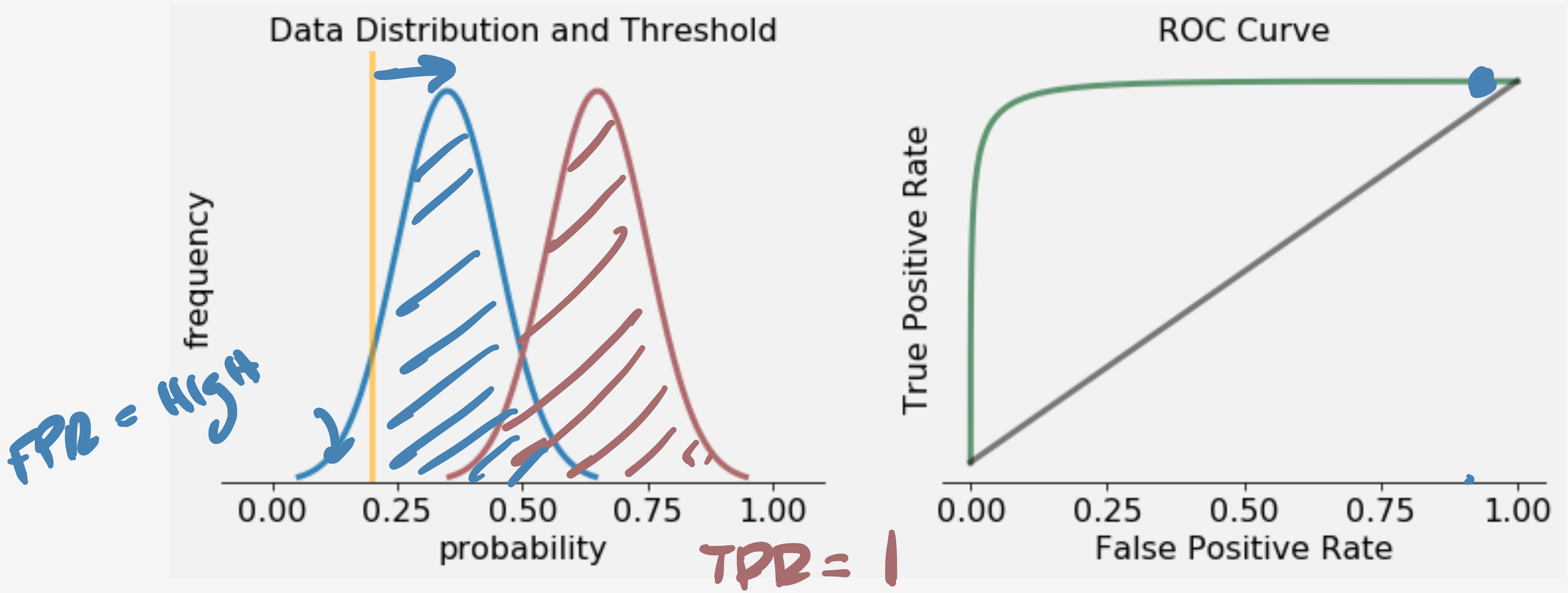
The ROC Curve



A ROC Curve is a plot of FPR (horizontal) vs. TPR (vertical) fall all possible threshold values

Extremely convenient to see how model would perform at all thesholds simultaneously, rather than looking at misclassification rate for thresholds individually

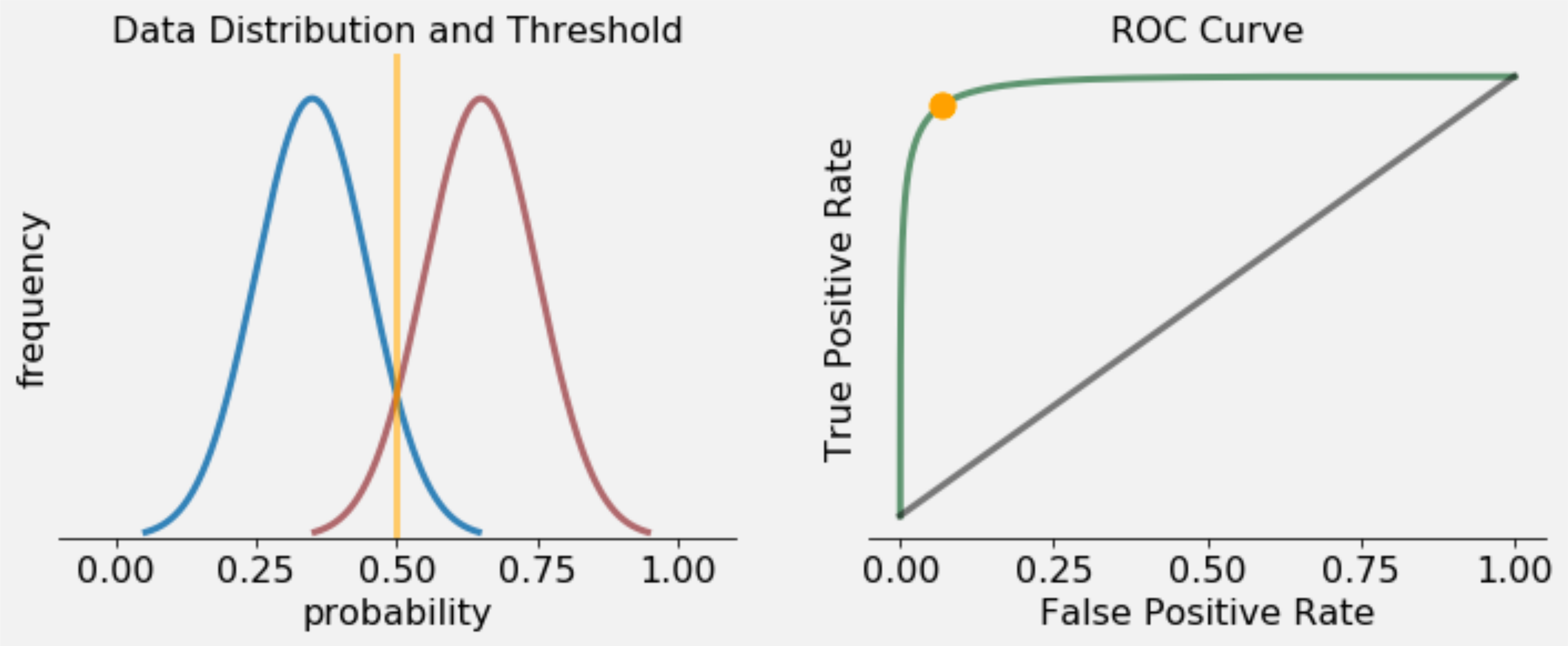
The ROC Curve



A ROC Curve is a plot of FPR (horizontal) vs. TPR (vertical) fall all possible threshold values

Extremely convenient to see how model would perform at all thesholds simultaneously, rather than looking at misclassification rate for thresholds individually

