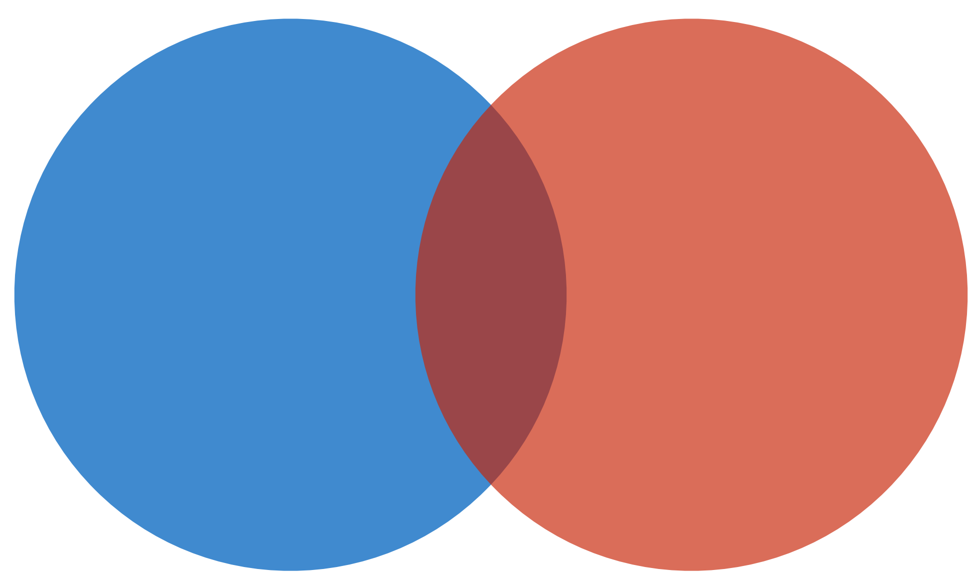
Classification

Goal: Given a *feature vector*, return an integer indexed by the set of possible *classes.*

In most cases, we care about *binary* classification in which there are only two classes (signal versus background)

There are some cases where we care about *multi-class classification*

Feature vector can be many-dimensional



Harder = more overlap between for **S** and **B**

Classification

Goal: Given a *feature vector*, return an integer indexed by the set of possible *classes.*

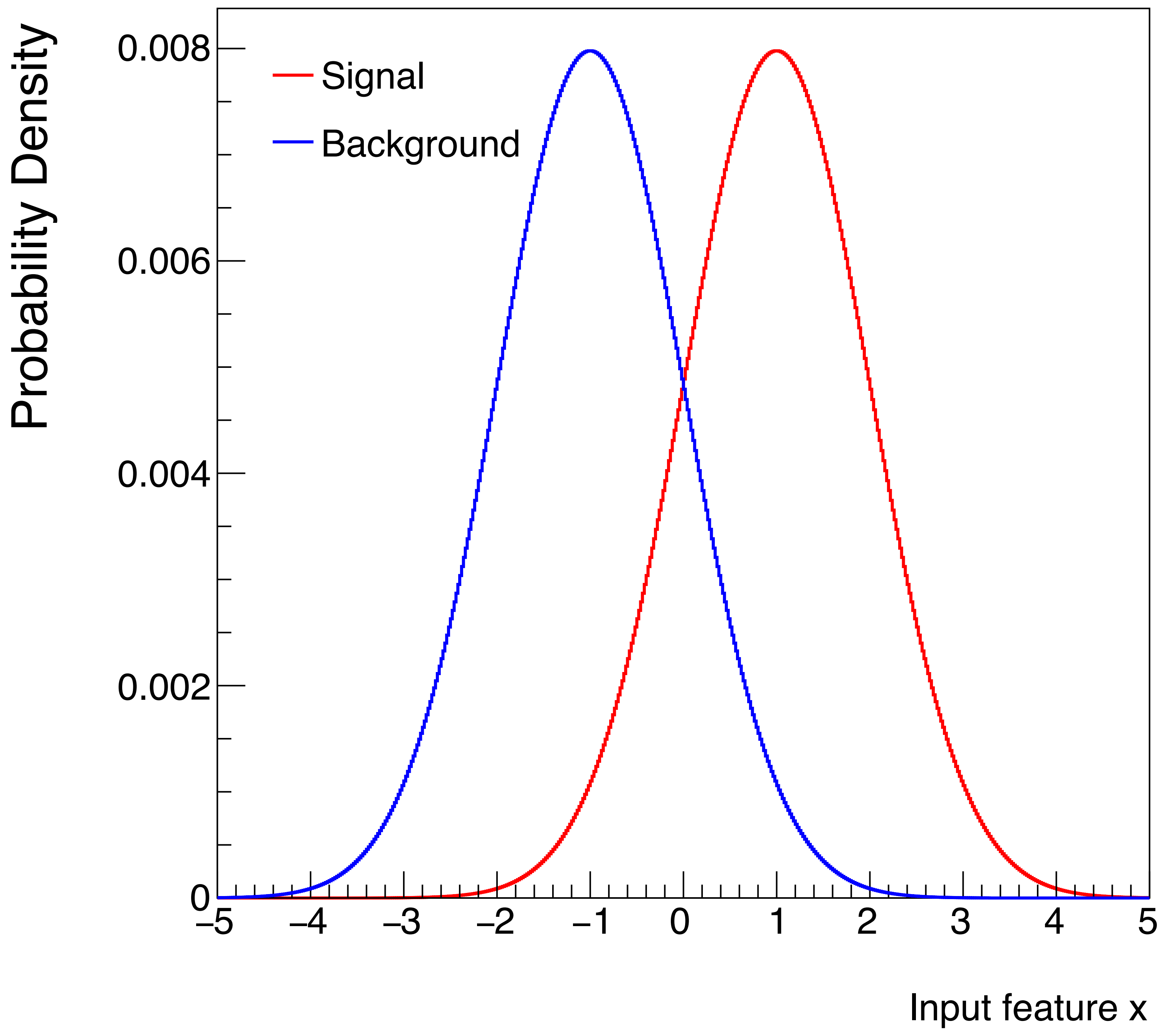
In practice, we don't just want one classifier, but an entire set of classifiers indexed by:

True Positive Rate = signal efficiency = Pr(label signal | signal) = sensitivity

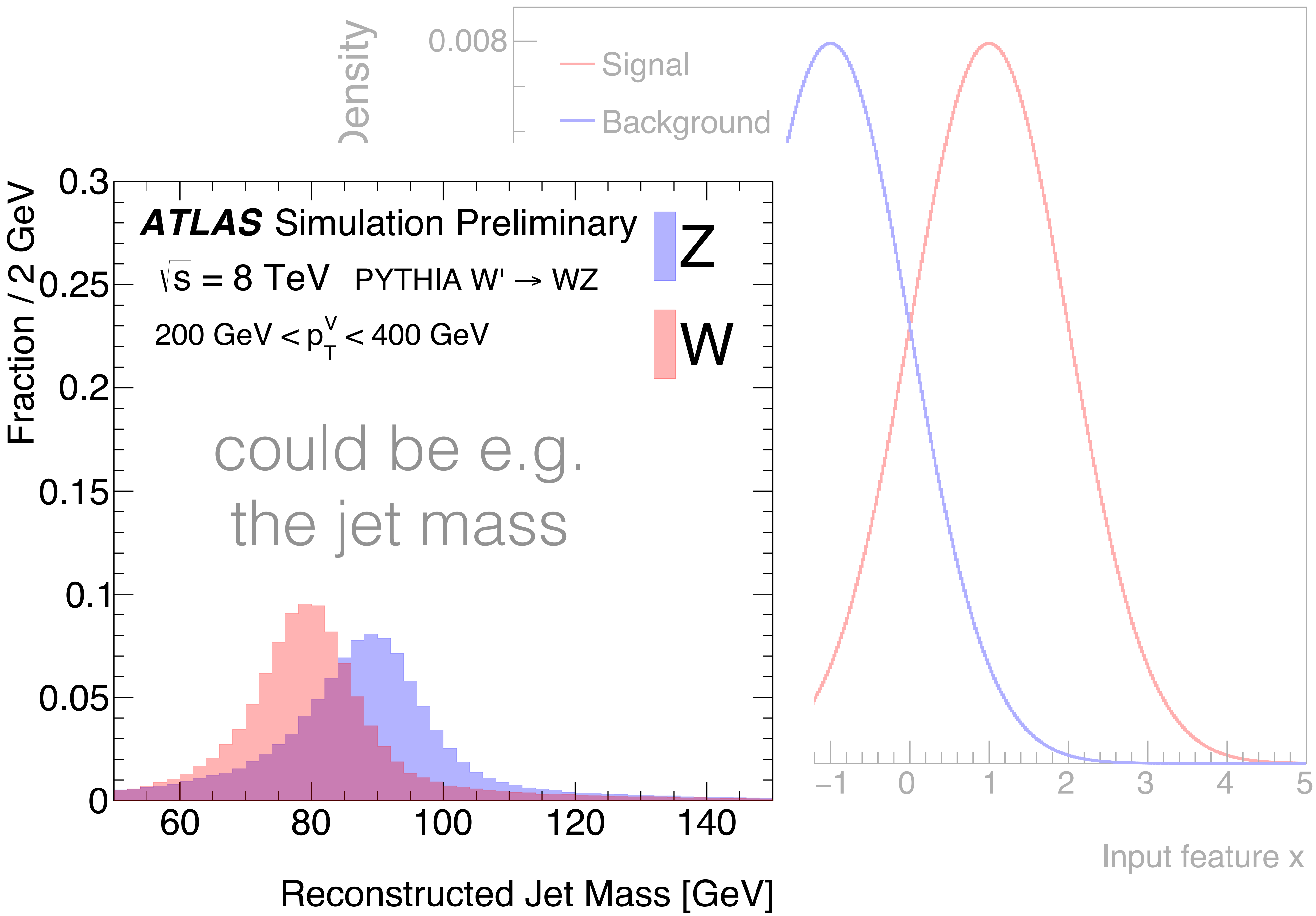
True Negative Rate = 1 - background efficiency = rejection = Pr(label background | background) = specificity

For a given TPR, we want the lowest possible TNR!

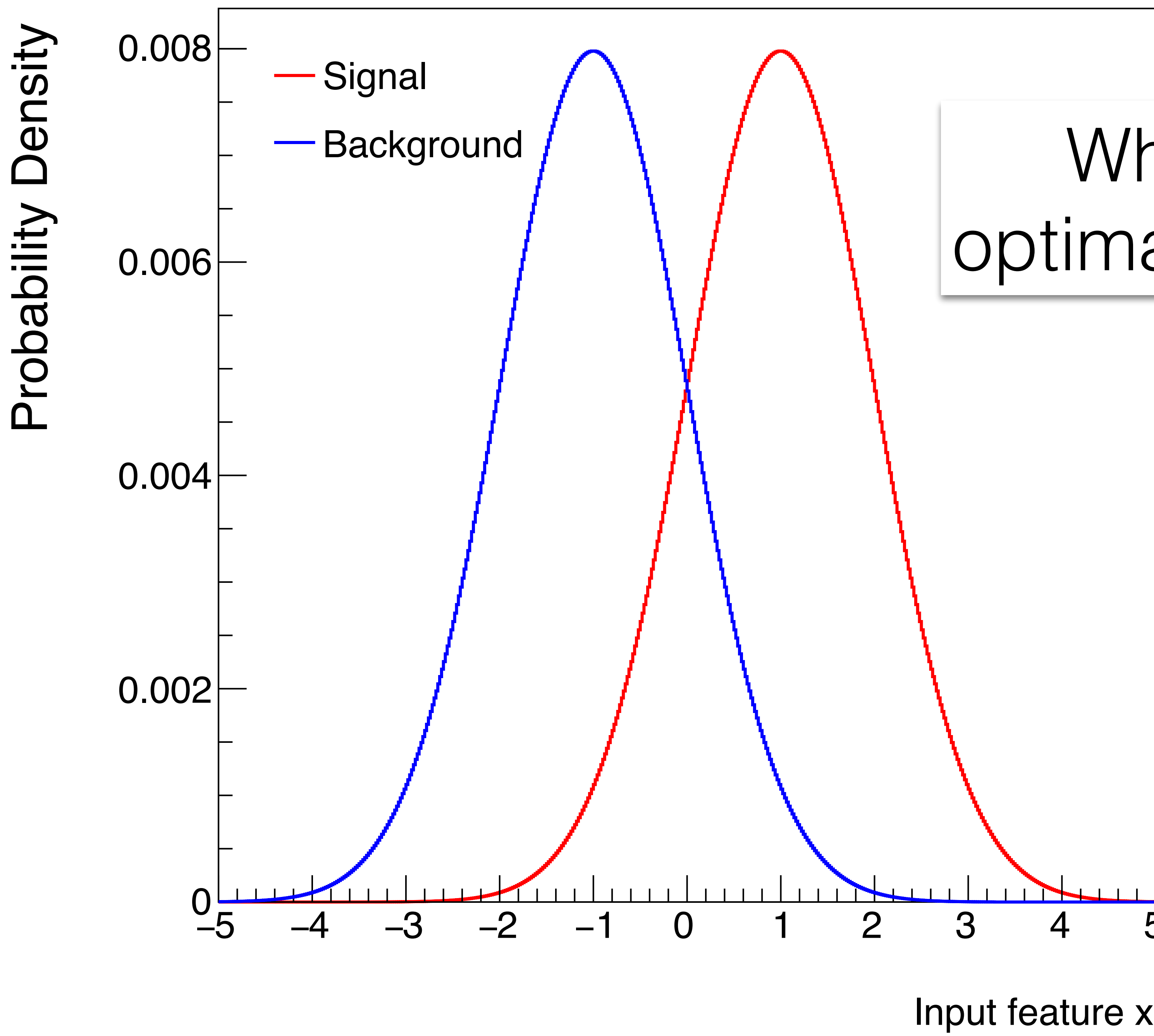
Let’s consider an important special case: binary classification in 1D



