Introduction to Data Science

BRIAN D'ALESSANDRO

ADJUNCT PROFESSOR, NYU

FALL 2016

EVALUATION METRICS

REMINDER

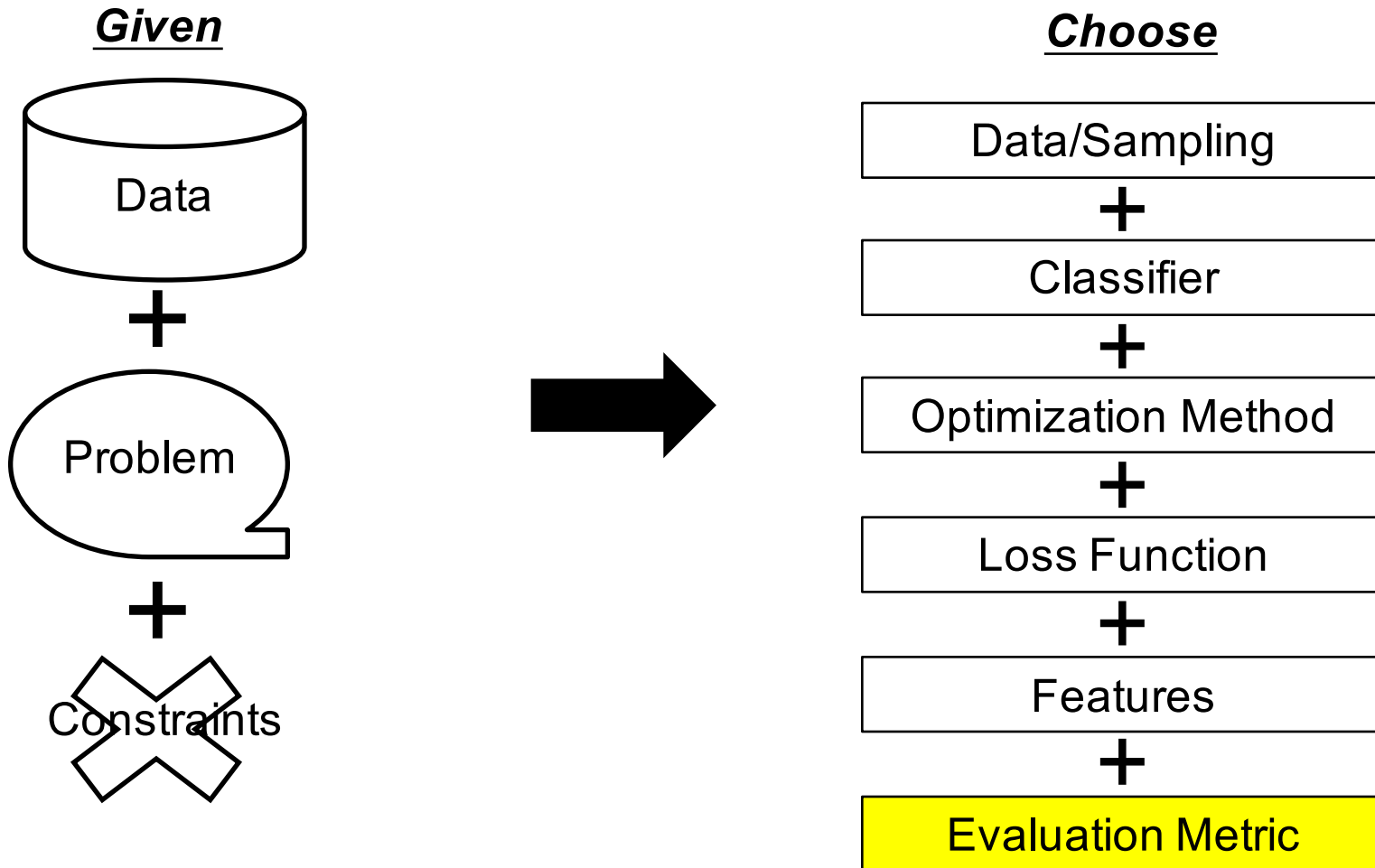You will never build the *perfect* model… but we can always have a *best* model.

So far we have discussed the following design options:

[Algorithm, Feature Set , Hyper-parameters (complexity)]

We also need to choose an evaluation metric!

A COMMON THEME

Few problems have out of the box solutions

Given

Data
+
Problem
+
~~Constraints~~

➡️

Choose

Data/Sampling
+
| Classifier |
+
| Optimization Method |
+
| Loss Function |
+
| Features |
+
| Evaluation Metric |

The Data Scientist has to navigate these choices

THE RIGHT METRIC DEPENDS ON YOUR GOALS

- **Ranking** - Who are the top k prospects for my campaign, or what 10 items should I recommend?

- **Classification** – Is this email spam or not? Is this number a '1' or a '7'?

- **Density Estimation** – What is the probability that this transaction is fraud? What is the expected spend of a new credit card customer?

METRICS FOR THESE GOALS

Ranking   Area under the Receiver Operator Curve (AUC)
                       Area under the Cumulative Lift Curve (ACLC)

Classification   Lift (LFT)
                       Accuracy (ACC)
                       F-Score (FSC)
                       Precision (PRE)
                       Recall (RCL)

Density Estimation Mean Absolute Error (MAE)
                        Mean Squared Error (MSE)
                        Cross-Entropy /Log-Likelihood (LL)

TRAIN VS. VAL/TEST LOSS

We don't need to use the same error/loss/risk function on our training data as we do our validation or test data.

Training Loss

vs.

Testing Loss

$$R_{train} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{L}(f(x_i^{train}), y_i^{train})$$

$$R_{test} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{L}(f(x_i^{test}), y_i^{test})$$

Usually used because they are well posed and "easy" to actually optimize (i.e., quadratic and smooth).

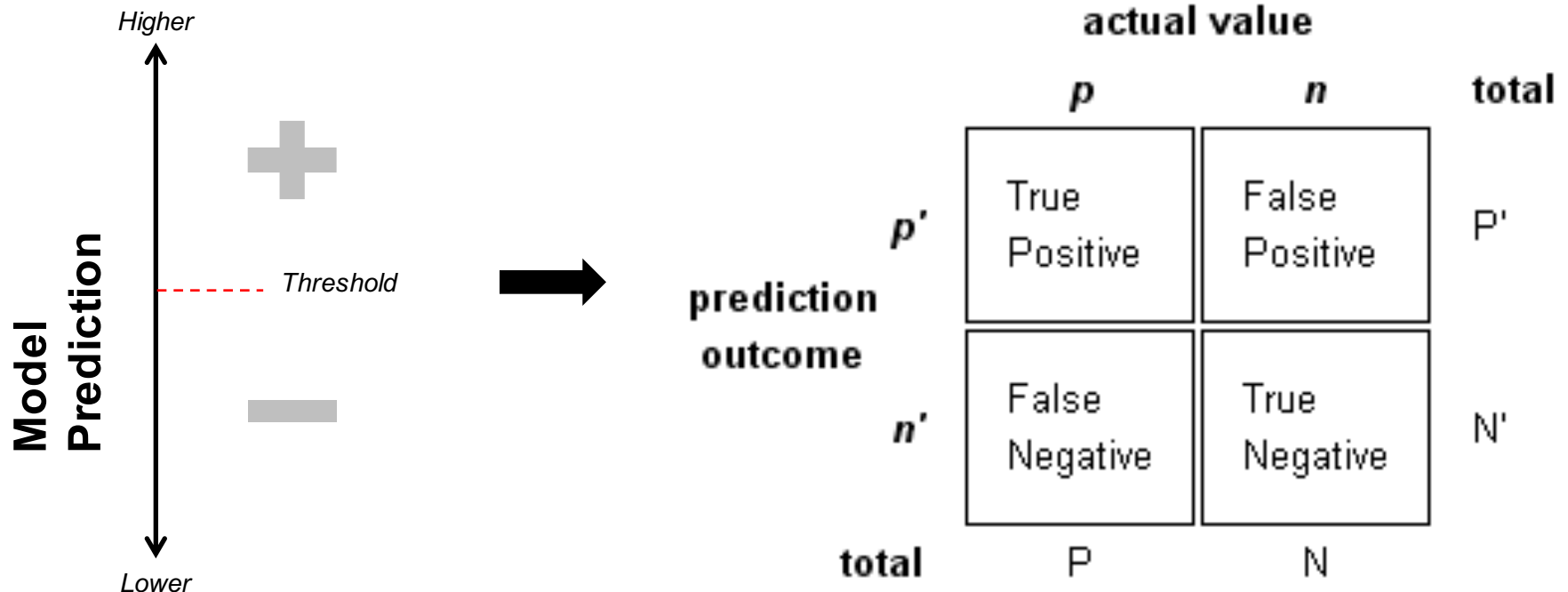Logistic Loss, Hinge Loss, Least Squares etc.