The ROC Curve

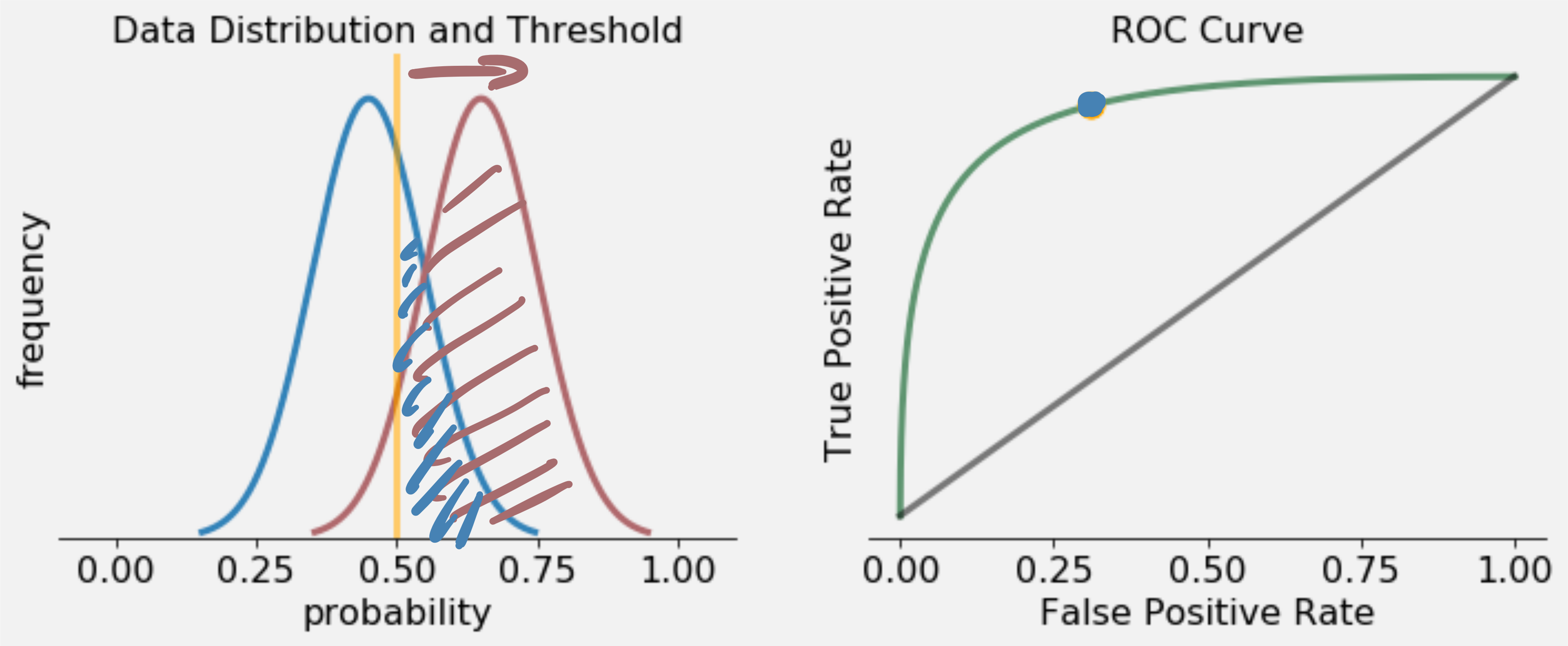


The threshold gives the parameterization of the ROC curve (i.e. it moves the dot)

When the threshold separates the two classes fairly well, the curve is far away from diagonal

What happens if we can't separate the classes very well?