Let’s consider an important special case: binary classification in 1D

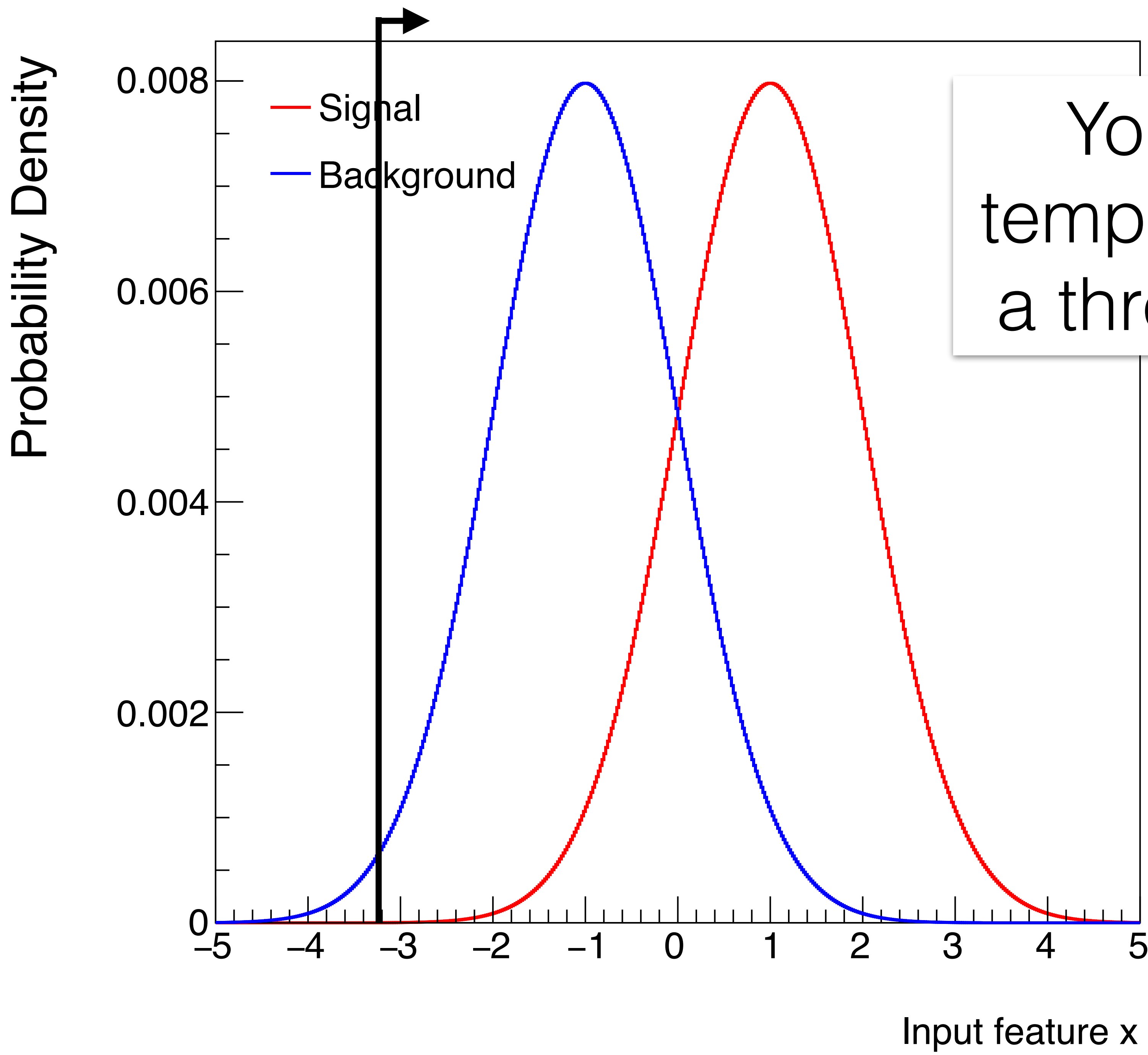


Let's consider an important special case: binary classification in 1D



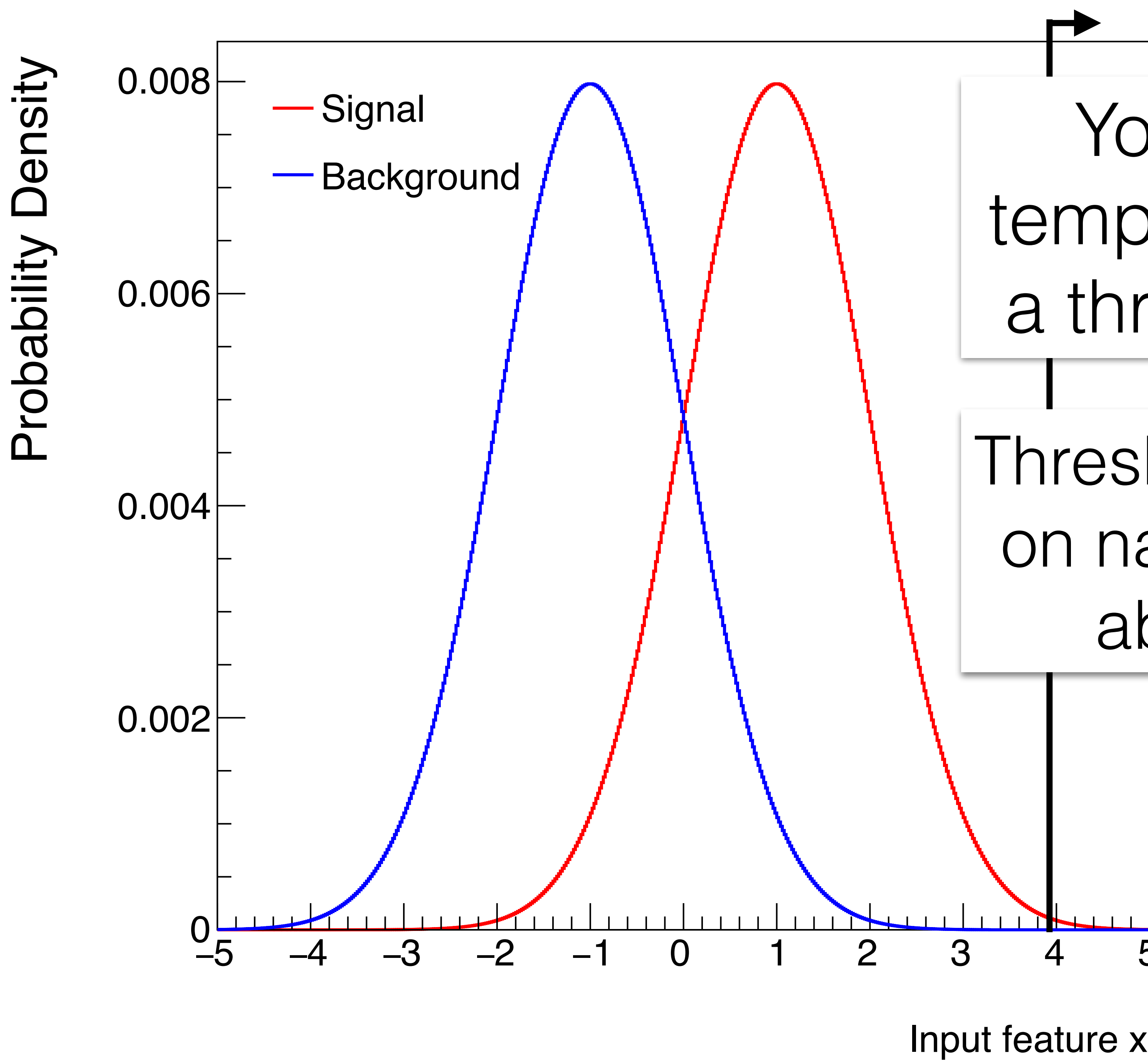
What is the optimal classifier?

