

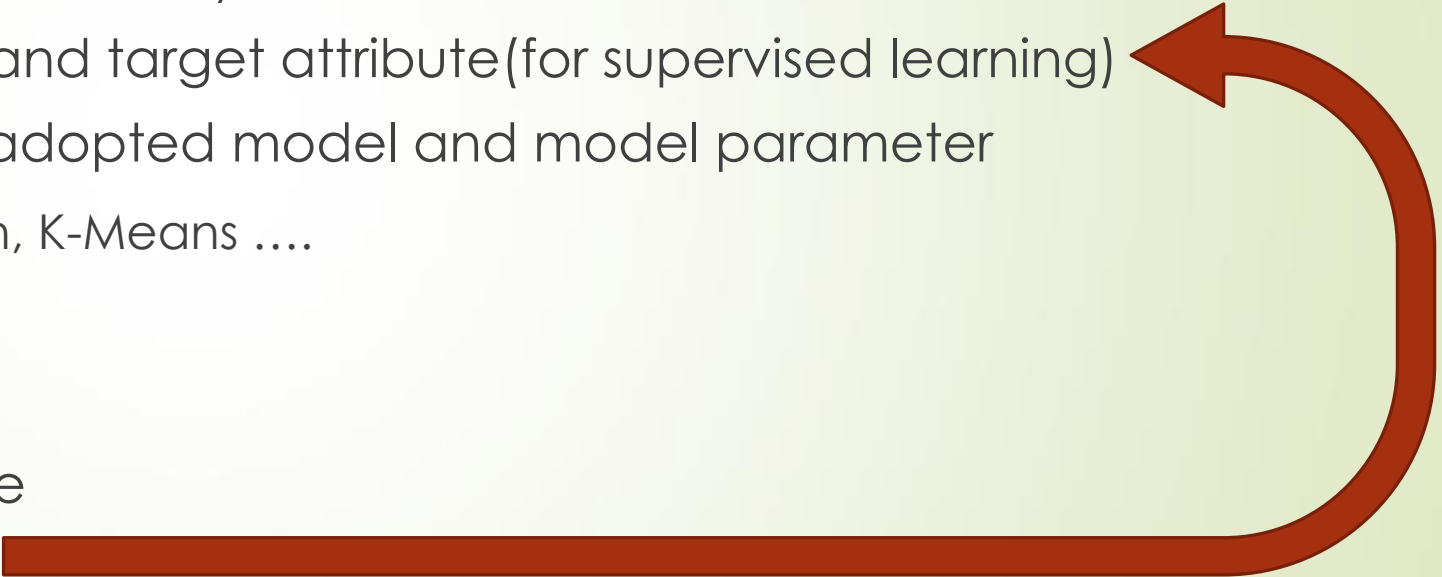
What is a Good Model

1

Dr. Yi Long (Neal)

Most contents (text or images) of course slides are from the following textbook
Provost, Foster, and Tom Fawcett. Data Science for Business: What you need to know about data mining and data-analytic thinking. " O'Reilly Media, Inc.", 2013

General Working Flow For Prediction

- Read data: supposed to be structured data
 - Explore/ Pre-Process Data : handling missing data, class imbalance ...
 - ✓ Supposed to be preprocessed already in this lab course
 - Select attributes as features and target attribute(for supervised learning)
 - Build/Initialize model: select adopted model and model parameter
 - ✓ SVM, KNN, LogisticRegression, K-Means
 - Fit model to data
 - ✓ Training supervised model
 - Evaluate model performance
 - ✓ Prediction accuracy
 - Apply the final model to predict unseen data
- 

General Working Flow For Prediction

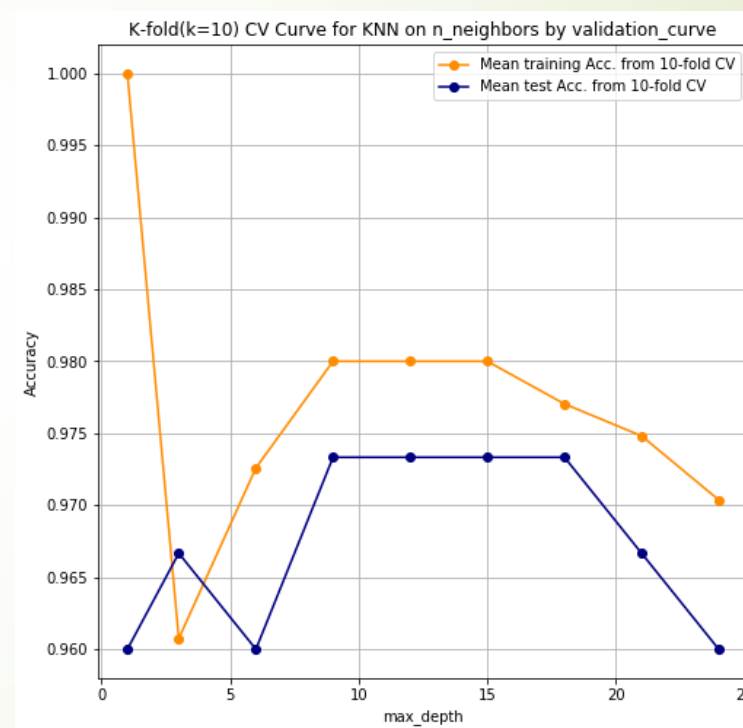
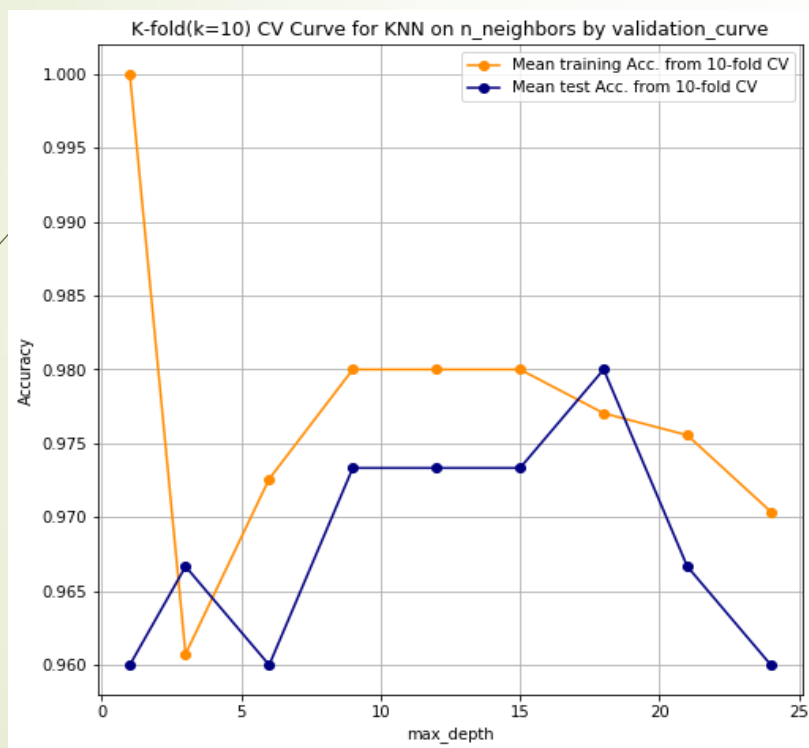
- Read data: supposed to be structured data
- Explore/ Pre-Process Data : handling missing data, class imbalance ...
 - ✓ Supposed to be preprocessed already in this lab course
- Select attributes as features and target attribute(for supervised learning)
- Build/Initialize model: select adopted model and model parameter
 - ✓ SVM, KNN, LogisticRegression, K-Means
- Fit model to data
 - ✓ Training supervised model
- Evaluate model performance
 - ✓ Prediction accuracy
- Apply the final model to predict unseen data