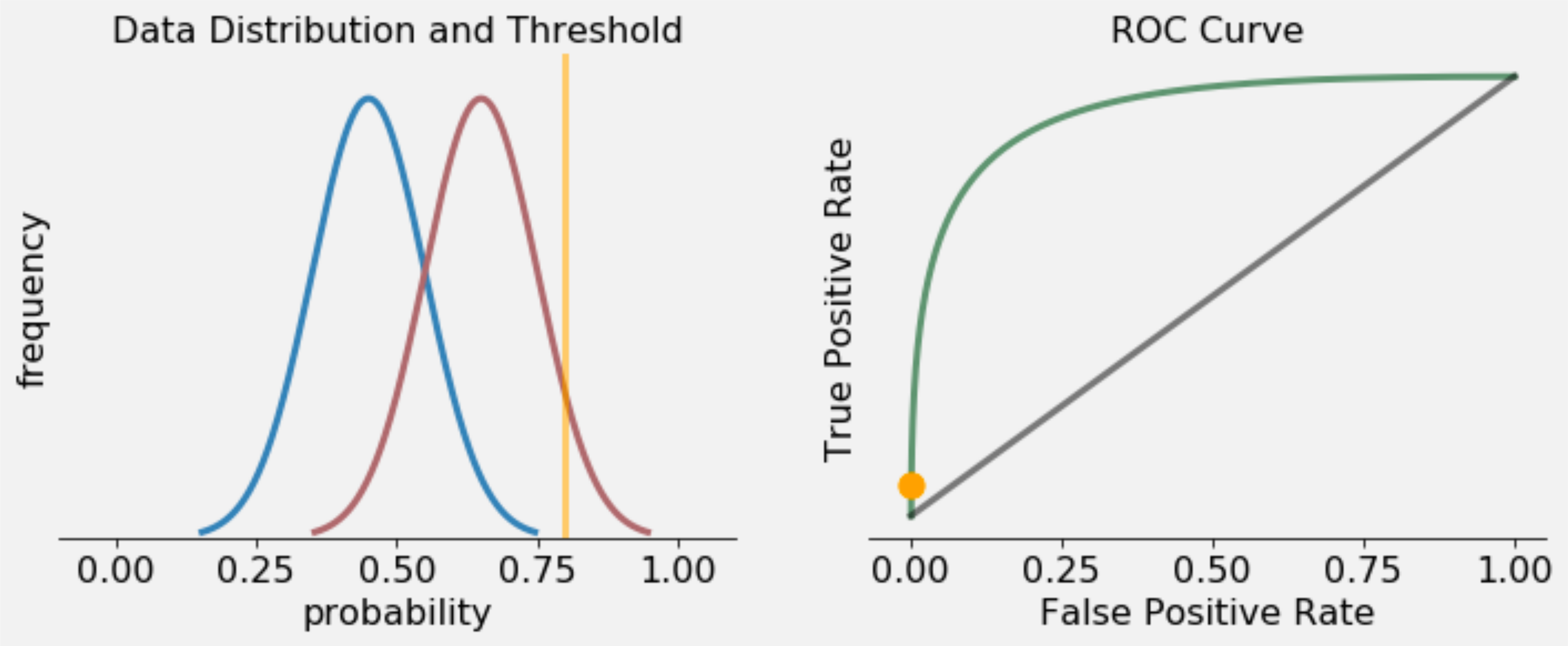
The ROC Curve



Now we're not doing so well at separating the classes

The ROC curve starts bending towards the center

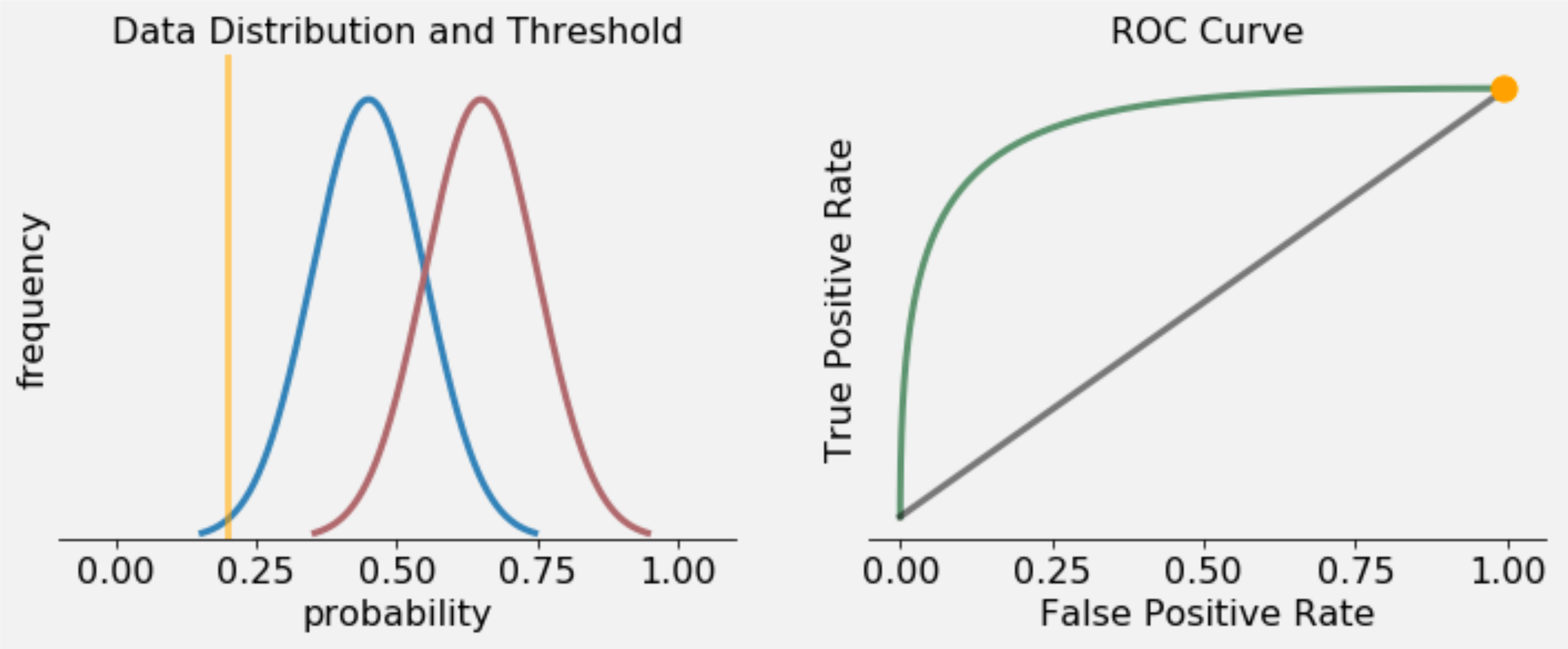
The ROC Curve



Now we're not doing so well at separating the classes