Let's consider an important special case: binary classification in 1D



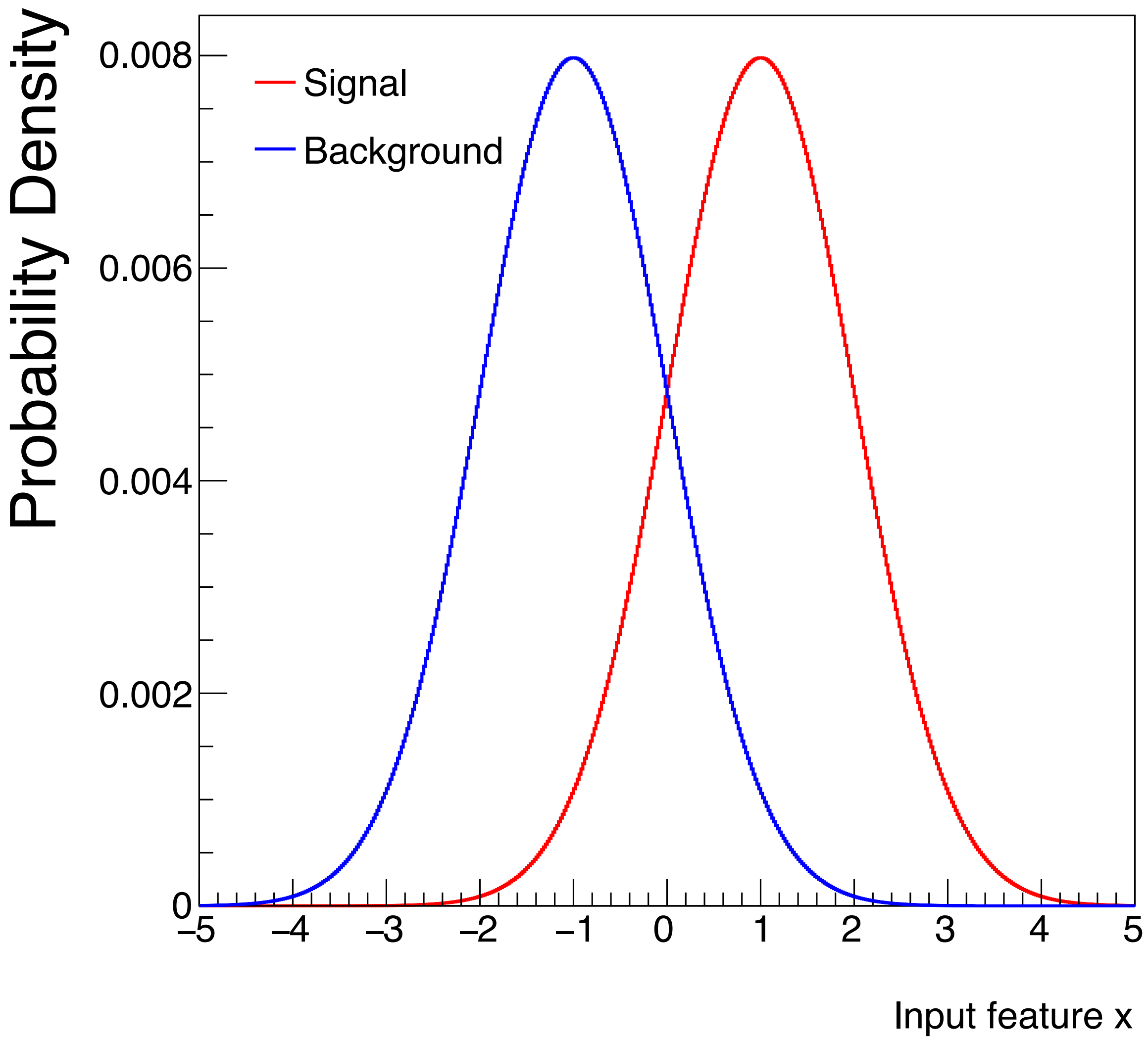
You may be tempted to place a threshold on x