Often the most suitable loss function is not quadratic or too complex to train efficiently.

Precision, AUC, Recall.

CONFUSION MATRIX

Many of the metrics we use derive from the confusion matrix. For binary classification we assume there exists some real valued function f(x) and a decision threshold δ.

$$\hat{Y} = I(f(x) > \delta)$$

		actual value		
		p	**n**	total
prediction outcome	**p'**	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
	total	P	N	

Classification Metrics

We can derive many classification metrics from the confusion matrix.

actual value

**Terminology and derivations
from a confusion matrix**

true positive (TP)
 eqv. with hit

true negative (TN)
 eqv. with correct rejection

false positive (FP)
 eqv. with false alarm, Type I error

false negative (FN)
 eqv. with miss, Type II error

sensitivity or true positive rate (TPR)
 eqv. with hit rate, recall
$$TPR = TP/P = TP/(TP + FN)$$

false positive rate (FPR)
 eqv. with fall-out
$$FPR = FP/N = FP/(FP + TN)$$

accuracy (ACC)
$$ACC = (TP + TN)/(P + N)$$

specificity (SPC) or True Negative Rate
$$SPC = TN/N = TN/(FP + TN) = 1 - FPR$$

positive predictive value (PPV)
 eqv. with precision
$$PPV = TP/(TP + FP)$$

negative predictive value (NPV)
$$NPV = TN/(TN + FN)$$

false discovery rate (FDR)
$$FDR = FP/(FP + TP)$$

Matthews correlation coefficient (MCC)
$$MCC = (TP * TN - FP * FN)/\sqrt{PNP'N'}$$

F1 score
$$F1 = 2TP/(P + P') = 2TP/(2TP + FP + FN)$$

Source: Fawcett (2006).

SOME EXPOSITION

Precision - of all the instances I predict are positive, how many actually are positive? Lift is the precision normalized by the base rate (P(Y)).

Recall- how many of the total positives out there did I classify as positive?

TOWARDS A RANKING METRIC

Classification metrics depend on choosing a single threshold. But what if you don't know or need the threshold?
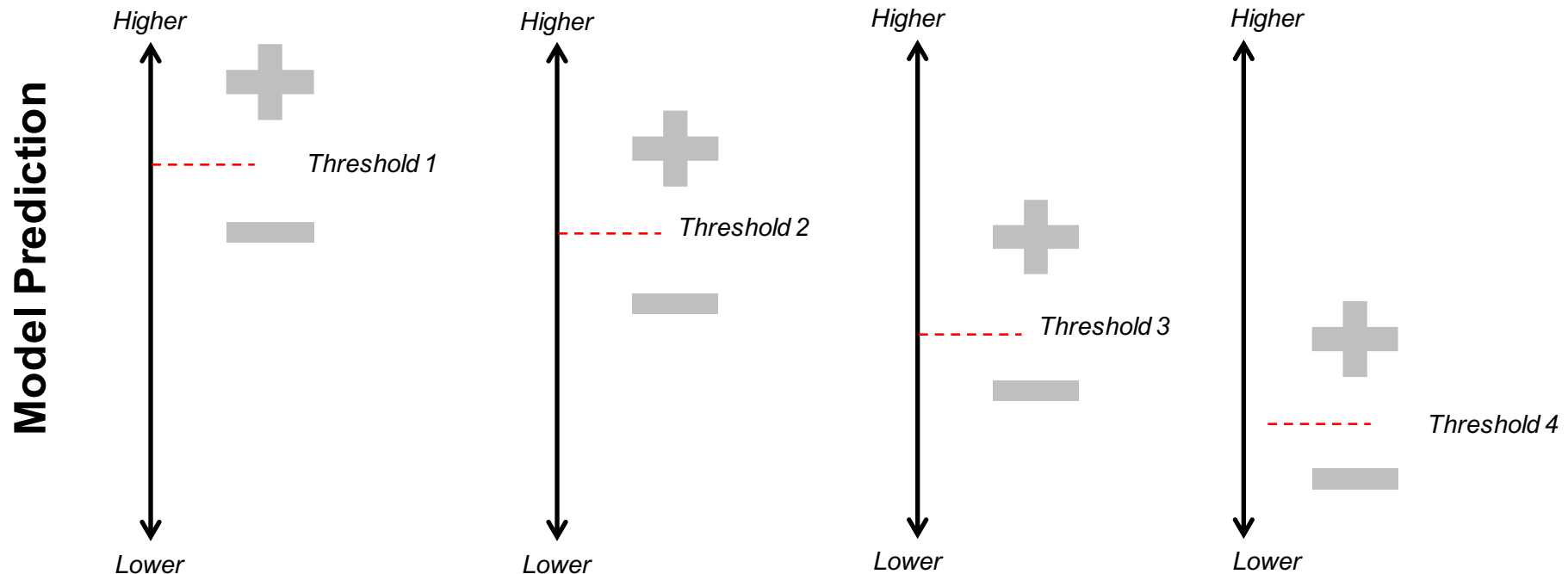
For each threshold we will get different accuracy, lift, precision and recall.
We want an evaluation method that considers the trade-off on these metrics when using different thresholds.

THE THRESHOLDING TRADE-OFF

Each threshold we choose creates a trade-off between false positive rate and true positive rate.


True Positive Rate: *TP/(TP+FN)*

Threshold 4

Threshold 3

Threshold 2

Threshold 1

False Positive Rate: *FP/(FP+TN)*
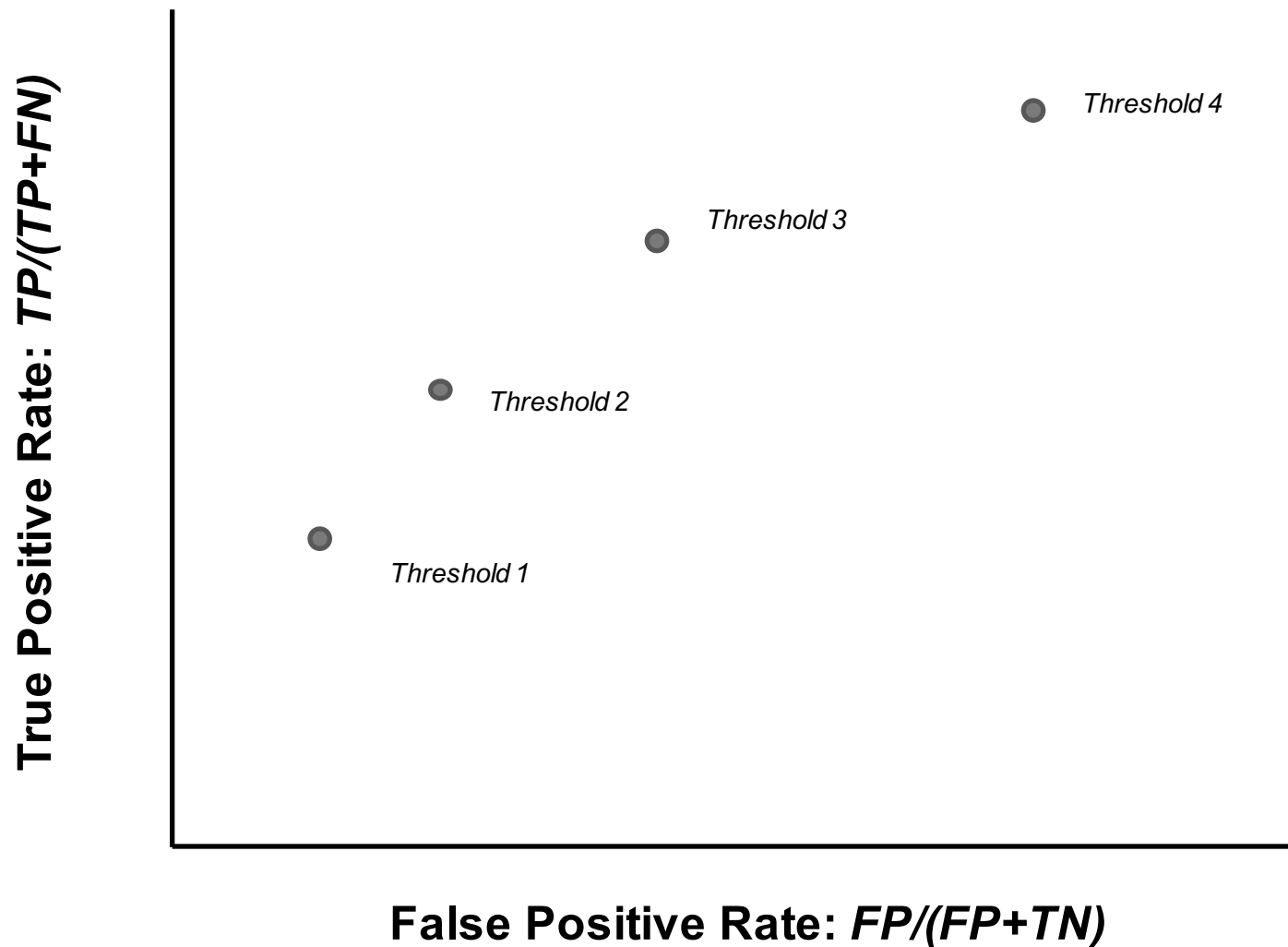
THE ROC CURVE

If we consider every threshold and plot the trade-off, we arrive at the ROC curve.