The ROC curve starts bending towards the center

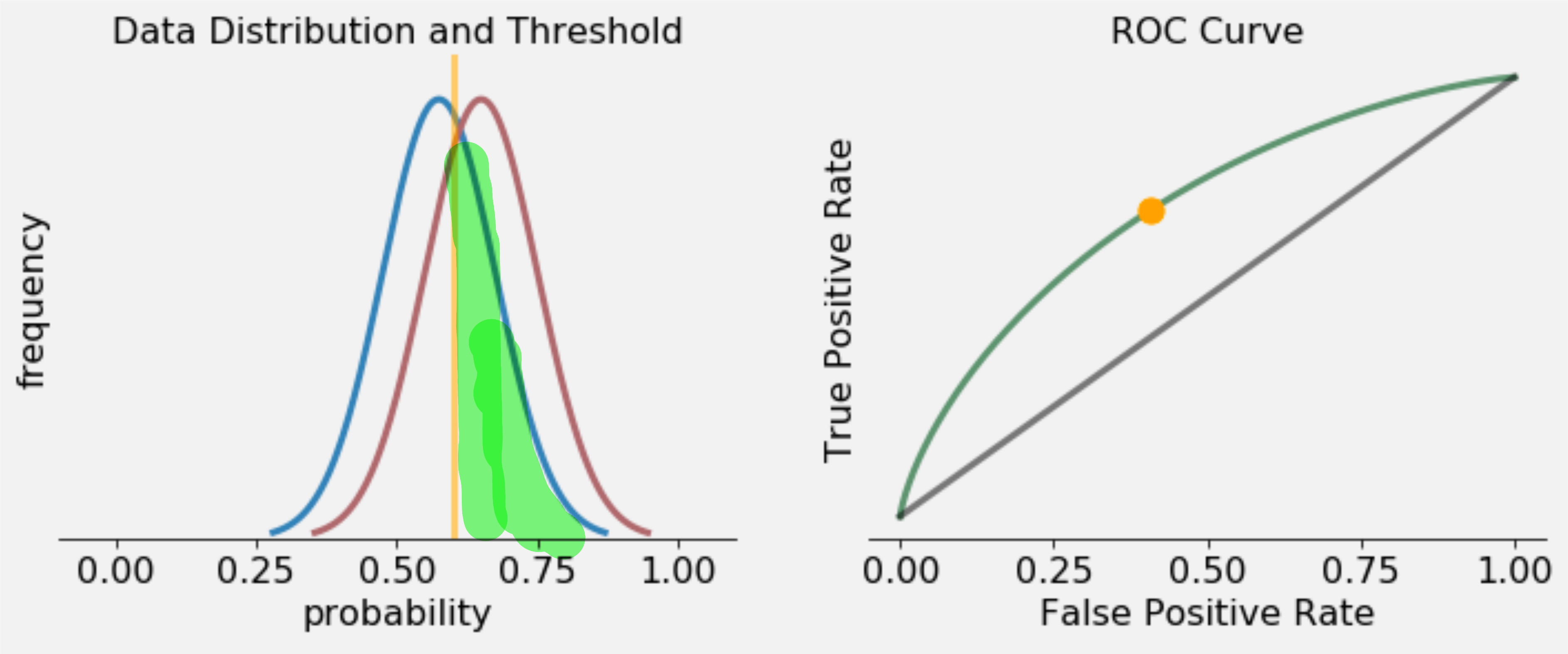
The ROC Curve



Now we're not doing so well at separating the classes

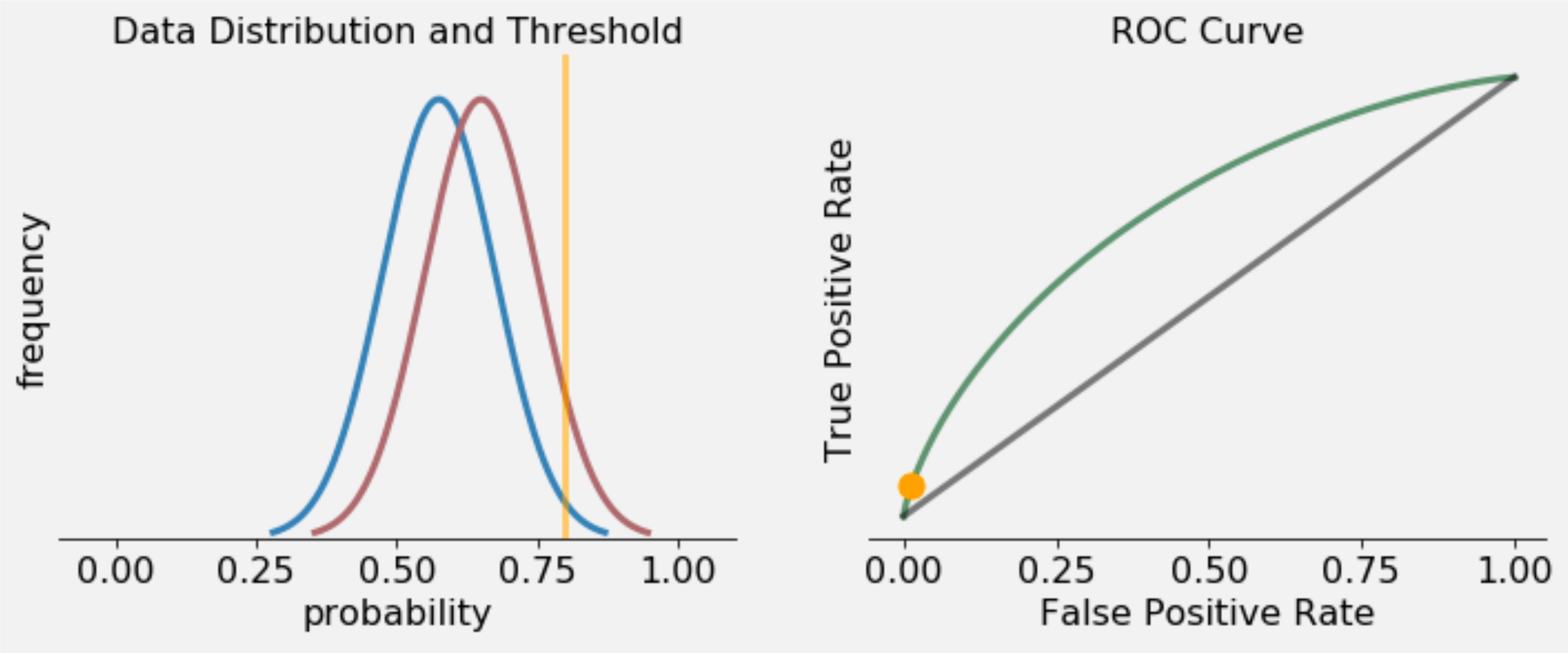
The ROC curve starts bending towards the center