Let’s consider an important special case: binary classification in 1D

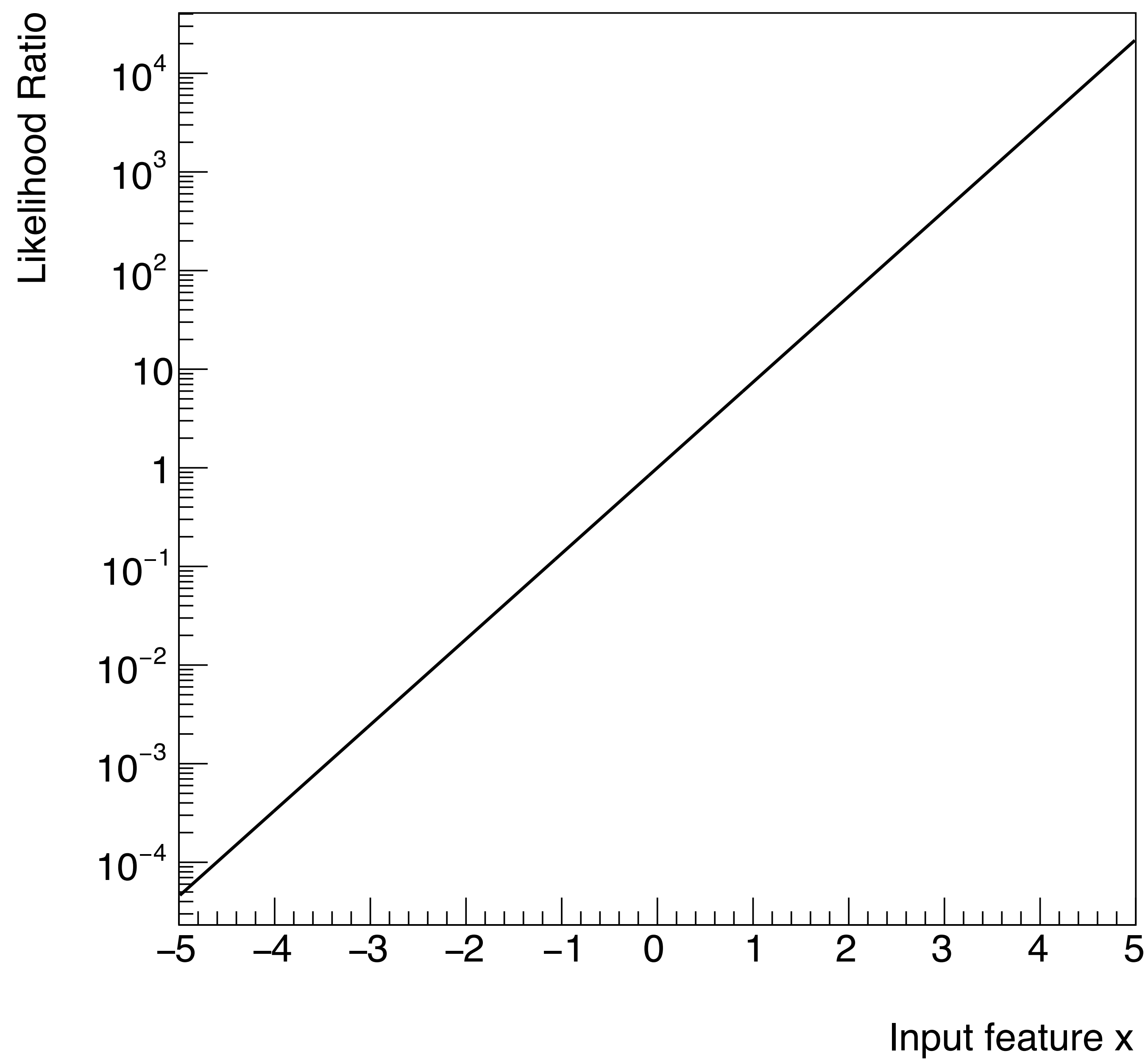


You may be tempted to place a threshold on x

Threshold depends on natural relative abundance

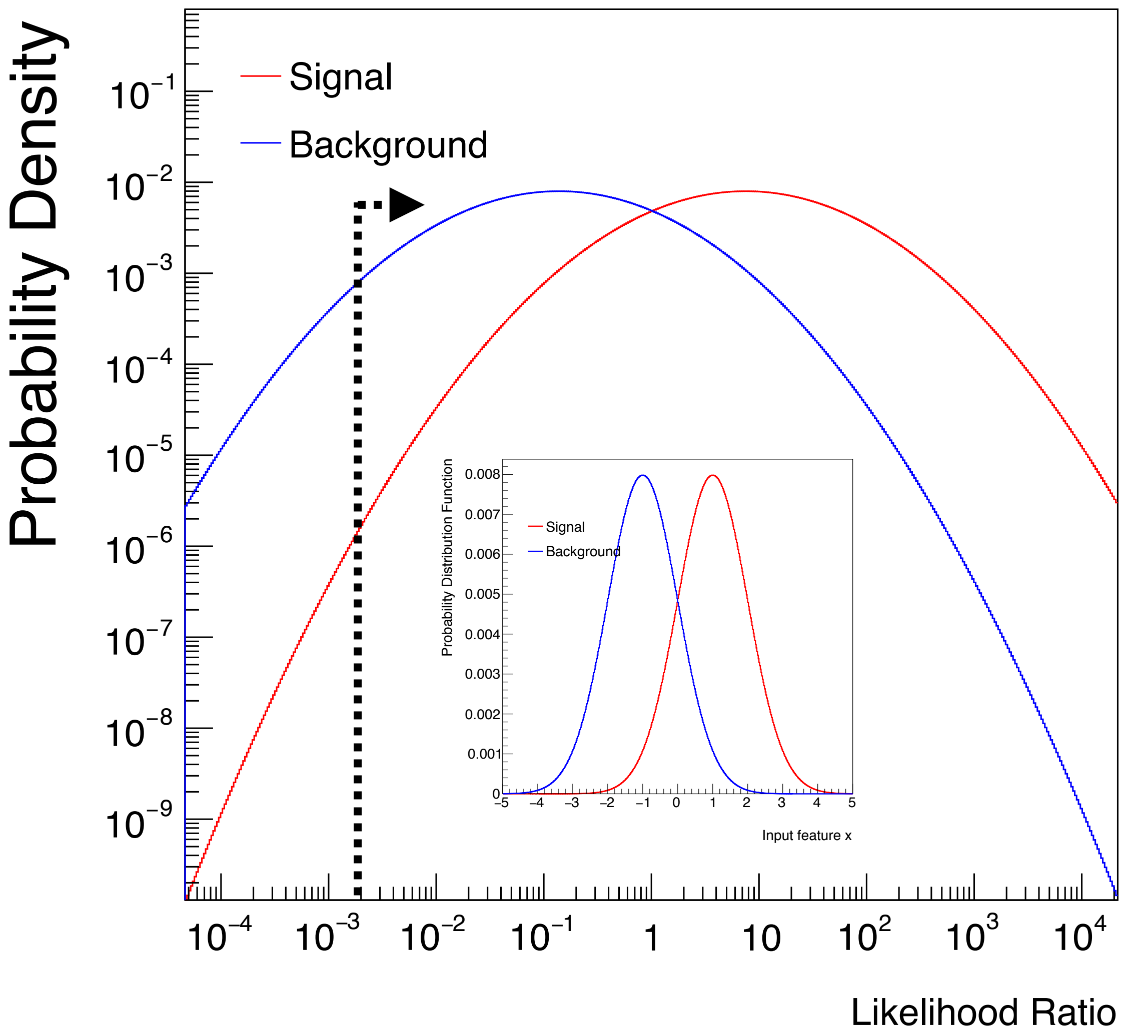


Is the simple threshold cut optimal?

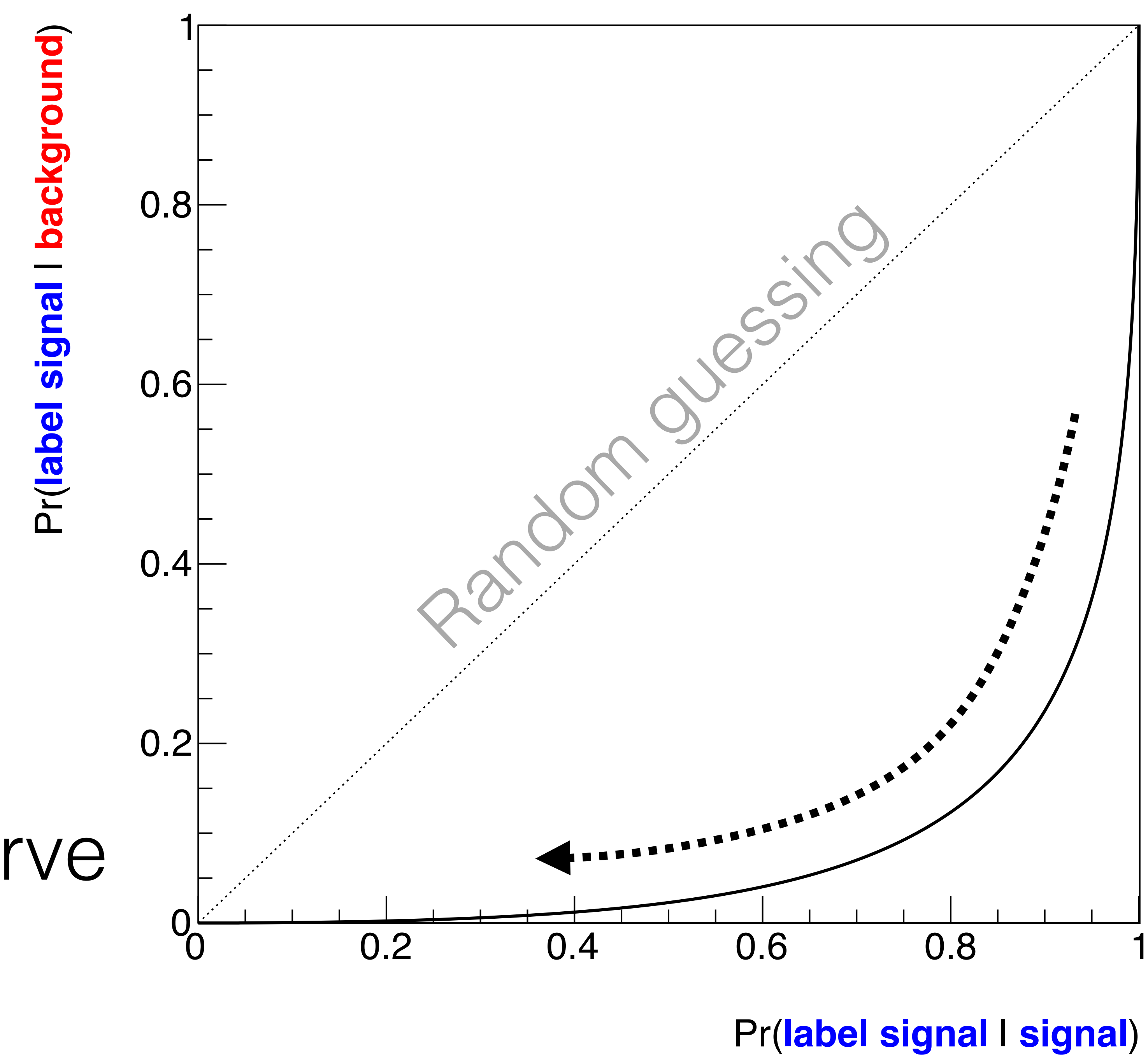


In this simple case, the log LL is proportional to x: **no need for non-linearities**!