sklearn.model_selection.
GridSearchCV



Q1-2 in Lab 7

Choose bet **K** for KNN



Occam's razor: choose the **simplest** model/hypothesis that can fit the data well

k as Complexity Parameter of KNN

- Number k controls the complexity of KNN : lower k ➔ more complex model
- Choose optimal k by (nested) cross-validation

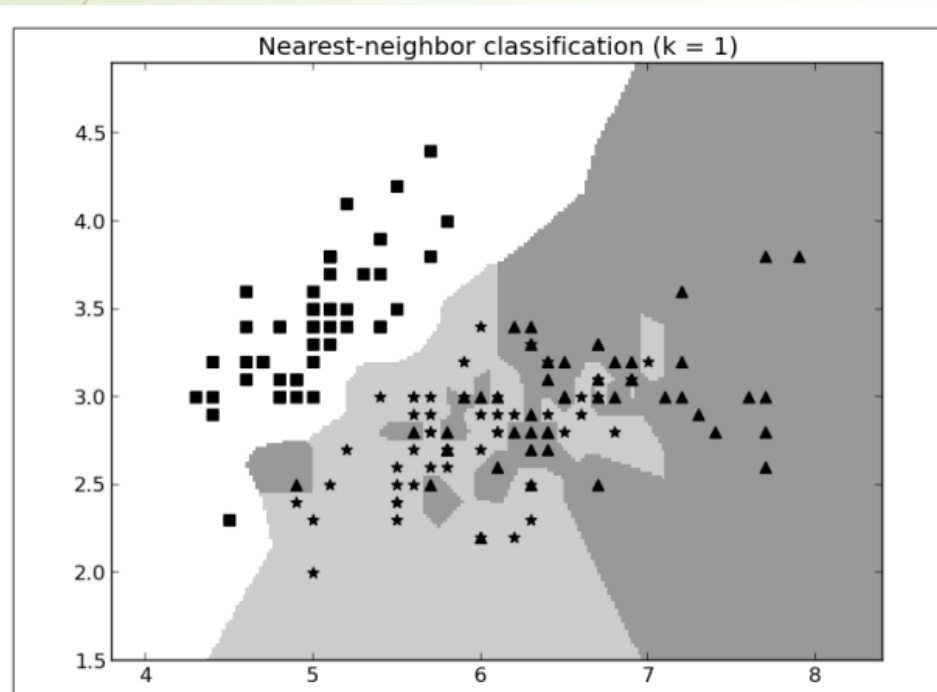


Figure 6-4. Classification boundaries created on a three-class problem created by 1-NN (single nearest neighbor).

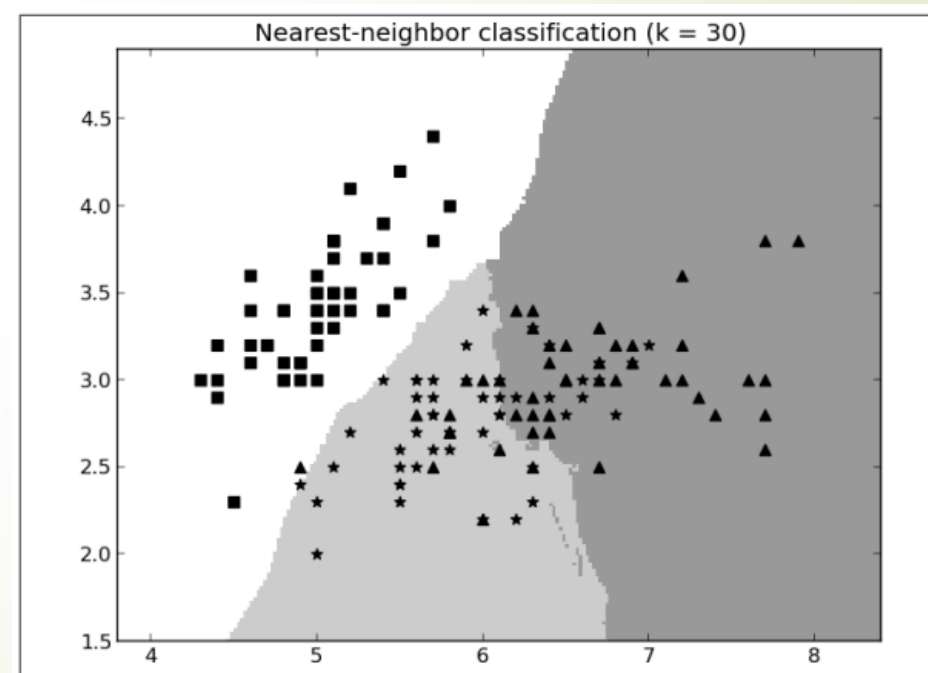


Figure 6-5. Classification boundaries created on a three-class problem created by 30-NN (averaging 30 nearest neighbors).