The ROC Curve



And as we do a poorer job of separating the classes …

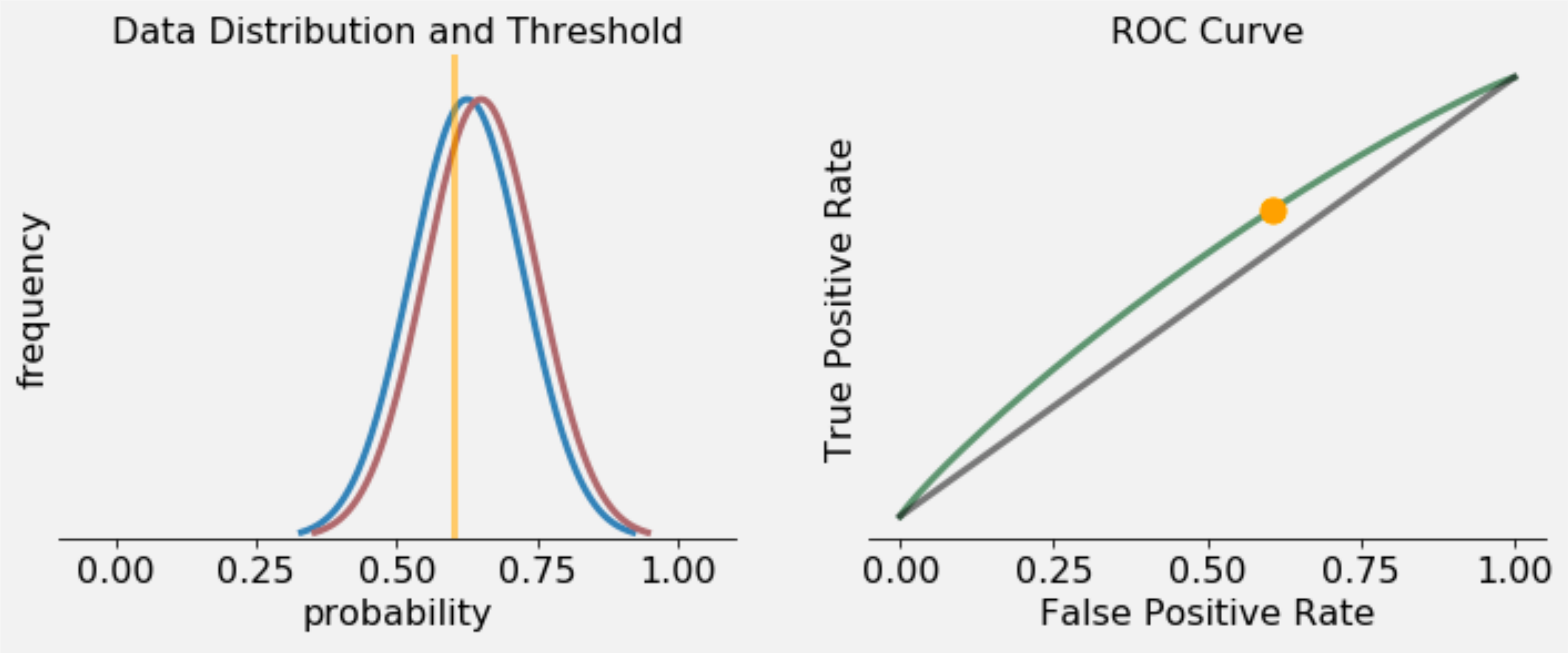
The ROC Curve



And as we do a poorer job of separating the classes …

The curve continues to bend

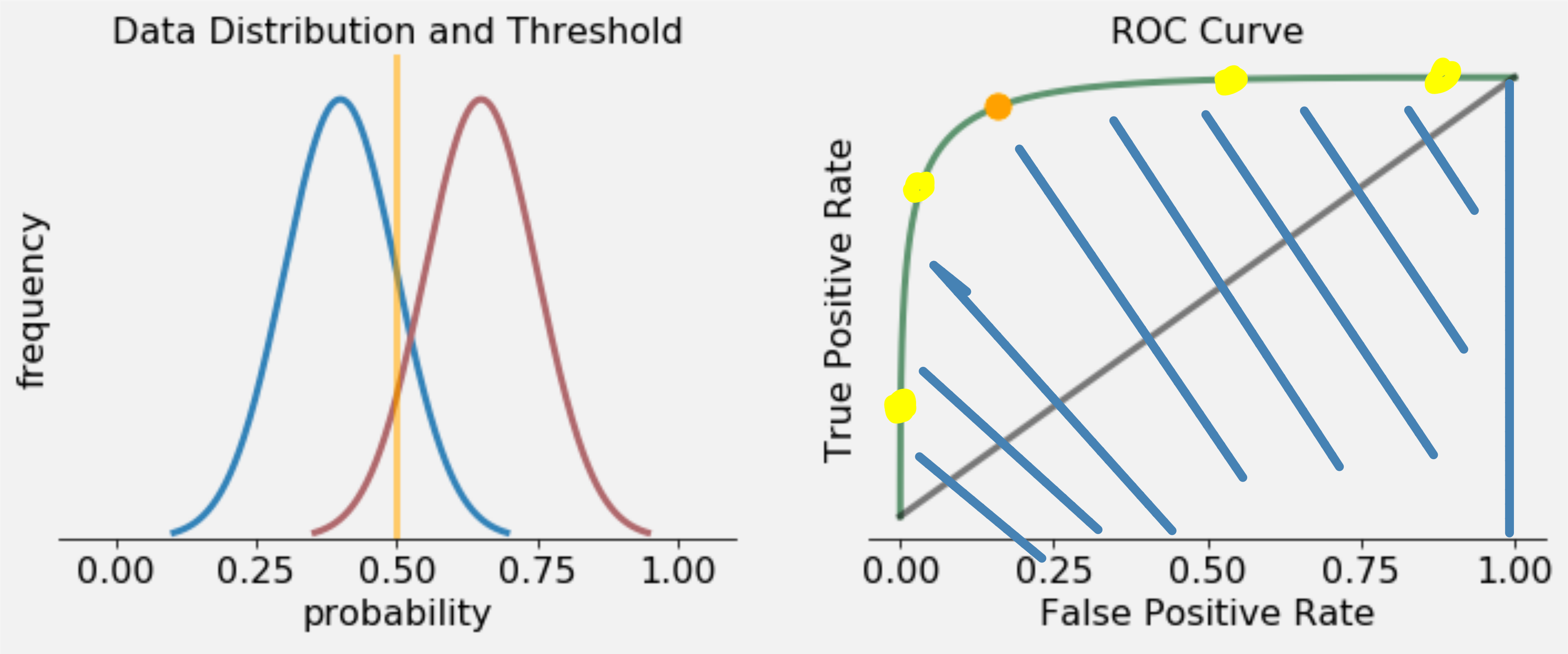
The ROC Curve



And if we do a terrible job, the curve approaches the random chance line

Indicating that our classifier is not much better than a random guess

The ROC Curve

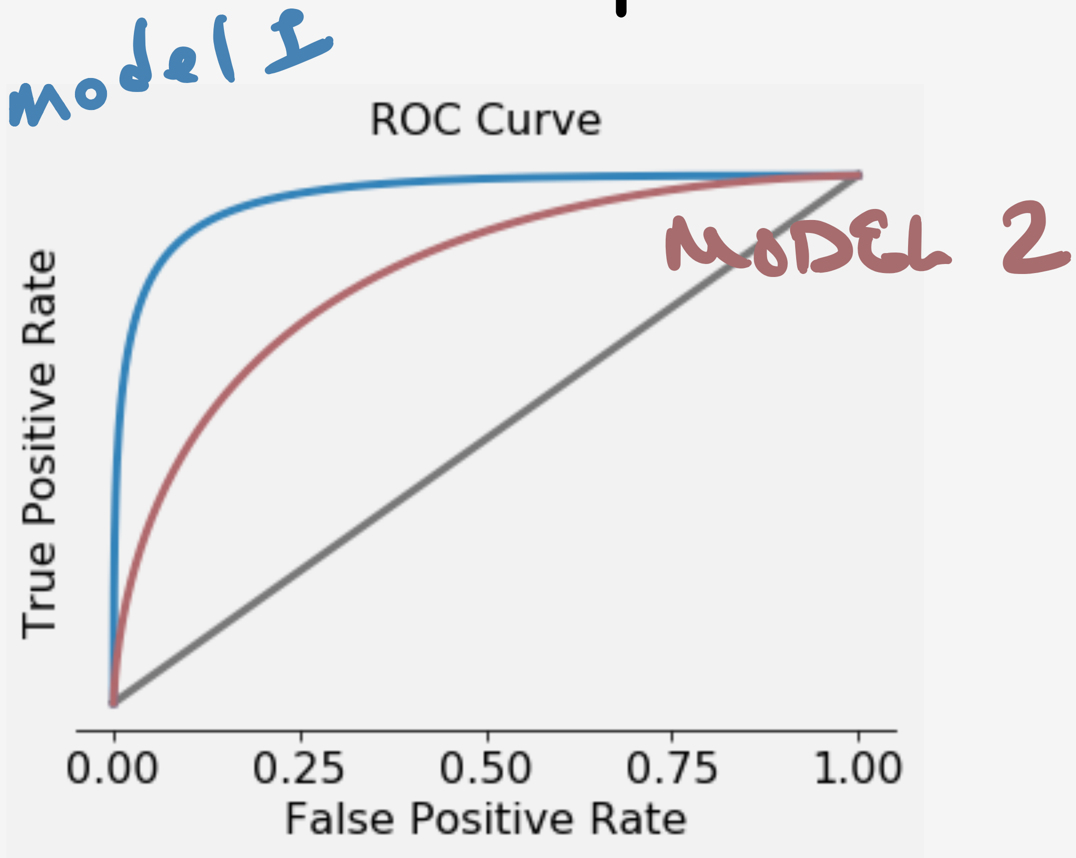


The ROC curve addresses the cases when we're worried about FPs and TPs simultaneously

But, if you want a single number, evaluating how the model will do in all cases

You can compute the AUC (Area Under the ROC Curve)