Threshold cut is optimal



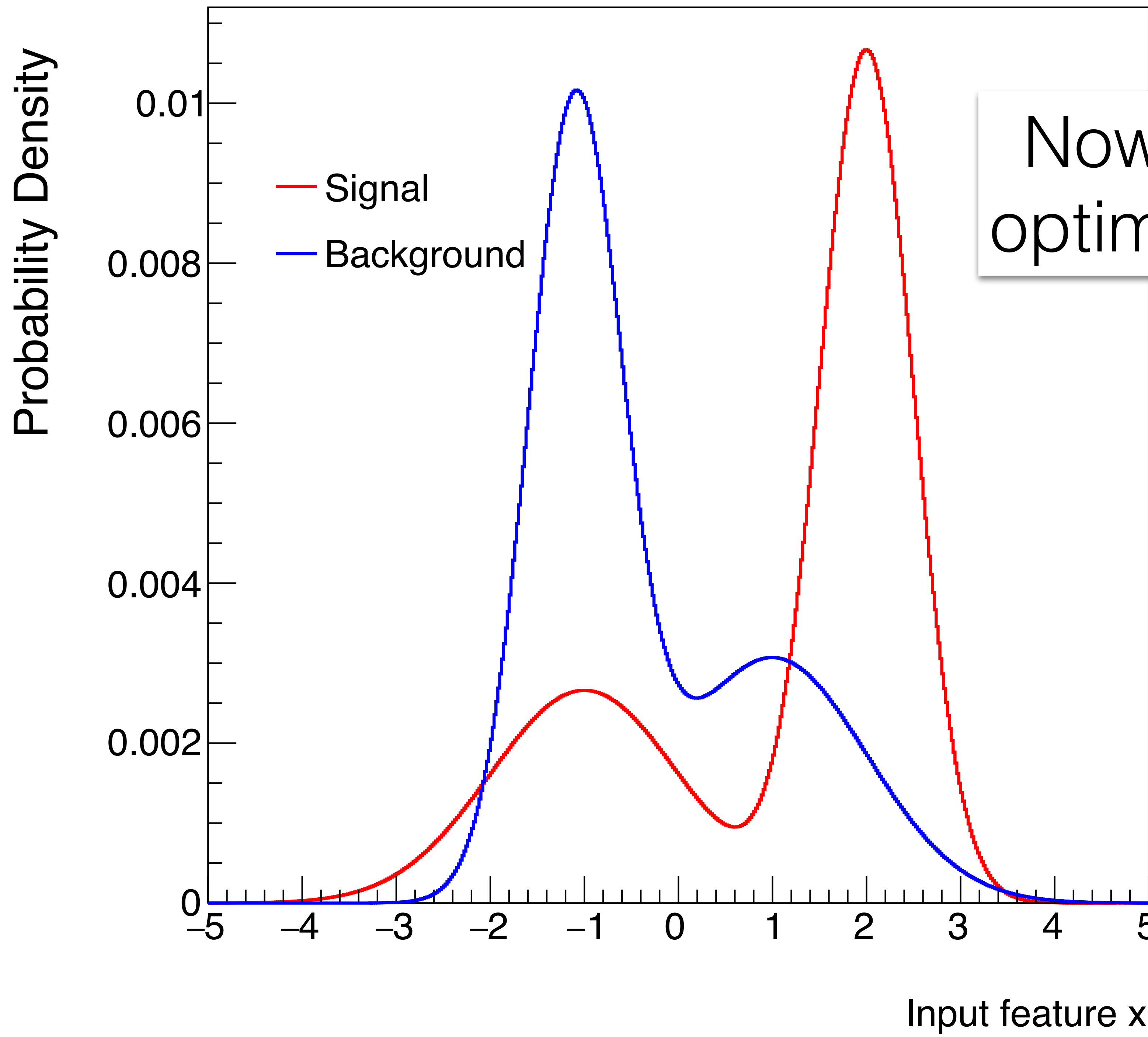
The optimal procedure is a threshold on the LL



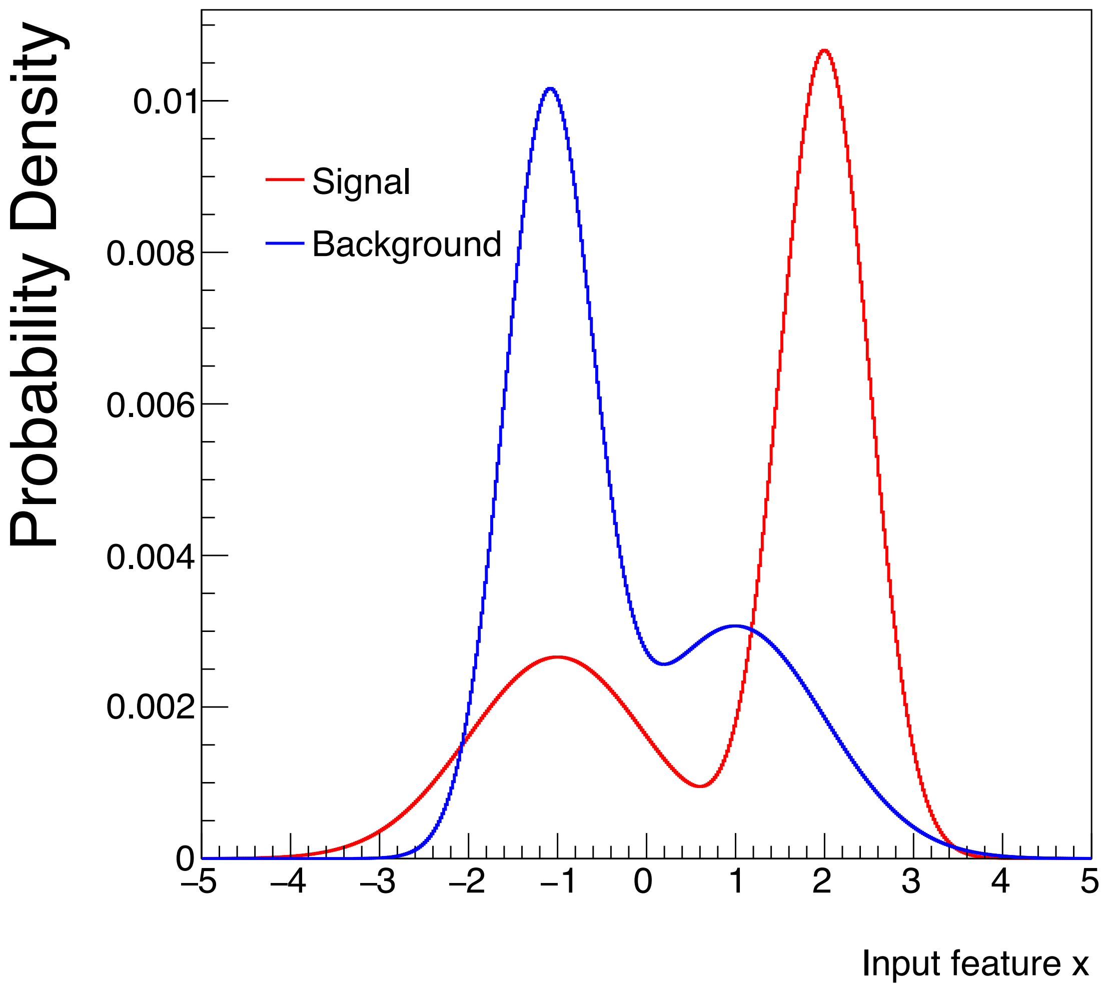
“Receiver Operating Characteristic” (**ROC**) Curve