Group Project

- 3-4 students each group
- Any one of the suggested topics OR other but subject to approval within this week
- Submission:
 - Report (no more than 10 pages, with earlier deadline for reviewing)
 - Presentation (10+ 2 mins) + Project Code (with dataset)
- Grade
 - Relevance, novelty, in-depth and rigorous evaluation (focused)
 - At least **TWO** different approaches (with **at least one baseline model** for comparison)

Outline

- Confusion Matrix
- A Key Analytical Framework & Baseline
- Visualizing Model Performance

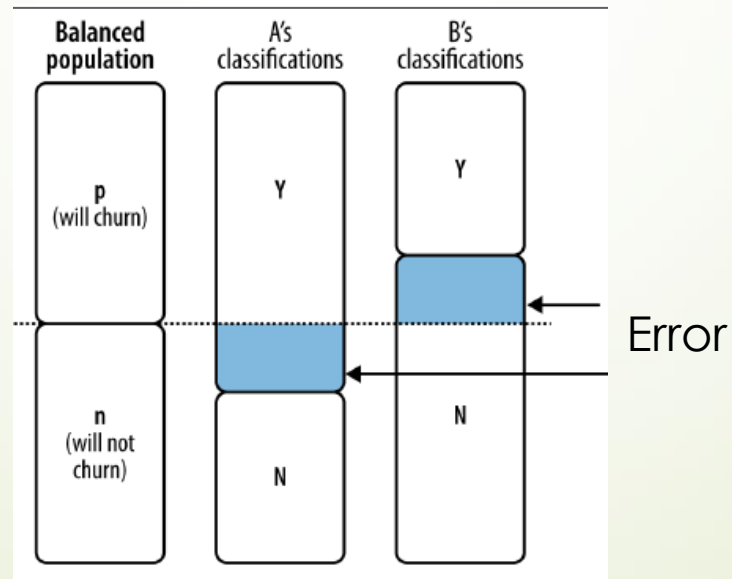
Plain Accuracy

- Too simplistic

$$\text{Accuracy} = \frac{\text{Number of correct decisions made}}{\text{Total number of decisions made}}$$

$$= 1 - \text{error rate}$$

- Model A and B has same accuracy, but ...



Positives and Negatives

- Classification terminology: we usually think of
 - ✓ A **positive** example as one worthy of attention or alarm
 - ✓ A **negative** example as uninteresting or benign
- Positive examples are usually bad outcomes
 - ✓ Medical test: positive test ➡ disease is present
 - ✓ Fraud detector: positive test ➡ unusual activity on account
- A classifier tries to distinguish the majority of cases(negatives, the uninteresting) from the small number of alarming cases(positives, alarming)
 - ✓ The positive class is often rare, or at least rarer than the negative class
 - ✓ Number of mistakes made on negative examples(false positive errors) will be relatively high, while cost of each mistake made on a positive example(false negative error) will be relatively high

Confusion Matrix

- Decompose and count the different types of correct and incorrect decisions made by a classifier
- A **confusion matrix** for a problem involving n classes is an $n \times n$ matrix
 - ✓ With the columns labeled with actual classes and the rows labeled with predicted classes

		Actual	
		p	n
Predicted	Y	True positives	False positives
	N	False negatives	True negatives

- It explicit shows how one class is being confused for another
 - ✓ With the columns labeled with actual classes and the rows labeled with predicted classes
 - ✓ The main diagonal contains the count of correct decisions (TP and TN), while the errors of the classifier are the false positives (FP) and false negatives (FN)

Building Confusion Matrix

- Each example in a test set has an actual class label and the class predicted by the classifier

Default Truth	Model Prediction
0	0
1	1
0	1
0	1
0	0
1	1
0	0
0	0
1	1
1	0

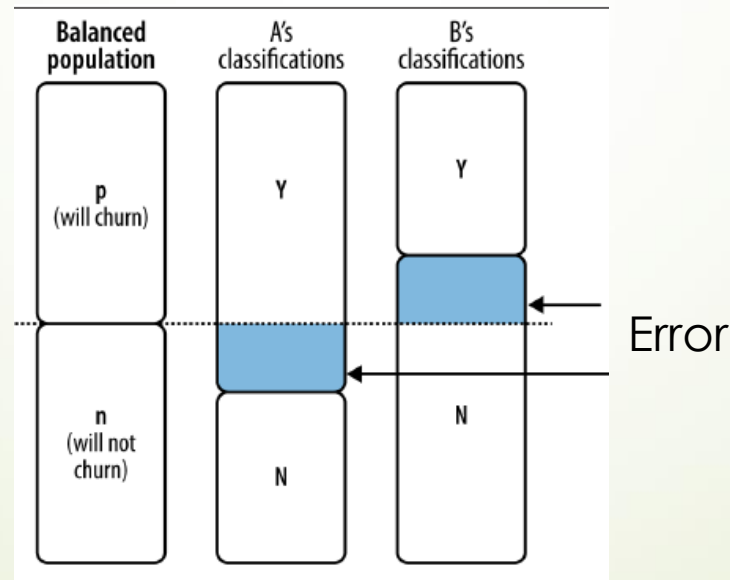


		Yes	No	
Predicted class Actual class		Default	No Default	Total
Positive	Default	3	1	4
	No Default	2	4	6
Total		5	5	10

Confusion Matrix of Model A and B

$$CM_A = \begin{matrix} & \text{churn} & \text{not churn} \\ \begin{matrix} Y \\ N \end{matrix} & \begin{pmatrix} 500 & 200 \\ 0 & 300 \end{pmatrix} \end{matrix}$$

$$CM_B = \begin{matrix} & \text{churn} & \text{not churn} \\ \begin{matrix} Y \\ N \end{matrix} & \begin{pmatrix} 300 & 0 \\ 200 & 500 \end{pmatrix} \end{matrix}$$