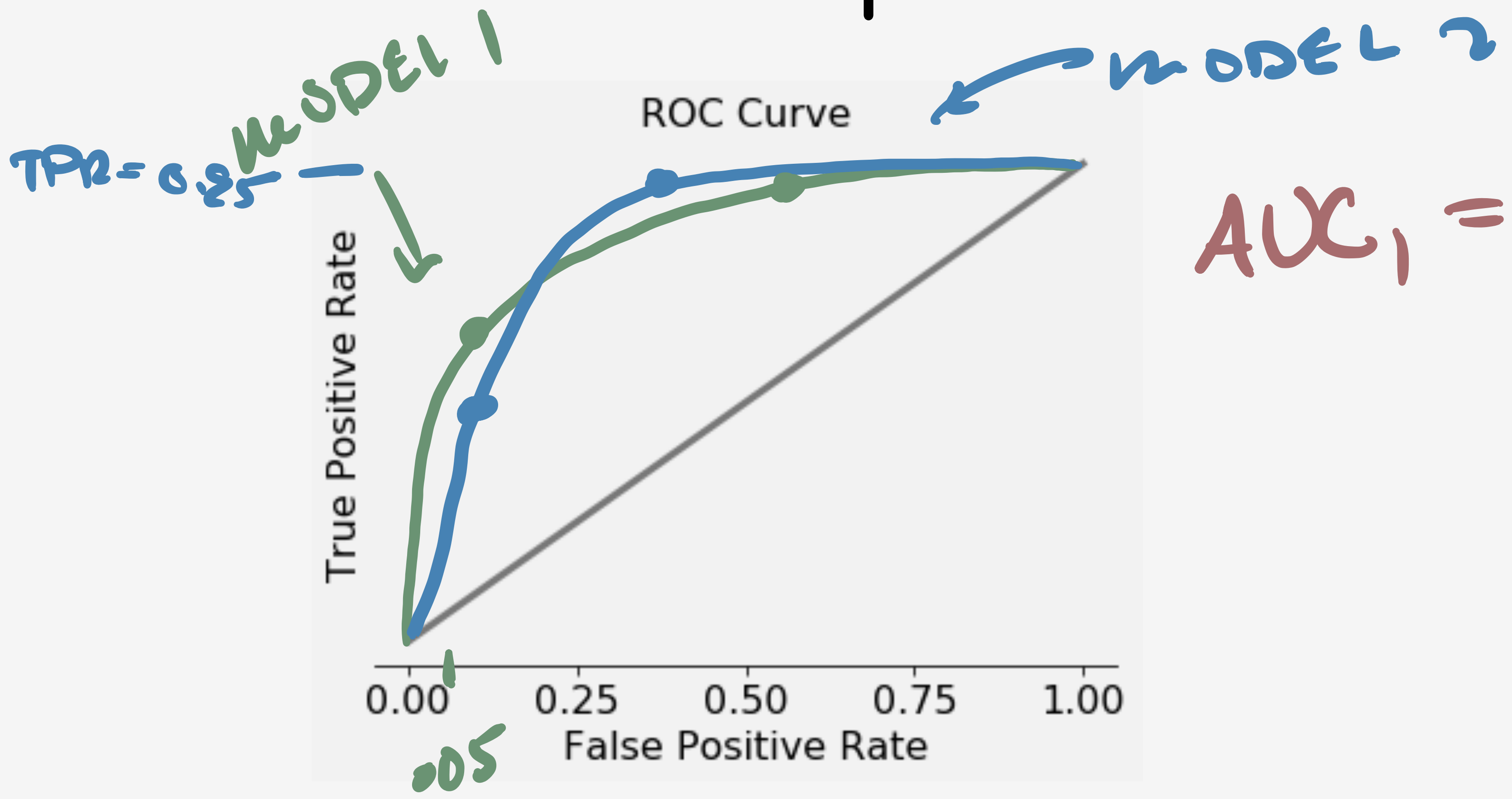
ROC-AUC to Compare Models



To compare two models, plot their ROC curves on the same axes

If one encloses the other, then it's better on both ends of the spectrum, and has higher AUC