True Positive Rate: *TP/(TP+FN)* (y-axis)

False Positive Rate: *FP/(FP+TN)* (x-axis)

Threshold 1
Threshold 2
Threshold 3
Threshold 4
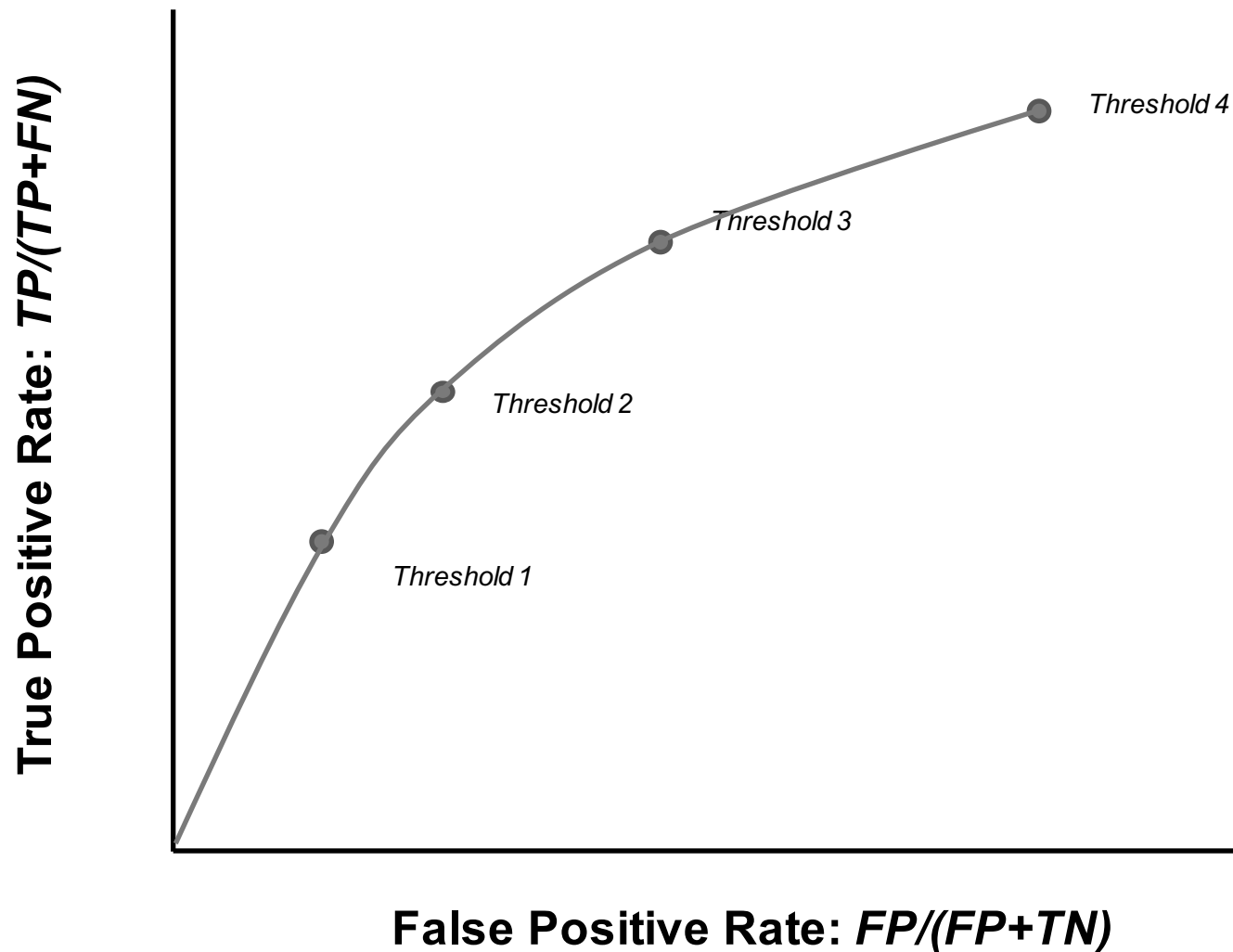
THE AREA UNDER THE ROC CURVE

The area under this curve gives a comprehensive summary of how well your classifier ranks.

True Positive Rate: $TP/(TP+FN)$

False Positive Rate: $FP/(FP+TN)$
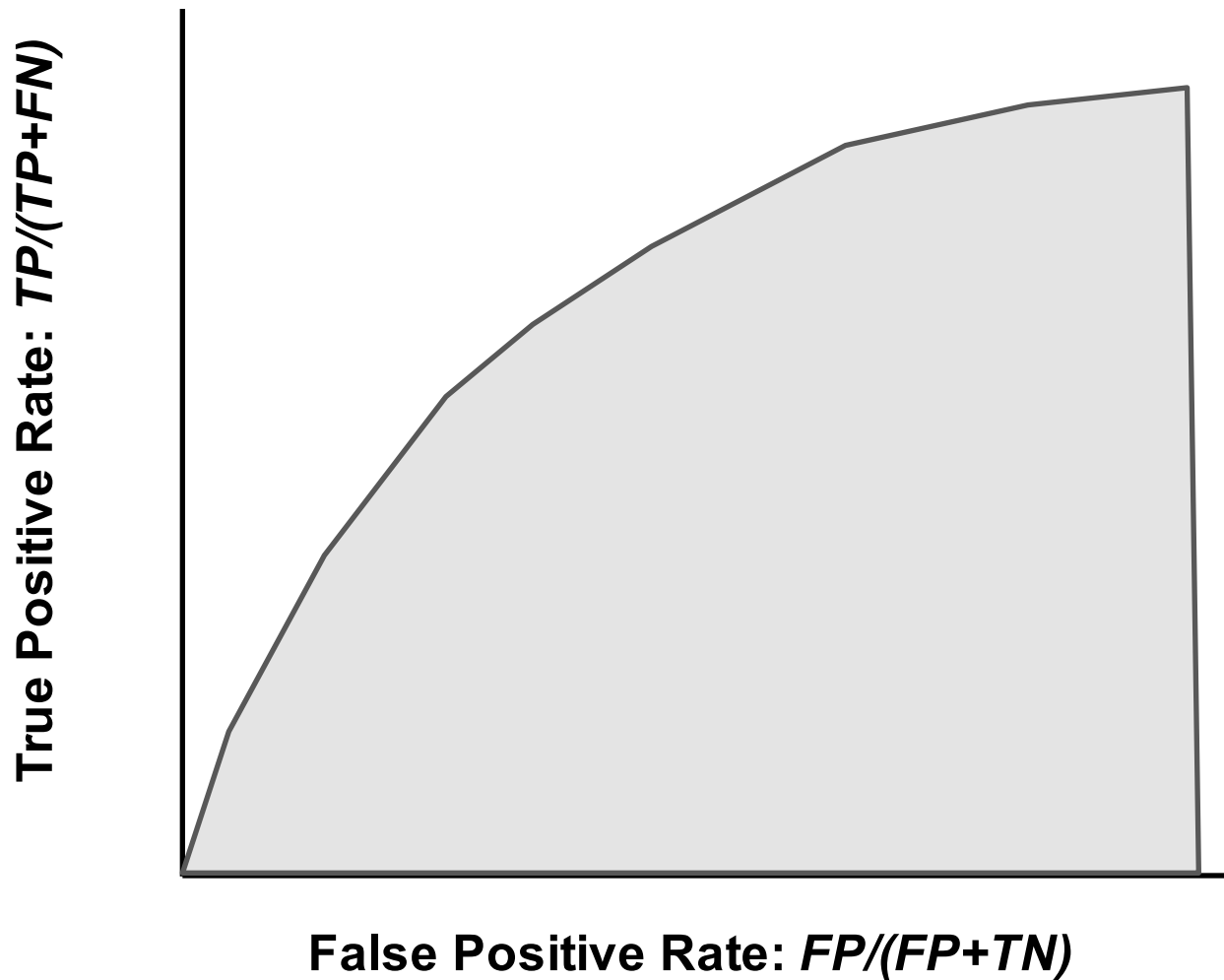
PROBABILISTIC INTERPRETATION OF AUC