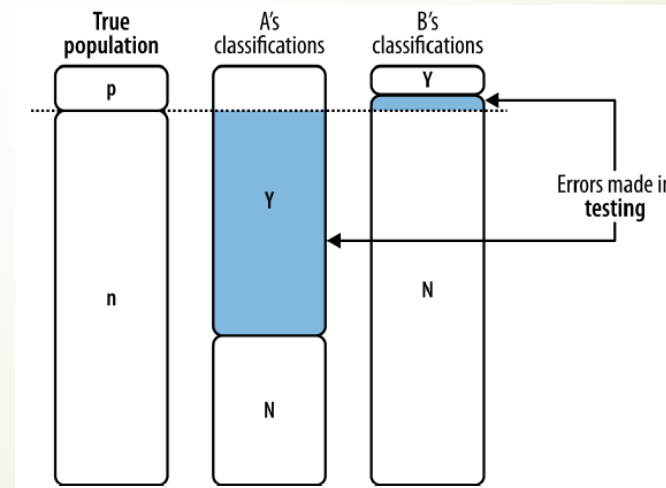
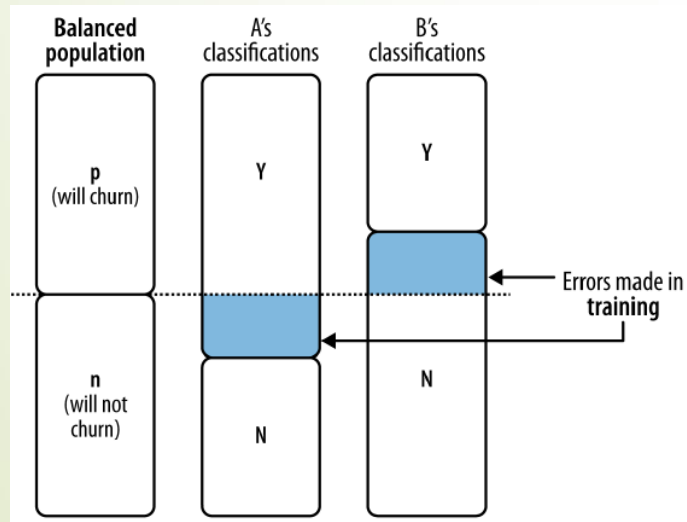


When the Population is Unbalanced

- In practical classification problems, one class is often rare (positive cases)
 - ✓ The unusual or interesting class is rare among the general population, the class distribution is unbalanced or skewed
 - ✓ Skews of 1:100 are common in fraud detection, and skews greater than 1:10⁶ have been reported in other classifier learning applications
- Evaluation based on accuracy does not work
 - ✓ We can easily get 99.9% accuracy by always predicting a case to be normal
- How about training models on a manually-built balanced dataset?

Understanding Population Matters

- We need to know more details about the population
- When only 10% of users will churn, we insist on training the model based on a manually-built balanced dataset
 - ✓ Classifier A often falsely predicts that customers will churn (not churned \rightarrow churned by prediction)
 - ✓ Classifier B often falsely predicts that customers will not churn (churned \rightarrow not churned by prediction)



Unequal Costs and Benefits

- How much do we care about the different **errors** and correct decisions?
 - ✓ Classification accuracy makes no distinction between false positive and false negative errors
 - ✓ In real-world applications, different kinds of errors lead to different consequences!
- Examples for medical diagnosis:
 - ✓ A patient has cancer(although he does not) ➤ false positive error, expensive, but not life threatening
 - ✓ A patient has cancer, but he is told that she has not ➤ false negative error, more serious
- Errors should be counted separately
 - ✓ Estimate cost or benefit of each decision

Look beyond Classification

- How to measure the accuracy/ quality of a regression model ?
 - ✓ Predict how much a given customer will like a given movie (1 star to 5 stars)
- Typical accuracy of regression: mean-squared error
- What does the mean-squared error describe?
 - ✓ Value of the target variable, e.g., the number of stars that a user would give as a rating for the movie
- Is the mean-squared error a meaningful metric?



Outline

- Confusion Matrix
- A Key Analytical Framework & Baseline
- Visualizing Model Performance

Expected Value

- The expected value computation provides a framework that is useful in organizing thinking about data-analytic problems
- The general form of an expected value calculation:

$$EV = p(o_1) \cdot v(o_1) + p(o_2) \cdot v(o_2) + p(o_3) \cdot v(o_3) \dots$$

- Examples for marketing revenue(traditional targeted marketing,):
 - ✓ Product Price: \$200; Product Cost: \$100; Targeting Cost: \$1
 - ✓ The probability of response for consumer \mathbf{X} in general is $P_R(\mathbf{X})$
 - ✓ Then $V_R = 200 - 100 - 1 = \$99$, $V_{NR} = -\$1$

$$\begin{aligned}\text{Expected benefit of targeting} &= p_R(\mathbf{x}) \cdot v_R + [1 - p_R(\mathbf{x})] \cdot v_{NR} \\ &= p_R(\mathbf{x}) \cdot \$99 - [1 - p_R(\mathbf{x})] \cdot \$1\end{aligned}$$

Cost-benefit Matrix for Classification

- ▶ A cost-benefit matrix specifies the cost or benefit for each(predicted, actual) pair
 - ✓ Correct classifications(true positives and negatives) correspond to $b(Y,p)$ and $b(N,n)$, respectively
 - ✓ Incorrect classifications(false positives and negatives) correspond to $c(Y,n)$ and $c(N,p)$, respectively [often negative benefits or costs]
 - ✓ Costs and benefits **cannot** be estimated from data

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

Example of Cost-benefit Matrix

Targeted marketing example :