ROC-AUC to Compare Models

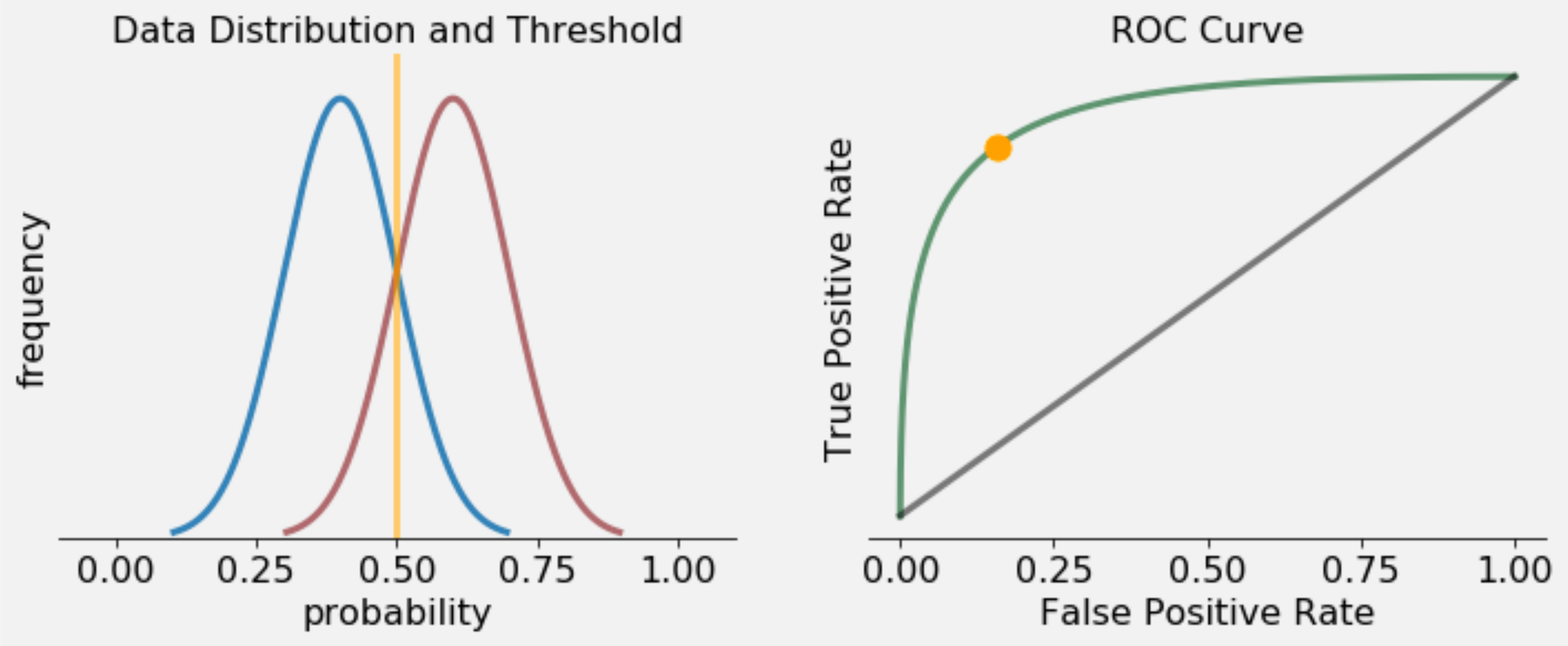


$AUC_1 = AUC_2$

To compare two models, plot their ROC curves on the same axes

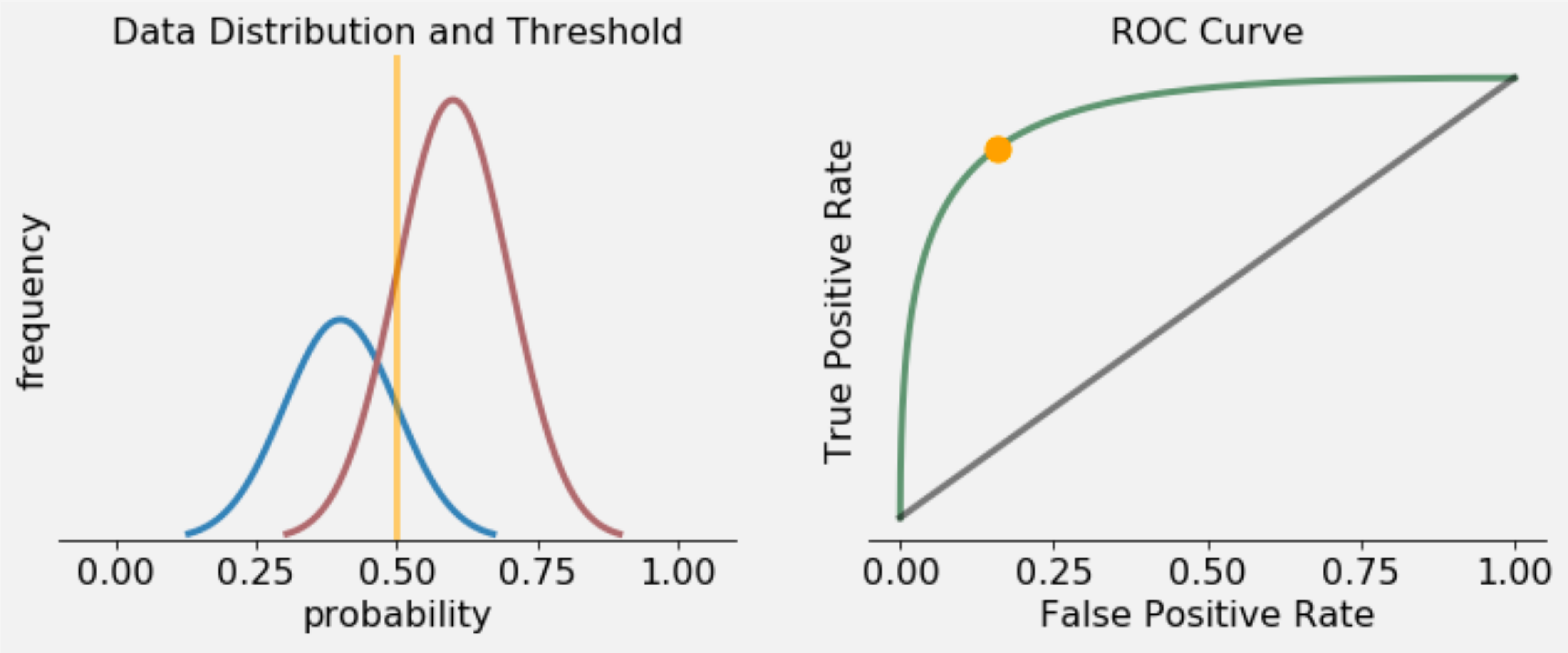
But if they have similar AUCs, the plot may show that one will do better on one end of the spectrum, and the other on the other end of the spectrum

The ROC Curve



And here's my favorite part

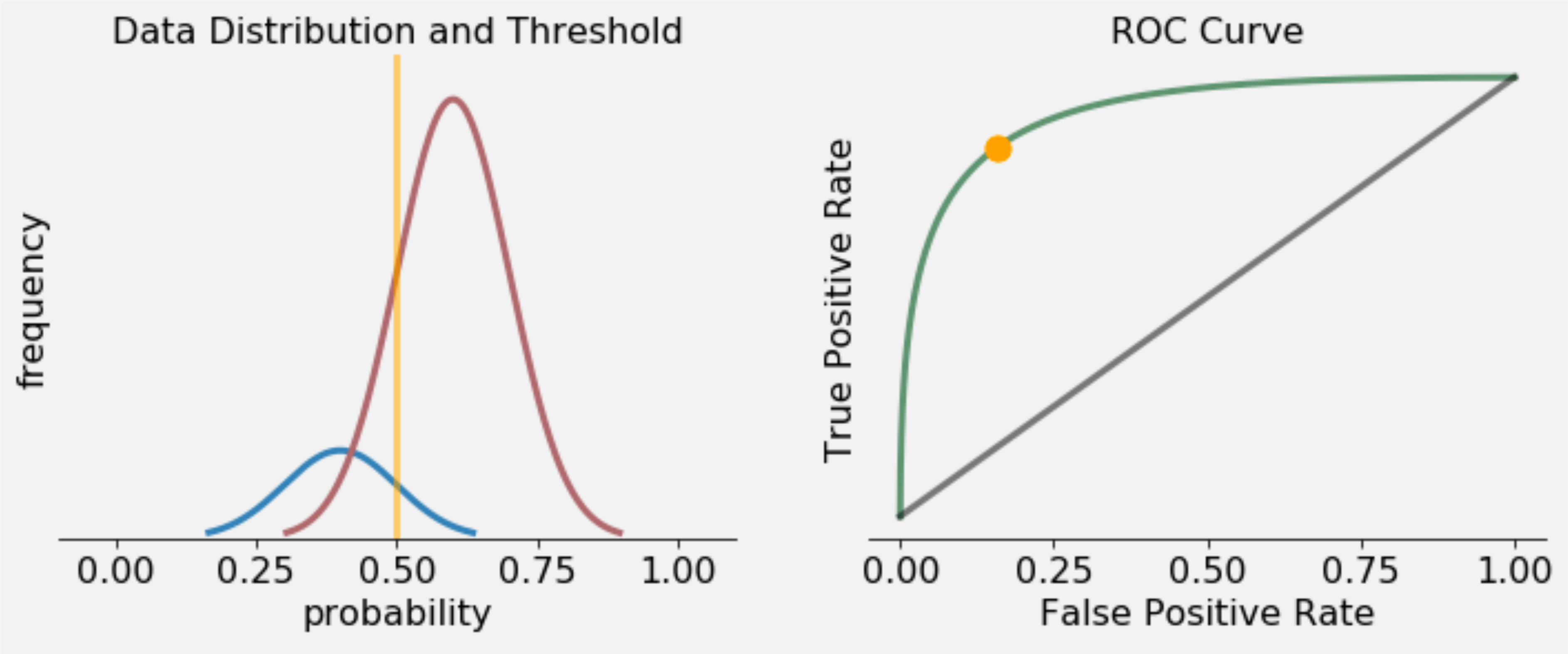
The ROC Curve



And here's my favorite part

Since it's based on proportions of pos-to-pos and neg-to-neg predictions, ROC is insensitive to skewed class imbalances

The ROC Curve



And here's my favorite part

Since it's based on proportions of pos-to-pos and neg-to-neg predictions, ROC is insensitive to skewed class imbalances

Constructing a ROC Curve

You need a classifier that is able to rank examples by confidence (probability)

- Order all examples by prediction confidence
- Move threshold to each point, one at a time
- If point is true positive, move vertically (1/NP)
- If point is true negative, move horizontally (1/NN)