Let $s^+(x)$ and $s^-(x)$ be the PDF of $f(x)$ for positive and negative labels, respectively. Let $S^+(x)$ and $S^-(x)$ be the CDF of $f(x)$ for positive and negative labels, respectively.

$$AUC = P(f(x^+) > f(x^-)) = \int_{-\infty}^{\infty} s^+(x)\, S^-(x)\, dx = \int_{-\infty}^{\infty} s^-(x)\, (1 - S^+(x))\, dx$$
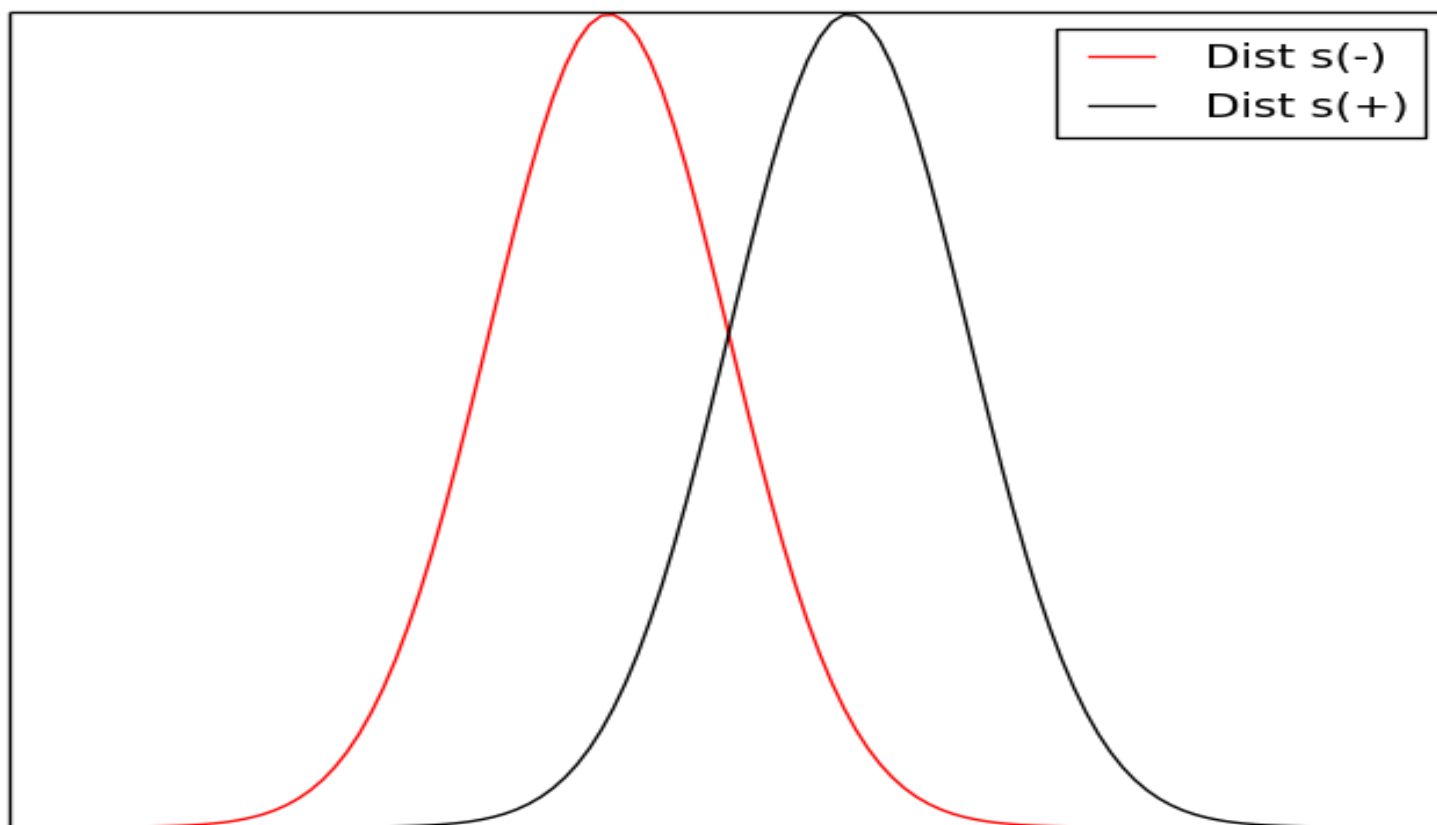
DIFFERENT EXAMPLES

A good AUC depends on the problem. Although AUC=0.6 means reasonably bad separation of the classes, it could still create value. Also, really high AUCs are often too good to be true and should be treated with suspicion.
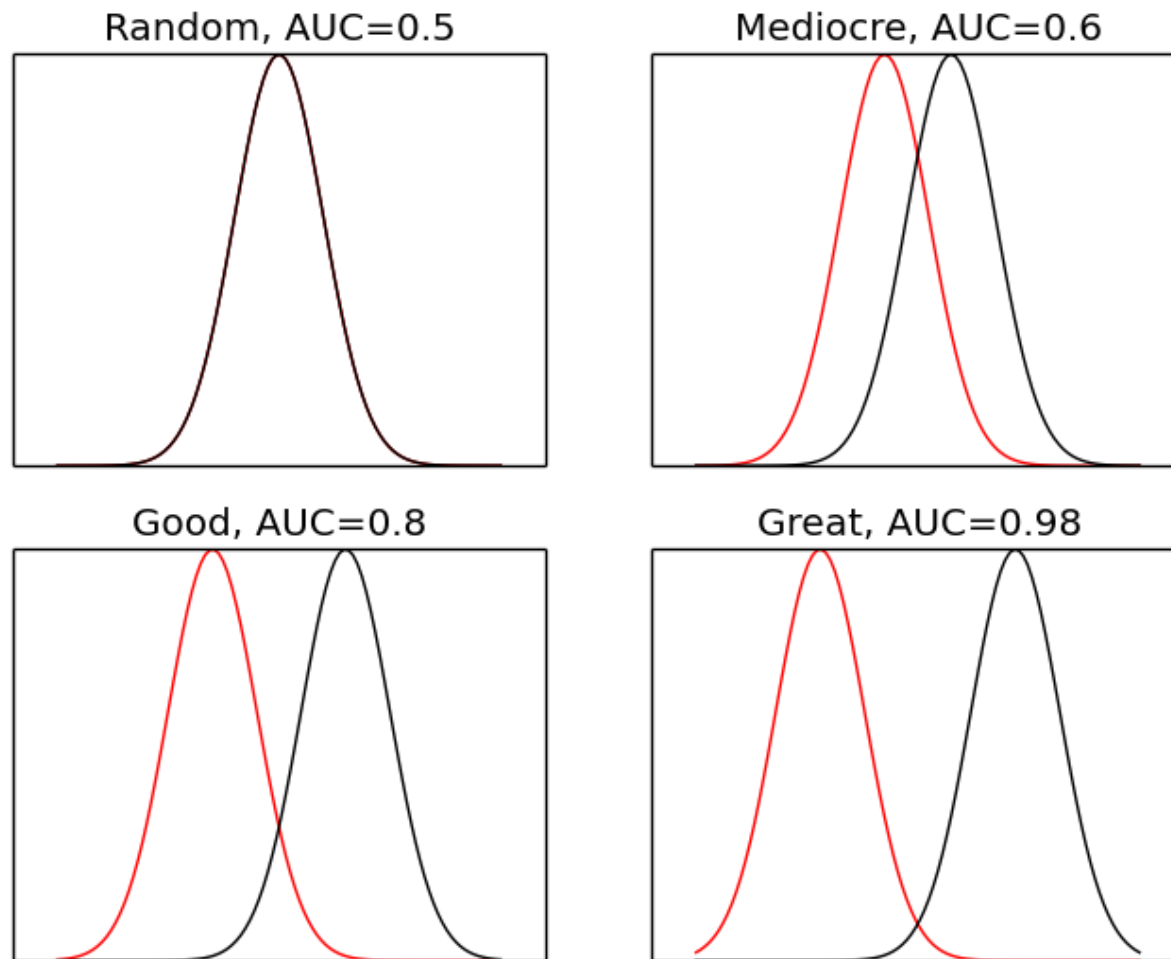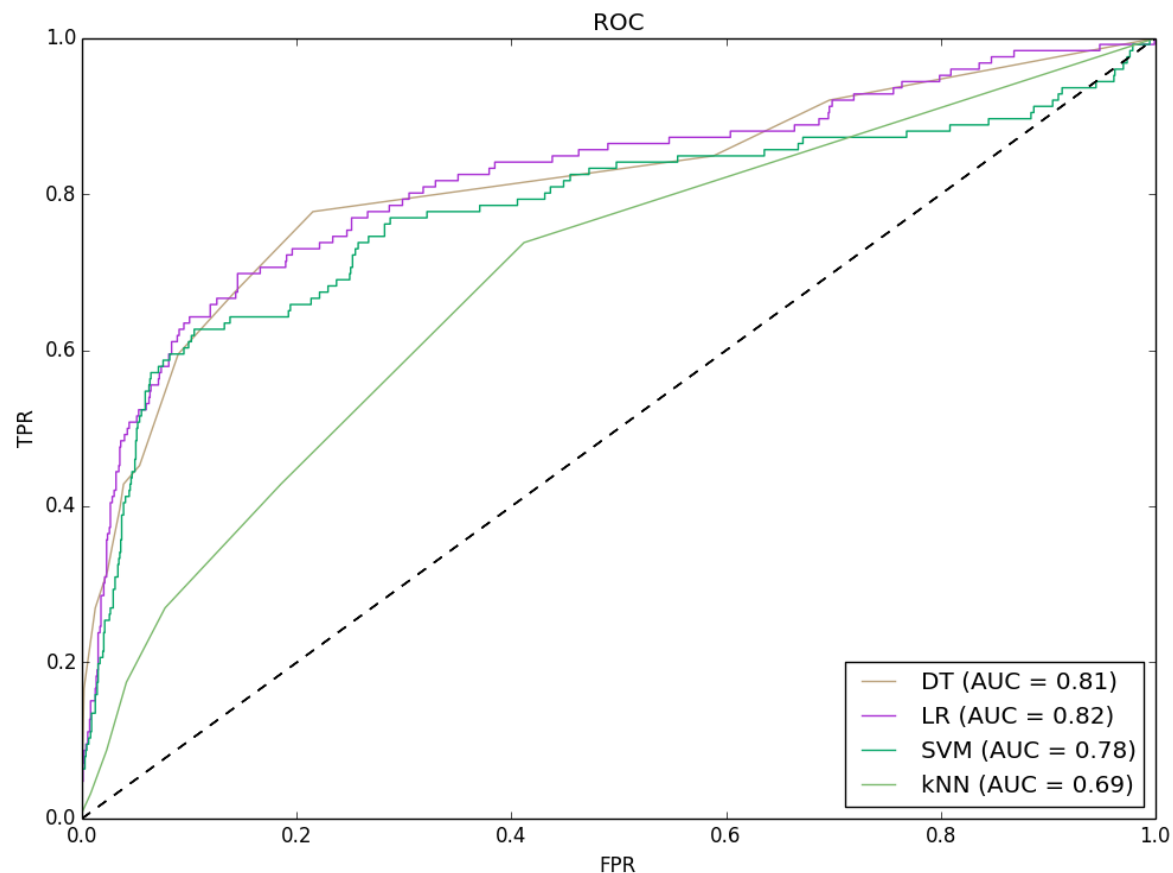
COMPARING AUCS