- ✓ False positive occurs when we classify a consumer as a likely responder and therefore target him, but he does not respond ➡ benefit/cost $b(Y,n)=-1$
- ✓ False negative is a consumer who was predicted not to be a likely responder, but would have bought if offered. No money spent, nothing gained ➡ benefit $b(N,p)=0$
- ✓ True positive is a consumer who is offered the product and buys it ➡ benefit $b(Y,p)=200-100-1=99$
- ✓ True negative is a consumer who was not offered a deal but who would not have bought it ➡ benefit $b(N,n)=0$

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



		Actual	
		p	n
Predicted	Y	99	-1
	N	0	0

Probability of Different Decisions

- The probability of making different decisions can be estimated from confusion matrix
 - ✓ Normalize the count of observations to rate

Predicted	Actual	
	p	n
	Y	N
Y	56	7
N	5	42



$$T = 110$$

$$p(\mathbf{Y}, \mathbf{p}) = 56/110 = 0.51 \quad p(\mathbf{Y}, \mathbf{n}) = 7/110 = 0.06$$

$$p(\mathbf{N}, \mathbf{p}) = 5/110 = 0.05 \quad p(\mathbf{N}, \mathbf{n}) = 42/110 = 0.38$$

Expected Profit

- Given the probability of making different decisions and corresponding benefit/cost, we can calculate the expected profit as

$$\text{Expected profit} = p(\mathbf{Y}, \mathbf{p}) \cdot b(\mathbf{Y}, \mathbf{p}) + p(\mathbf{N}, \mathbf{p}) \cdot b(\mathbf{N}, \mathbf{p}) + p(\mathbf{N}, \mathbf{n}) \cdot b(\mathbf{N}, \mathbf{n}) + p(\mathbf{Y}, \mathbf{n}) \cdot b(\mathbf{Y}, \mathbf{n})$$

		Actual	
		p	n
Predicted	Y	99	-1
	N	0	0

$$\begin{aligned} p(\mathbf{Y}, \mathbf{p}) &= 56/110 = 0.51 & p(\mathbf{Y}, \mathbf{n}) &= 7/110 = 0.06 \\ p(\mathbf{N}, \mathbf{p}) &= 5/110 = 0.05 & p(\mathbf{N}, \mathbf{n}) &= 42/110 = 0.38 \end{aligned}$$



$$\Rightarrow \text{Expected profit} = 99 \cdot 0.51 - 1 \cdot 0.06 = 50.43$$

Factoring out Class Priors

- Factoring the class priors out allows us to separate the influence of class imbalance from the fundamental predictive power of the model
 - ✓ The class priors, $p(p)$ and $p(n)$, specify the likelihood of seeing positive and negative instances, respectively.
- A rule of basic probability: $p(x, y) = p(y) \cdot p(x \mid y)$

$$\text{Expected profit} = p(\mathbf{Y}, \mathbf{p}) \cdot b(\mathbf{Y}, \mathbf{p}) + p(\mathbf{N}, \mathbf{p}) \cdot b(\mathbf{N}, \mathbf{p}) + p(\mathbf{N}, \mathbf{n}) \cdot b(\mathbf{N}, \mathbf{n}) + p(\mathbf{Y}, \mathbf{n}) \cdot b(\mathbf{Y}, \mathbf{n})$$



$$\text{Expected profit} = p(\mathbf{Y} \mid \mathbf{p}) \cdot p(\mathbf{p}) \cdot b(\mathbf{Y}, \mathbf{p}) + p(\mathbf{N} \mid \mathbf{p}) \cdot p(\mathbf{p}) \cdot b(\mathbf{N}, \mathbf{p}) + p(\mathbf{N} \mid \mathbf{n}) \cdot p(\mathbf{n}) \cdot b(\mathbf{N}, \mathbf{n}) + p(\mathbf{Y} \mid \mathbf{n}) \cdot p(\mathbf{n}) \cdot b(\mathbf{Y}, \mathbf{n})$$



$$\text{Expected profit} = p(\mathbf{p}) \cdot [p(\mathbf{Y} \mid \mathbf{p}) \cdot b(\mathbf{Y}, \mathbf{p}) + p(\mathbf{N} \mid \mathbf{p}) \cdot c(\mathbf{N}, \mathbf{p})] + p(\mathbf{n}) \cdot [p(\mathbf{N} \mid \mathbf{n}) \cdot b(\mathbf{N}, \mathbf{n}) + p(\mathbf{Y} \mid \mathbf{n}) \cdot c(\mathbf{Y}, \mathbf{n})]$$