What if the distribution of x is complicated?

Real life is complicated!

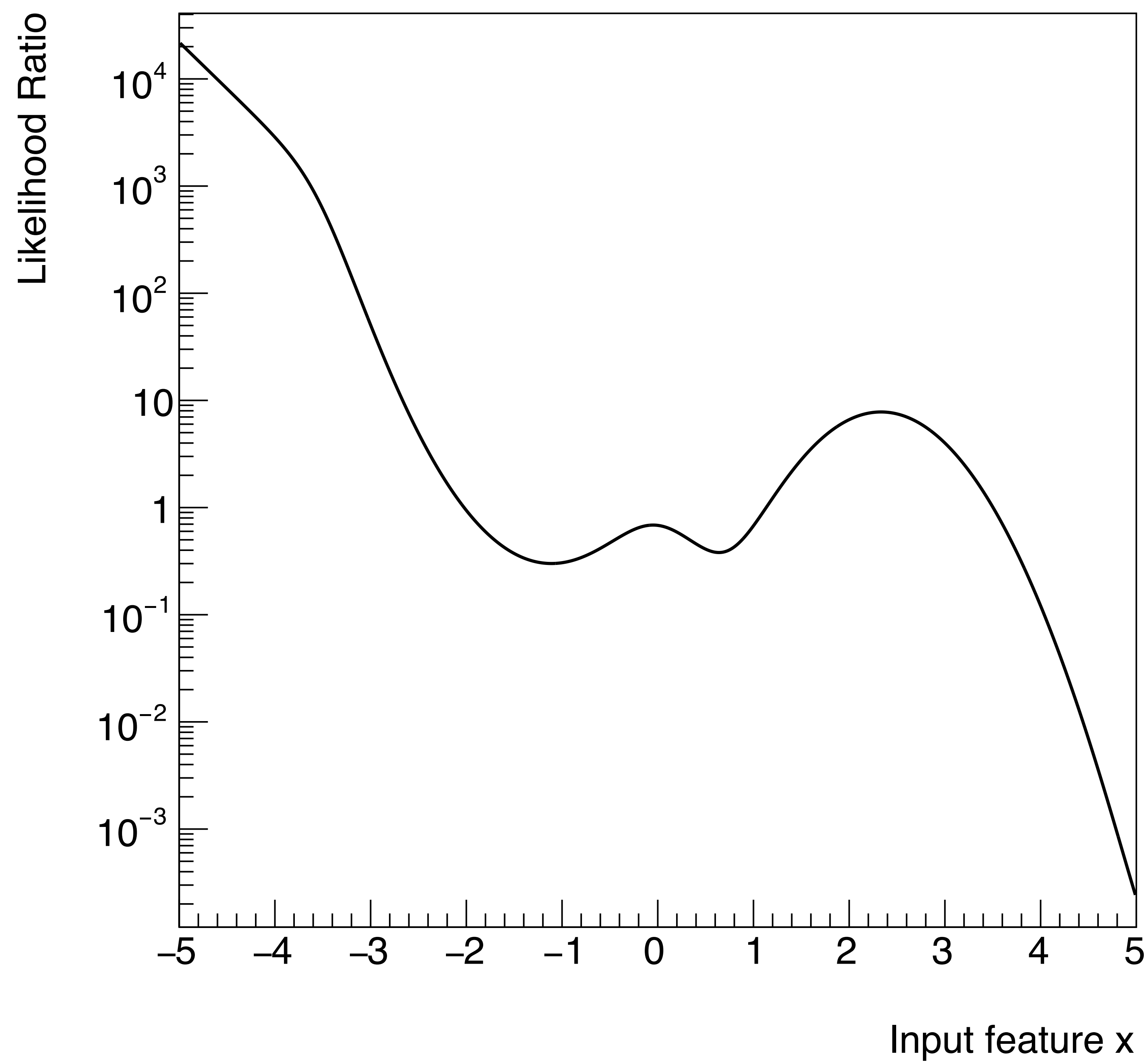


Now what is the optimal classifier?