We built 4 different classifiers using the ads dataset. We can compare the models using ROC analysis.
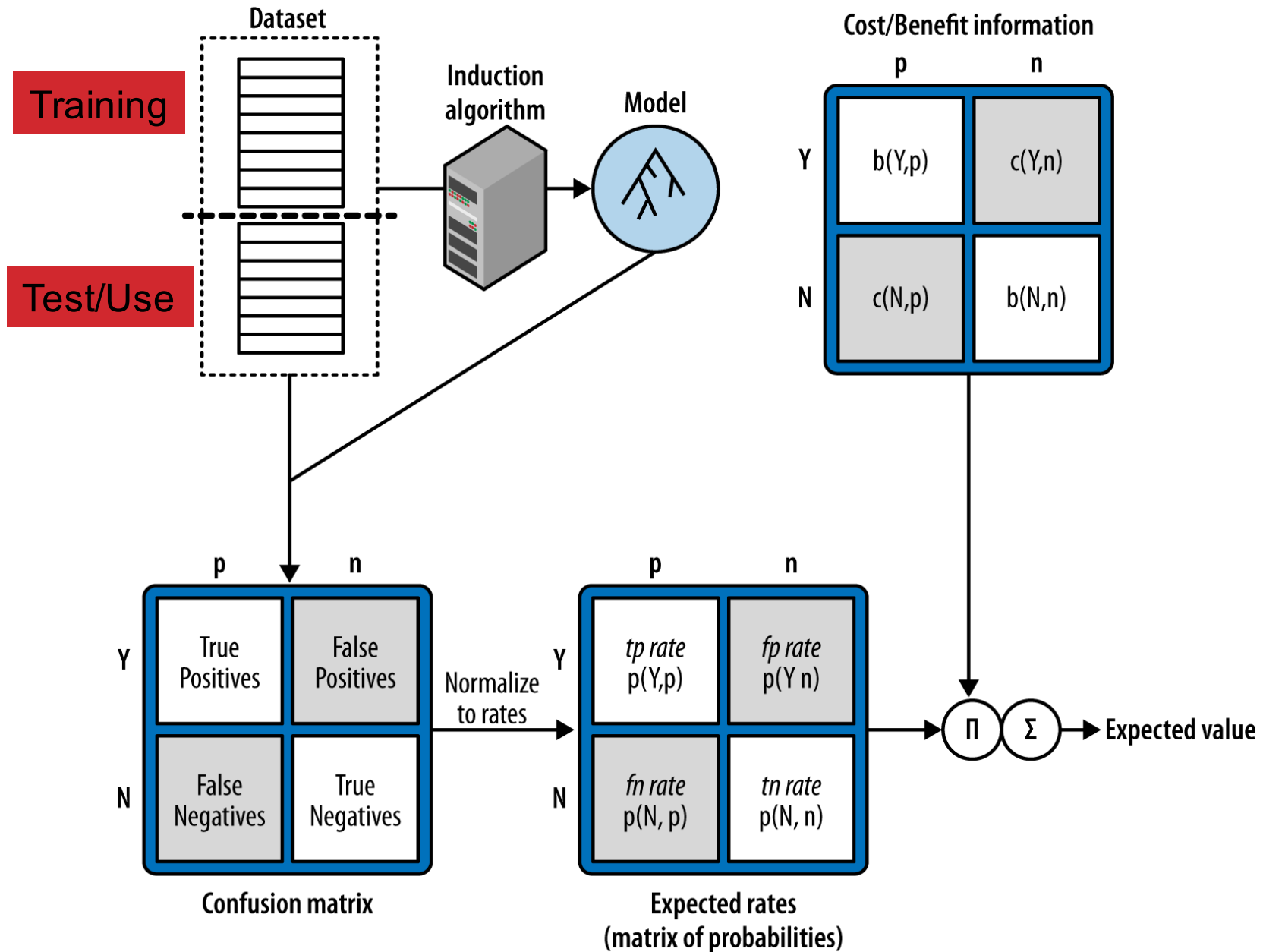
- A universally better model has higher TPR at all FPR (LR > kNN)
- Some models overlap. Better model depends on whether you value TPR or FPR more (DT is best where FPR<0.05)

FUN AUC FACTS

• **Nice interpretation**: gives the probability that a positive instance will have a higher score than a negative instance (equivalent to Mann-Whitney U statistic)

• **Base Rate Invariant:** AUC is invariant to P(+) in the data set (unlike lift metrics). Useful for doing comparisons across data sets with different base rates. Or after down sampling.