Example

Predicted	Actual	
	p	n
	Y	N
Y	56	7
N	5	42



$$T = 110$$

$$P = 61$$

$$p(p) = 0.55$$

$$p(Y|p) = 56/61 = 0.92$$

$$p(N|p) = 5/61 = 0.08$$

$$N = 49$$

$$p(n) = 0.45$$

$$p(Y|n) = 7/49 = 0.14$$

$$p(N|n) = 42/49 = 0.86$$



$$T = 110$$

$$P = 61$$

$$p(\mathbf{p}) = 0.55$$

$$tp\ rate = 56/61 = 0.92$$

$$fn\ rate = 5/61 = 0.08$$

$$N = 49$$

$$p(\mathbf{n}) = 0.45$$

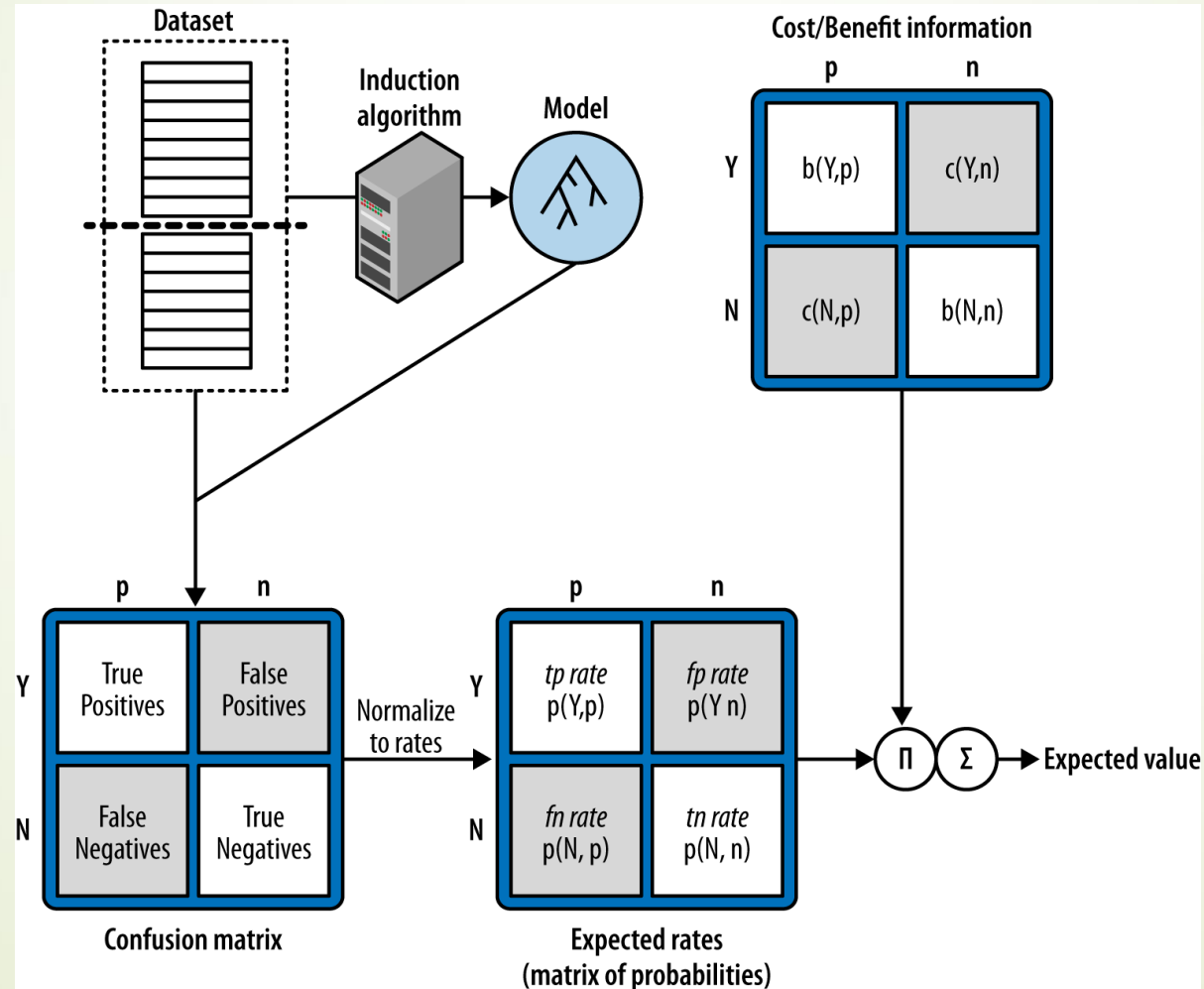
$$fp\ rate = 7/49 = 0.14$$

$$tn\ rate = 42/49 = 0.86$$



$$\begin{aligned}
 \text{expected profit} &= p(\mathbf{p}) \cdot [p(Y | \mathbf{p}) \cdot b(Y, \mathbf{p}) + p(N | \mathbf{p}) \cdot c(N, \mathbf{p})] + \\
 &\quad p(\mathbf{n}) \cdot [p(N | \mathbf{n}) \cdot b(N, \mathbf{n}) + p(Y | \mathbf{n}) \cdot c(Y, \mathbf{n})] \\
 &= 0.55 \cdot [0.92 \cdot b(Y, \mathbf{p}) + 0.08 \cdot b(N, \mathbf{p})] + \\
 &\quad 0.45 \cdot [0.86 \cdot b(N, \mathbf{n}) + 0.14 \cdot b(Y, \mathbf{n})] \\
 &= 0.55 \cdot [0.92 \cdot 99 + 0.08 \cdot 0] + \\
 &\quad 0.45 \cdot [0.86 \cdot 0 + 0.14 \cdot -1] \\
 &= 50.1 - 0.063 \\
 &\approx \$50.04
 \end{aligned}$$

Expected Value Calculation



Baselines for Comparisons

- Consider what would be a reasonable baseline against which to compare model performance
 - ✓ Demonstrate stakeholder that the model has added value
- This task depends on the actual application and is also essential
- General alternative baseline
 - ✓ Existing solutions
 - ✓ Completely random model / majority classifier
 - ✓ Decision stump: a decision tree with only one internal node
 - ✓ A simple (but not simplistic) alternative model: tomorrow weather as today
 - ✓ Domain knowledge

Other Metrics

- Based on the entries of the confusion matrix, we can describe various evaluation metrics
- Usually used in pairs and then are comprehensive than accuracy

		p	Actual	n
Predicted	Y	True positives	False positives	
	N	False negatives	True negatives	

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

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

$$\text{True Positive Rate} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN}$$

$$\text{F-measure (harmonic mean): } 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{Sensitivity: } \frac{TN}{TN+FP}$$