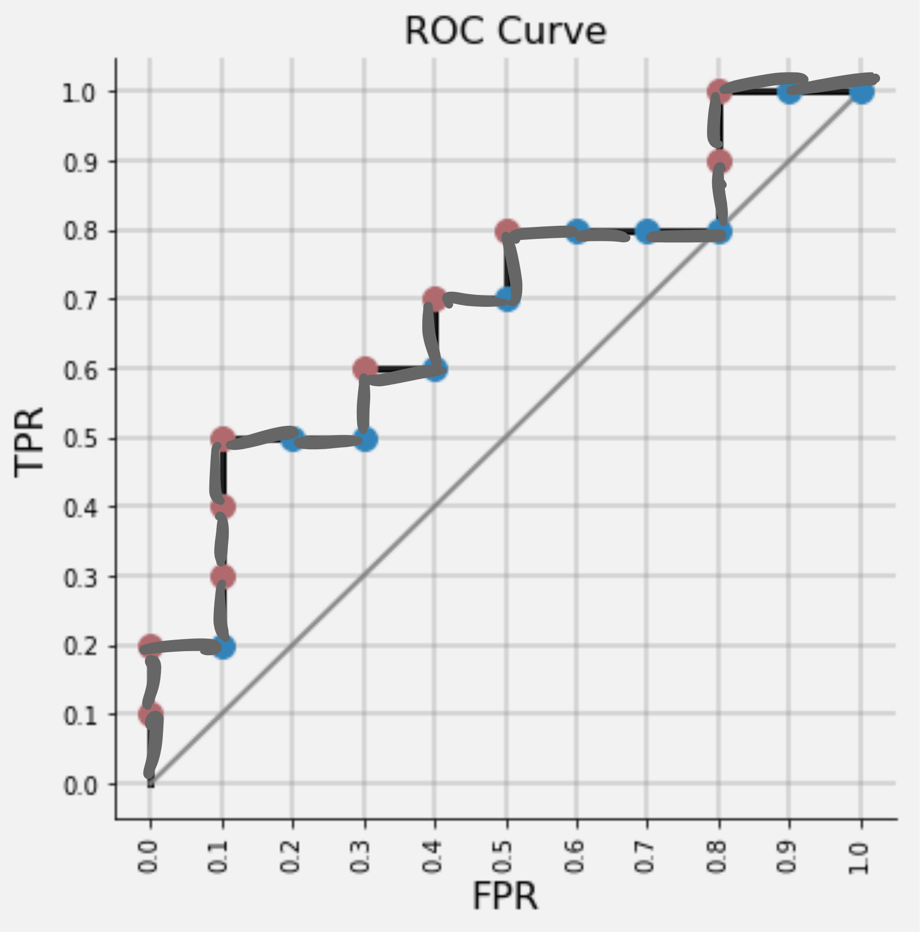
#	c	$\hat{p}$	#	c	$\hat{p}$
1	P	0.90	11	P	0.40
2	P	0.80	12	N	0.39
3	N	0.70	13	P	0.38
4	P	0.60	14	N	0.37
5	P	0.55	15	N	0.36
6	P	0.54	16	N	0.35
7	N	0.53	17	P	0.34
8	N	0.52	18	P	0.33
9	P	0.51	19	N	0.30
10	N	0.50	20	N	0.10

NP = 10 $\quad \frac{1}{10} \quad \frac{2}{10}$

NN = 1 $\quad \frac{1}{11} \quad \frac{2}{11}$ . . .

Constructing a ROC Curve

You need a classifier that is able to rank examples by confidence (probability)



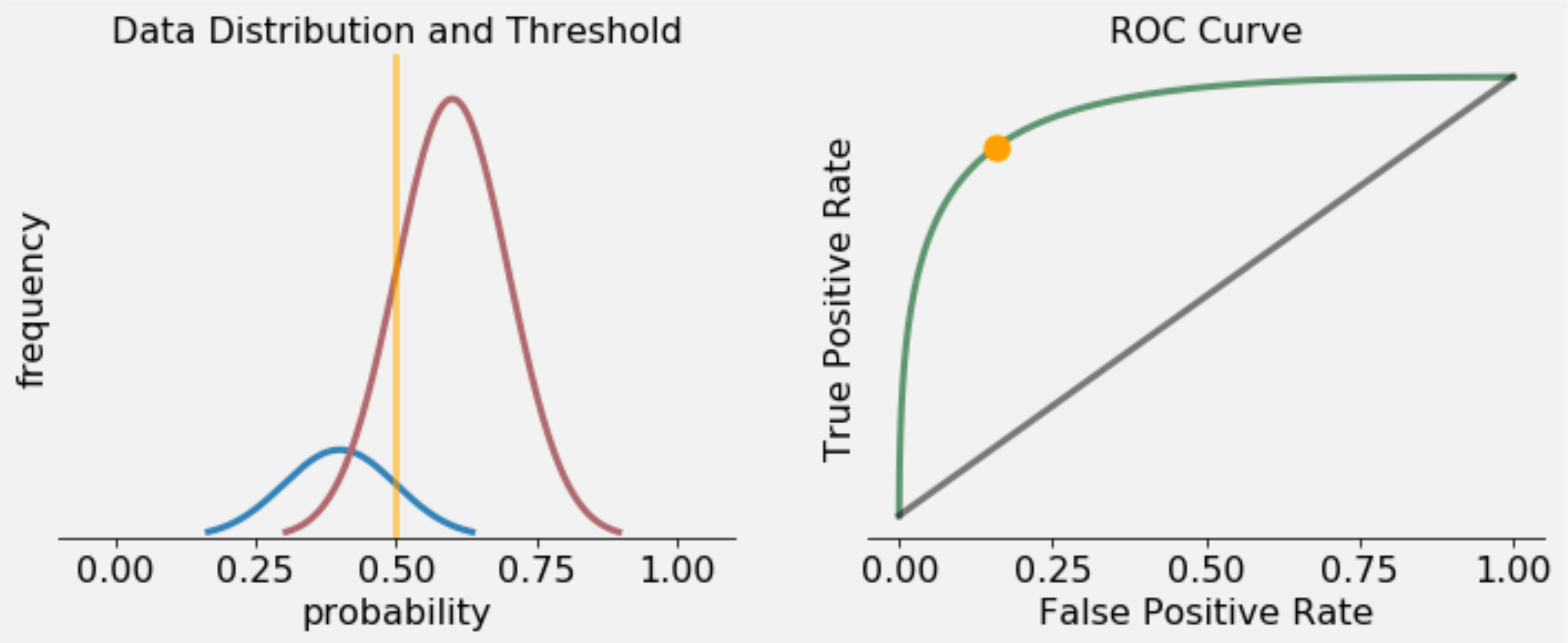
#	c	$\hat{p}$	#	c	$\hat{p}$
1	P	0.90	11	P	0.40
2	P	0.80	12	N	0.39
3	N	0.70	13	P	0.38
4	P	0.60	14	N	0.37
5	P	0.55	15	N	0.36
6	P	0.54	16	N	0.35
7	N	0.53	17	P	0.34
8	N	0.52	18	P	0.33
9	P	0.51	19	N	0.30
10	N	0.50	20	N	0.10

ROC Curve Wrap-Up

- Misclassification error / accuracy is unsatisfactory if you have imbalanced classes or care more about false positives or false negatives

- The ROC curve and AUC don't suffer from these limitations

- The ROC curve and AUC require a binary classifier that can rank examples by prediction



Acknowledgements

Much of the information on ROC curves was adopted from Kevin Markhom

If-Time: Precision vs Recall

The Confusion Matrix also gives rise to other common accuracy measures:

	Predicted Positive	Predicted Negative
Actually Positive	true positive (TP)	false negative (FN)
Actually Negative	false positive (FP)	true negative (TN)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

If-Time: Precision vs Recall

Precision: When the model predicts positive, what fraction of time is it correct?

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

Recall: When the true label is positive, what fraction of the time does model get it right?

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

Note: Recall is the same as TPR and is sometimes referred to as the **Sensitivity**