• **Is Nicely Bounded:** AUC scores range from [0,1], where 1 is a perfect classifier and 0 is a perfectly wrong classifier. A random classifier has an exact score of 0.5.
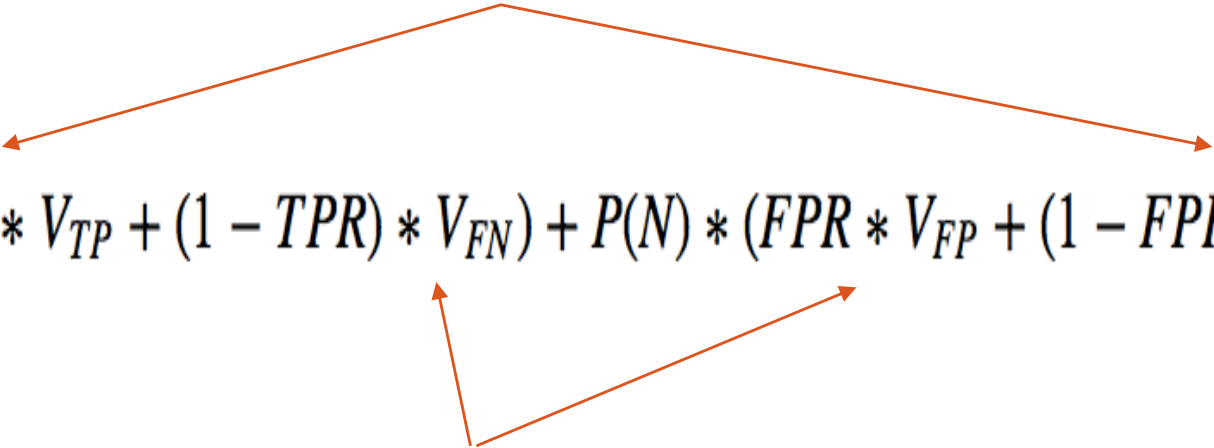
DESIGNING A DECISION PROCESS

Dataset

Training

Test/Use

Induction algorithm

Model

Cost/Benefit information

	p	n
Y	$b(Y,p)$	$c(Y,n)$
N	$c(N,p)$	$b(N,n)$

Confusion matrix

	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives

Normalize to rates

Expected rates (matrix of probabilities)

	p	n
Y	*tp rate* $p(Y,p)$	*fp rate* $p(Y\ n)$
N	*fn rate* $p(N, p)$	*tn rate* $p(N, n)$

Π Σ → **Expected value**

COST SENSITIVE CLASSIFICATION

After learning a classifier's TPR vs FPR curve, and filling in the cost-confusion matrix, we can then compute different expected values to find thresholds that optimize expected values.
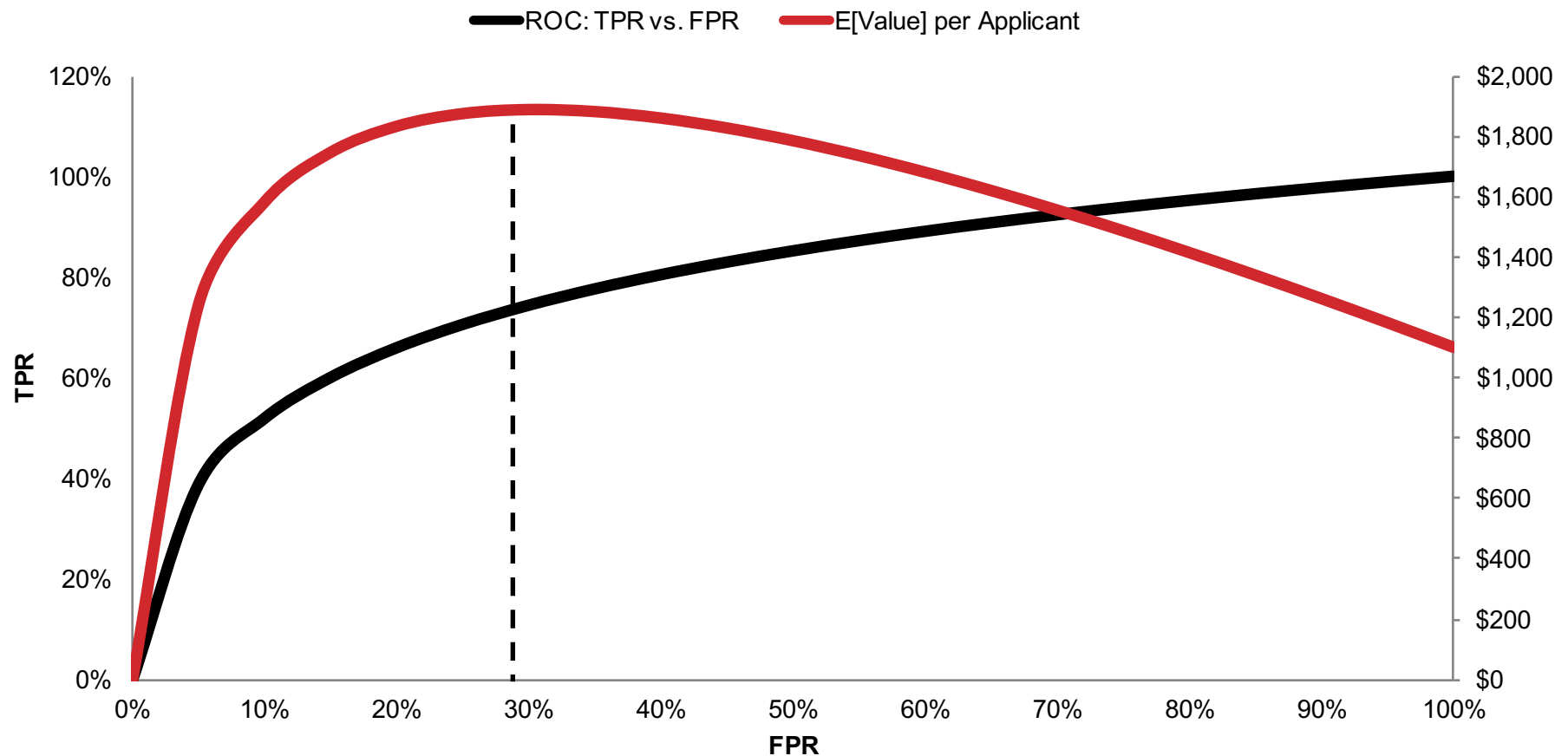
When we are right we generally incur some positive benefit

$$EV = P(Y) * (TPR * V_{TP} + (1 - TPR) * V_{FN}) + P(N) * (FPR * V_{FP} + (1 - FPR) * V_{TN})$$

When wrong we generally incur a negative value (loss)

COST SENSITIVE CLASSIFICATION

Optimizing a Decision Threshold

Legend: ROC: TPR vs. FPR — E[Value] per Applicant

Using the EV formula on the previous slide with [VTP, VFN, VFP, VTN] = [5000, 0, -8000, 0], we can see that an expected value optimizing threshold is one that produces a FPR of 30% and TPR of 74%.

COST SENSITIVE CLASSIFICATION

With a better model we can get more true positives per false positive, and our expected value per classification goes up.

EV Performance Curves for Two Different Models
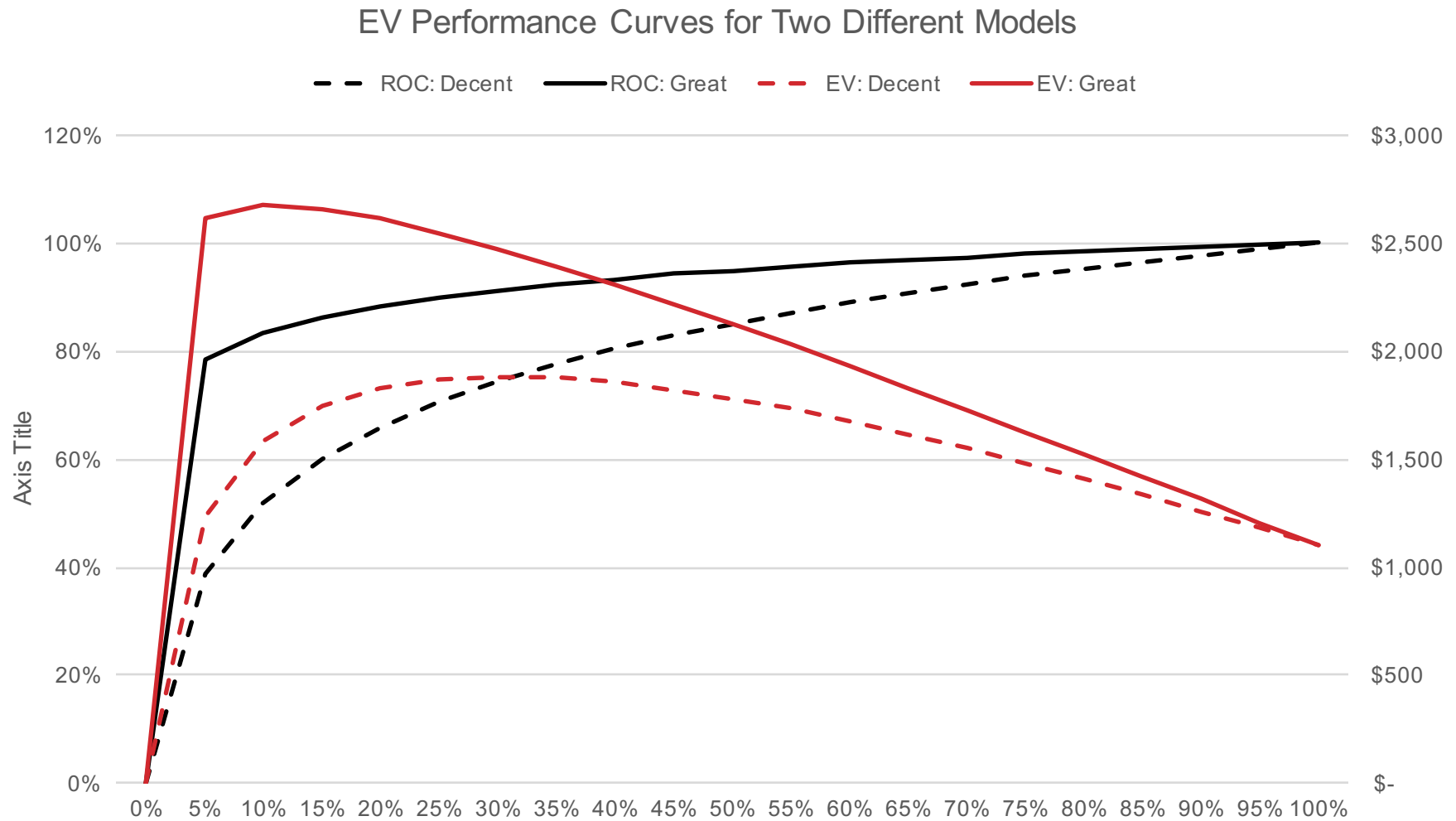
SOME THOUGHT STARTERS