$$\text{Specificity: } \frac{TP}{TP+FN}$$

Outline

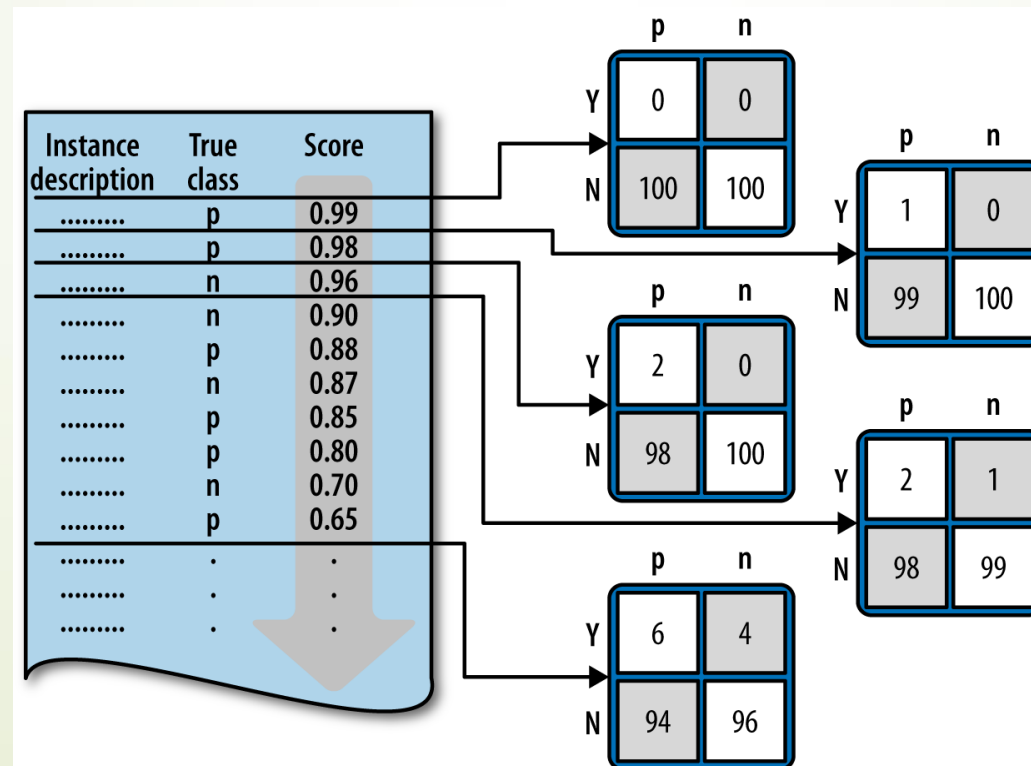
- Confusion Matrix
- A Key Analytical Framework & Baseline
- Visualizing Model Performance

Evaluation of Ranking Models

- Expected profit is instructive, but
 - ✓ Precise knowledge of the costs and benefits
 - ✓ Accurate estimation of case probabilities
- It is often useful to present visualizations rather than just calculations
- Ranking is more helpful than classifying, but
 - ✓ How do we compare different rankings?
 - ✓ Do we choose a proper threshold ?

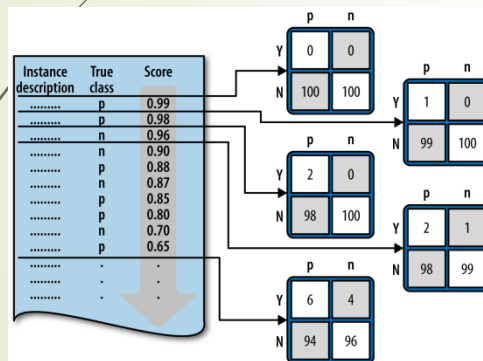
Confusion Matrix Based on Ranking

- By setting any threshold on the score, we can produce a single confusion matrix
- ✓ Score can be the distance from decision boundaries (linear classification)

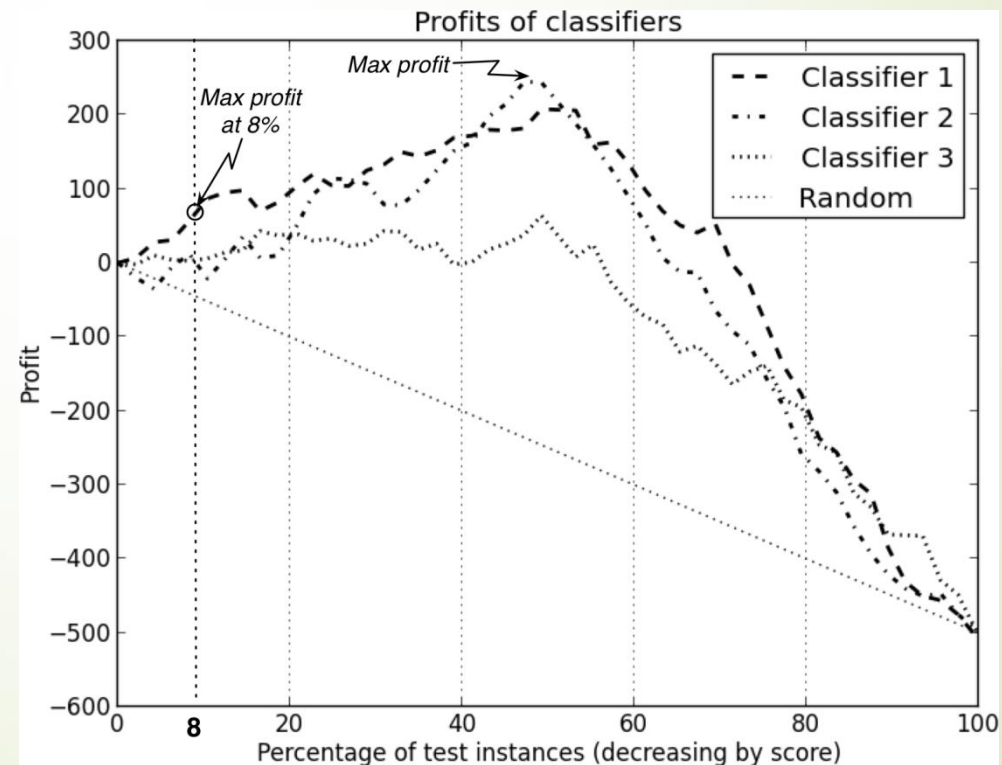


Profit Curves

- Profit Curves are informative
 - ✓ Target progressively larger proportions of the consumer with decreasing threshold
- We can then compute a list of corresponding expected profit