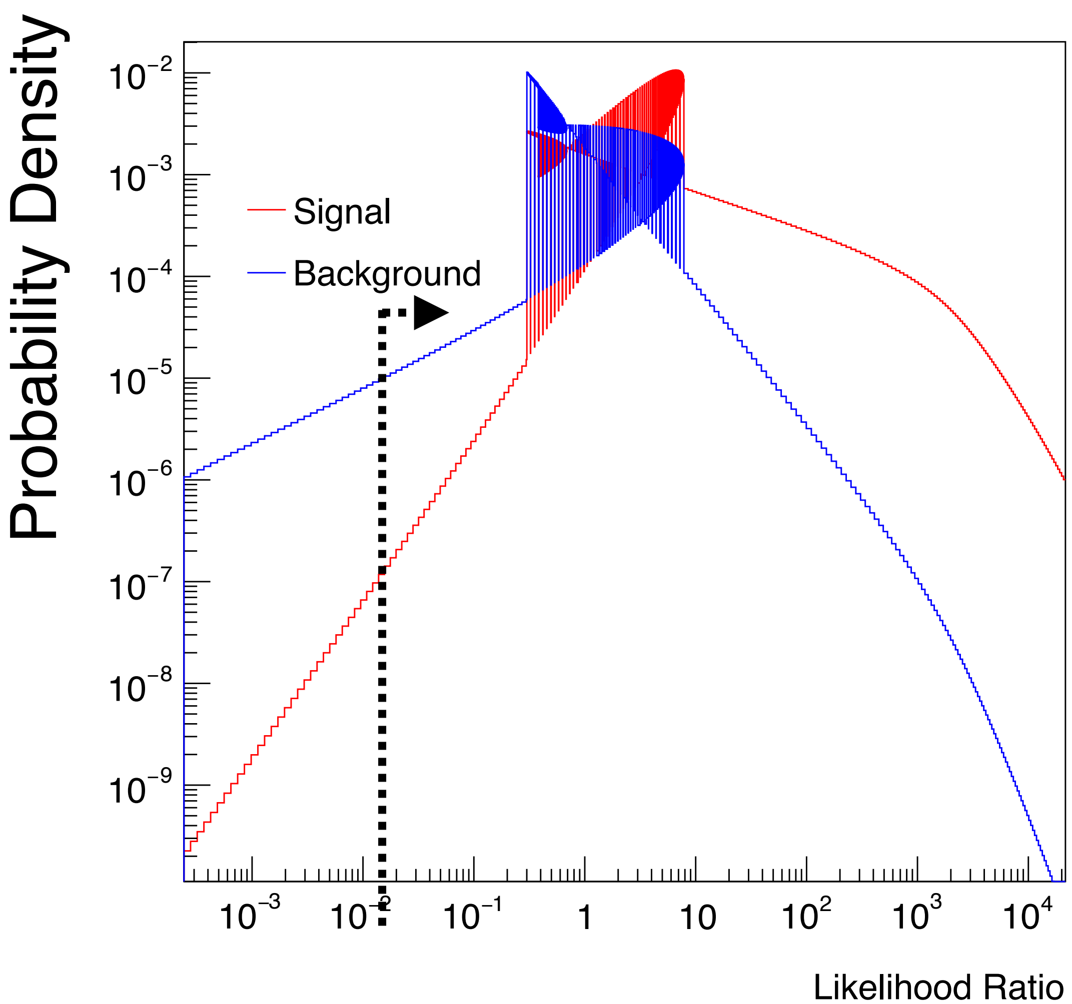


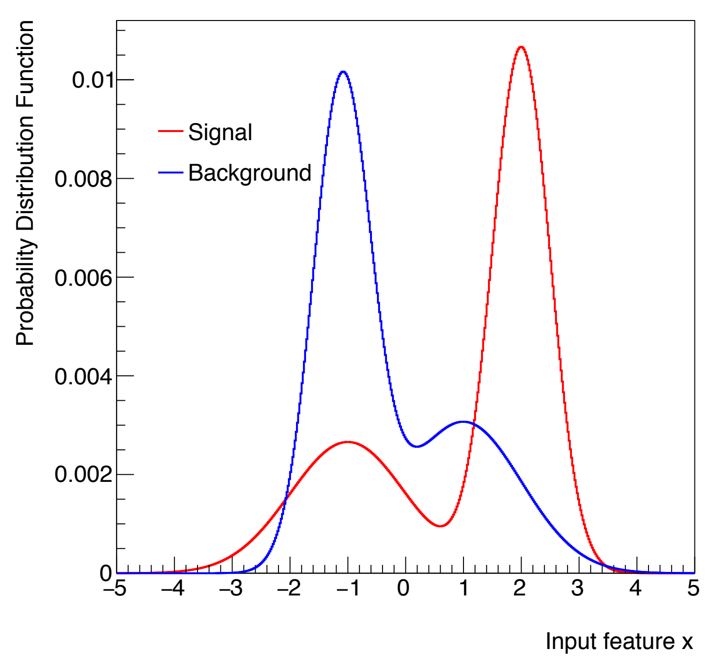
In this case, LL is highly non-linear (**non-monotonic**) function of x

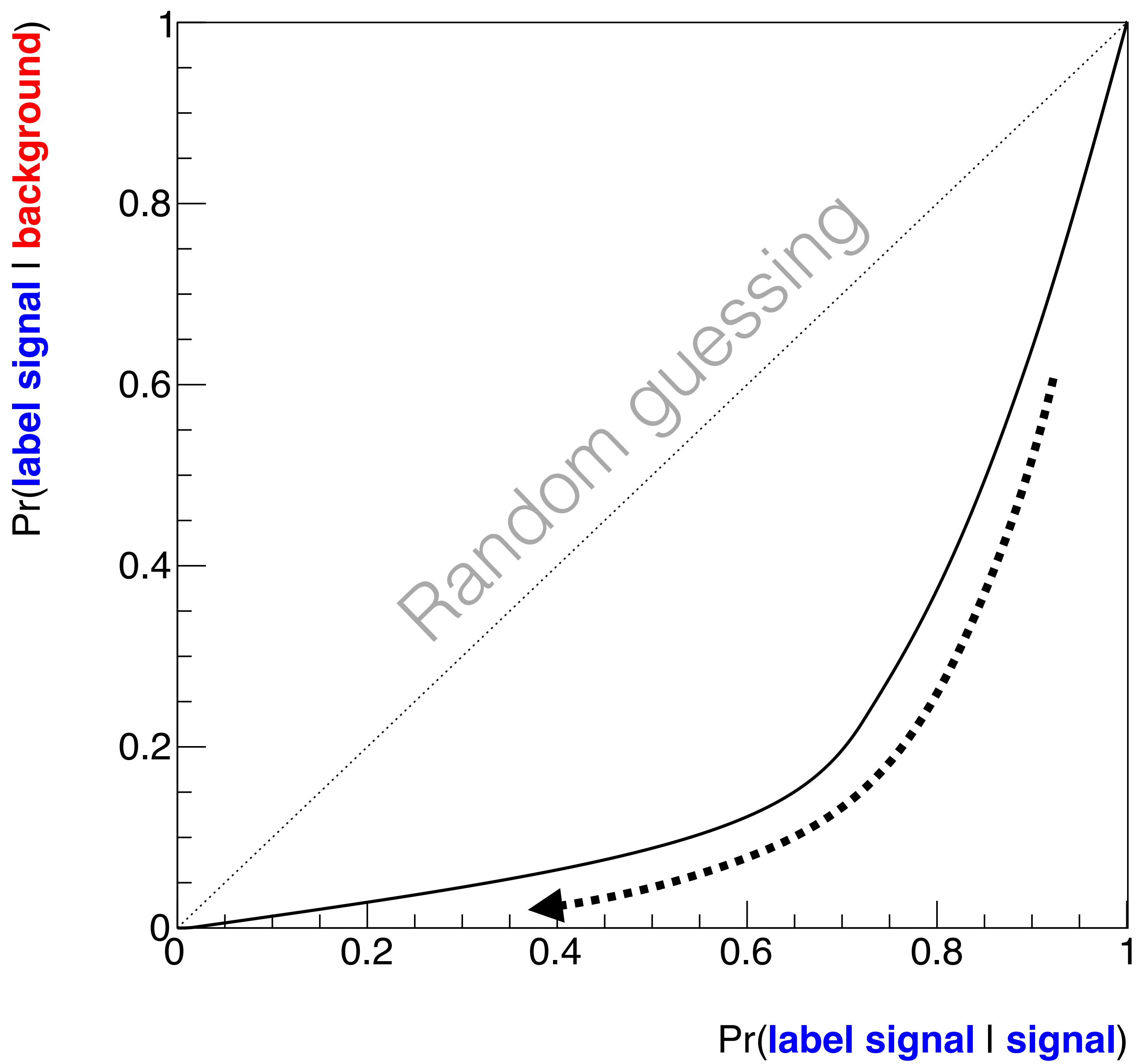
A threshold on x would be sub-optimal



ROC is worse than the Gaussians, but that is expected since the overlap in their PDFs is higher.







Why don’t we always just compute the optimal classifier?

In the last slides, we had to estimate the likelihood ratio - this required binning the PDF

binning works very well in 1D, but becomes quickly intractable as the feature vector dimension >> 1 (“curse of dimensionality”)

machine learning for classification is simply **the art of estimating the likelihood ratio with limited training examples**