For each of the following predictive modeling scenarios, answer the following:

1. Give a qualitative description, in terms relevant to the application domain, for a false positive and a false negative.
2. Make an assessment on the relative costs of a FP and a FN
3. If you were in charge of deploying the model (assume the model is fixed), how would you design the deployment system to minimize expected misclassification costs?

Modeling Scenarios:

- A medical screening test that classifies the presence of a brain tumor given fMRI images.

- A fraud detection system that automatically freezes an account if it suspects suspicious activity.

- A credit scoring system that automatically decides whether or not an applicant should receive a credit line.

- An automatic face tagging system for images uploaded to a social network.

LIFT

Lift can be both a ranking metric and a classification metric. For ranking, we can see which model fits the entire distribution of users better. For single classification, we can measure lift for a desired targeting threshold.

A Lift Chart on Ads Data with LR

Lift Properties

• **Nice interpretation**: the lift tells you exactly how many more positive outcomes you might expect relative to the baseline strategy. Also lends well to economic analysis

• **Base Rate Non-Invariant:** Lift will change if you alter P(+). This has implications for down sampling or for comparing models from different datasets.
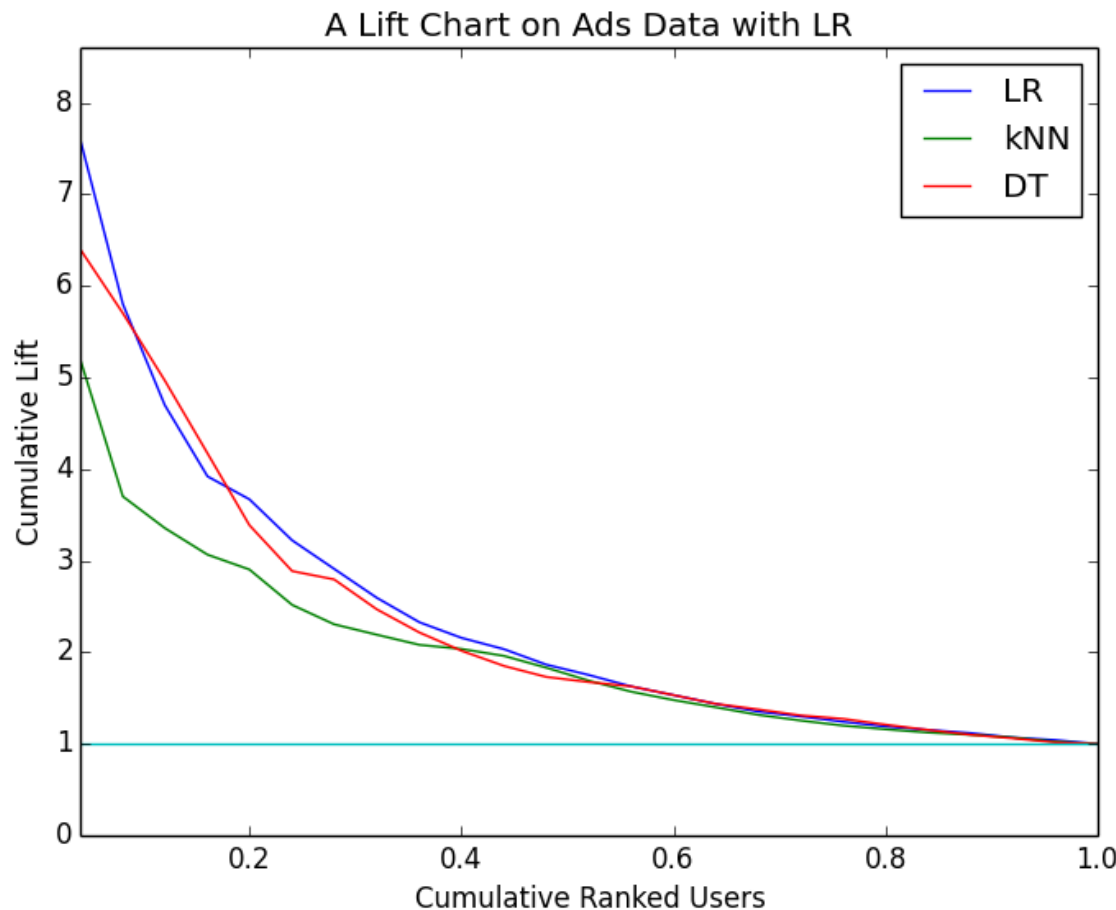
DENSITY ESTIMATION

Sometimes you want to evaluate how well your model estimates the underlying conditional distribution of your data: P(Y|X)

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |E[y|x_i] - y_i|$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (E[y|x_i] - y_i)^2$$

$$LL = \frac{1}{n} \sum_{i=1}^{n} y_i \, ln(P(y|x_i)) + (1 - y_i) \, ln(1 - P(y|x_i))$$

BACK TO ADS DATA

Scenario 1:

Constraints: a budget *k* and a population *n* (*k* and *n* on the same unit scale)

Goal: Maximize the ROI for the client

Solution: Target (k/n)% of the population, such that the selected set of *k* prospects maximizes the total number of conversions

What metrics can we use to choose the best model

BACK TO ADS DATA

Scenario 1:

Constraints: a budget k and a population n (k and n on the same unit scale)

Goal: Maximize the ROI for the client

Solution: Target (k/n)% of the population, such that the selected set of k prospects maximizes the total number of conversions

If we know k and n: **Lift or Precision**
If we don't know k and n: **AUC**

ROC CURVE OF INDIVIDUAL FEATURES

We can analyze the predictive power of individual features using AUC curves. Note the interesting shape of LAST_BUY AUC. What causes that?
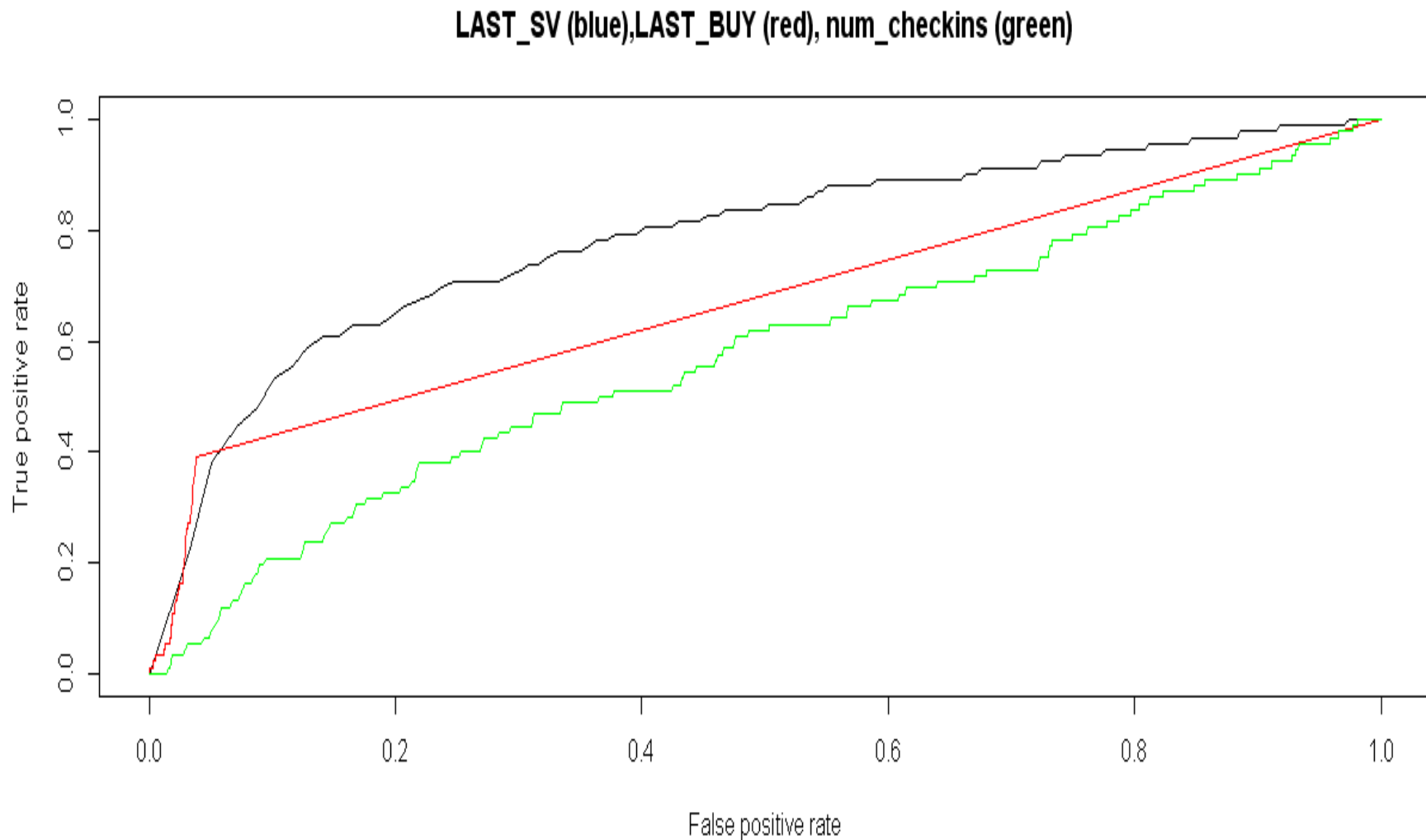
LAST_SV (blue),LAST_BUY (red), num_checkins (green)

BACK TO ADS DATA

Scenario 2:

Constraints: each impression costs $C, for each conversion, receive $Q, unlimited budget

Goal: Maximize profit for the firm

Solution: Target every opportunity where
$$E[Value] = P(Conv|X)*\$Q > \$C$$

What metrics can we use to choose the best model

BACK TO ADS DATA

Scenario 2:

Constraints: each impression costs $C, for each conversion, receive $Q, unlimited budget

Goal: Maximize profit for the firm

Solution: Target every opportunity where
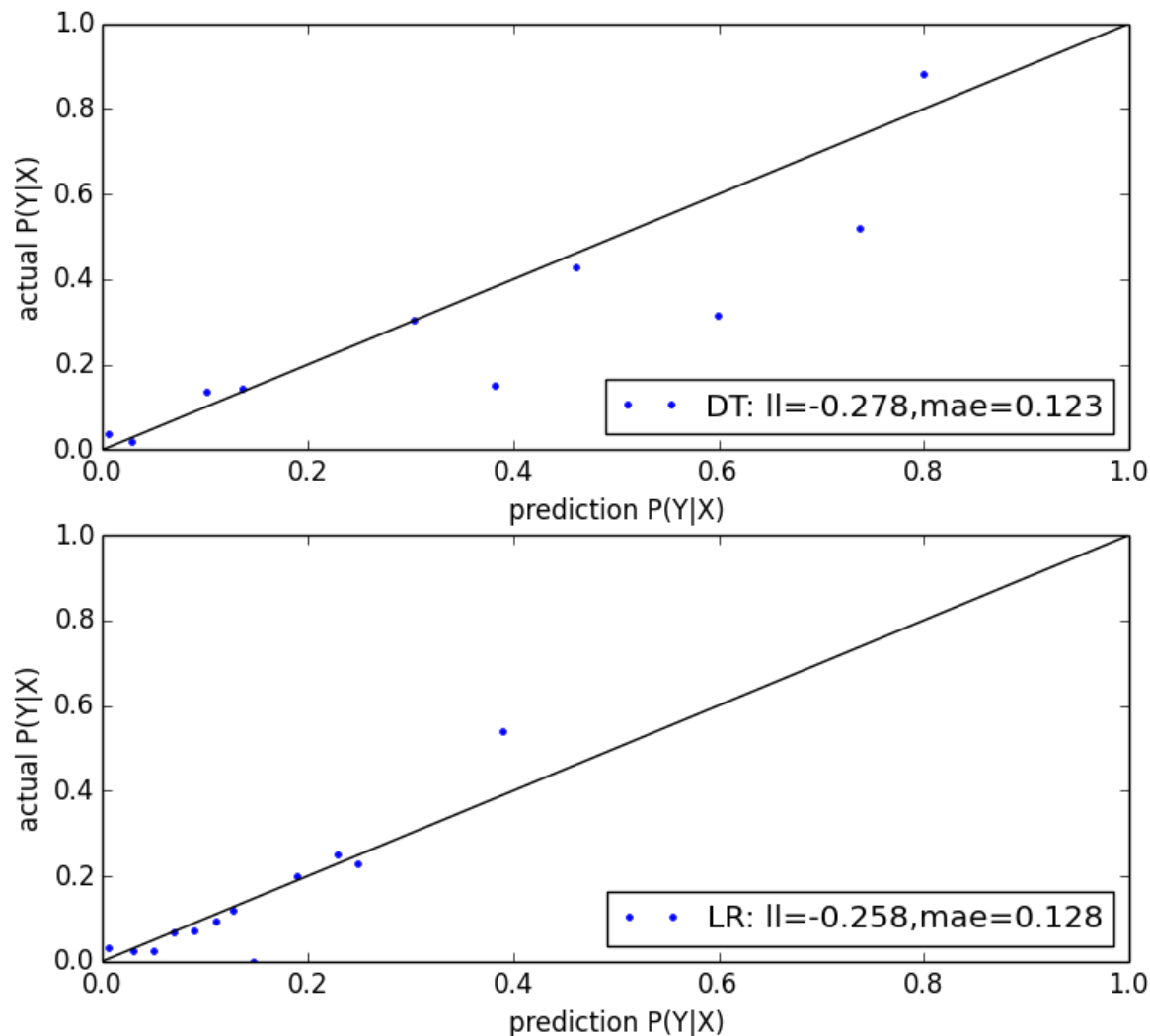$$E[Value]= P(Conv|X)*\$Q>\$C$$

To get a well calibrated estimate of P(Y|X), use

$$LL = \frac{1}{n} \sum_{i=1}^{n} y_i \ ln(P(y|x_i)) + (1 - y_i) \ ln(1 - P(y|x_i))$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |E[y|x_i] - y_i|$$

CALIBRATION PLOTS

We can generate calibration plots to visually inspect how well the predictions match the outcomes. To make the plot, we bin test instances by P(Y|X) and take mean(Y) against mean(P(Y|X)) for each bin.

Observations:

- DT predicts a higher range of probabilities

- LR is very well calibrated for lower valued predictions but not as good in the upper range.

- Why is MAE lower for DT?

METRICS DON'T ALWAYS AGREE

It is often the case that different metrics don't agree (in terms of rank) when comparing models built with different design choices.

In this case we build univariate LR models on each feature and compare AUC, LL and Gini Index (from SK Learn Decision Tree)

Feature	AUC	-LL	Gini
visit_freq	0.781	0.282	0.147
last_visit	0.780	0.306	0.528
multiple_visit	0.740	0.280	0.000
sv_interval	0.717	0.323	0.046
buy_freq	0.673	0.274	0.151
isbuyer	0.670	0.278	0.000
last_buy	0.665	0.321	0.015
buy_interval	0.581	0.310	0.000
multiple_buy	0.581	0.297	0.000
uniq_urls	0.580	0.322	0.051
num_checkins	0.567	0.326	0.062
expected_time_visit	0.564	0.329	0.000
expected_time_buy	0.518	0.327	0.000