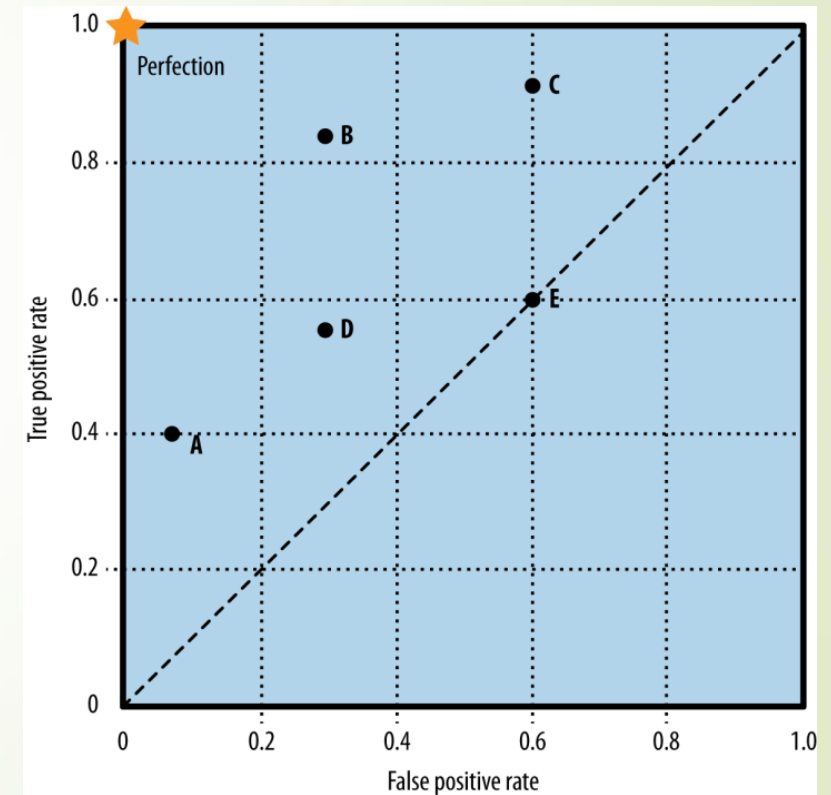


	p	n
Y	\$4	-\$5
N	\$0	\$0



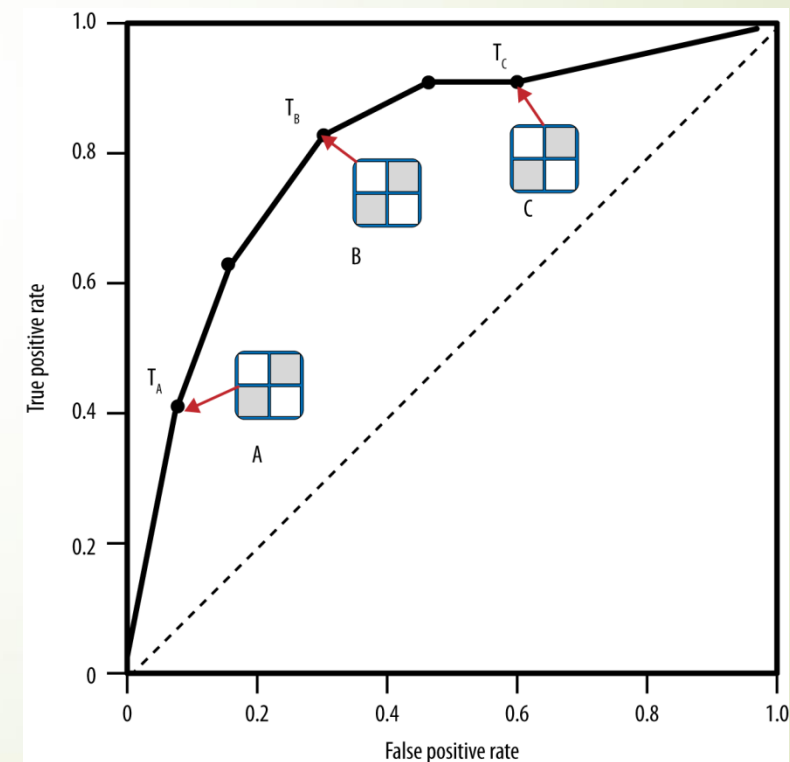
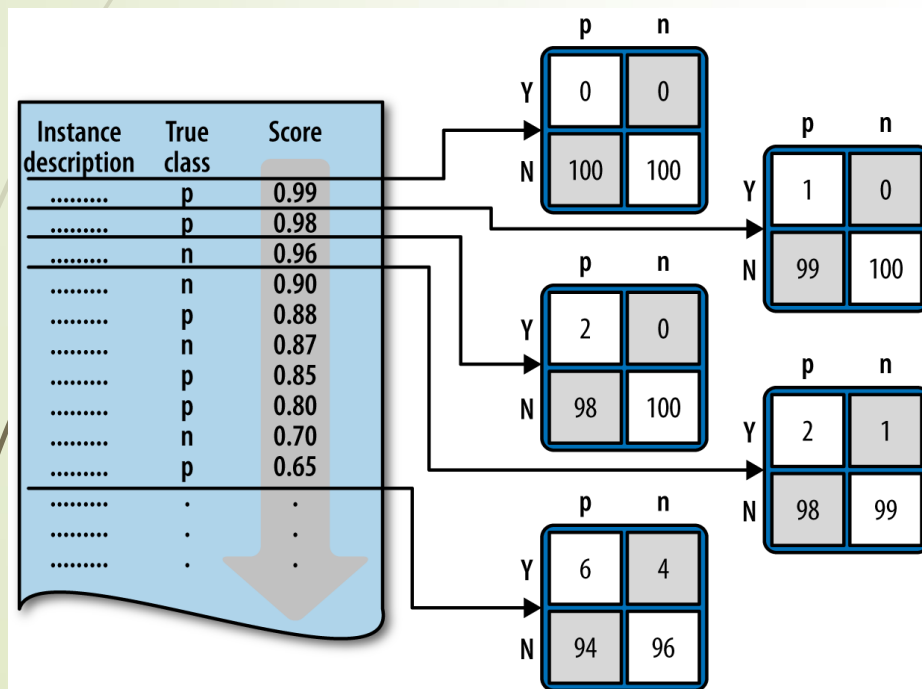
ROC Curves

- Profit curves require exact knowledge of cost-benefit matrix and class priors
- Receiver Operating Characteristics (ROC) graph shows the entire space of performance possibilities
 - ✓ Show the trade-offs that a classifier makes between benefits (true positives) and costs (false positives)
 - ✓ Each discrete classifier produces an (fp rate, tp rate) pair corresponding to a single point in ROC space
 - ✓ One point in ROC space is superior to another if it is to the northwest/top-left of the first



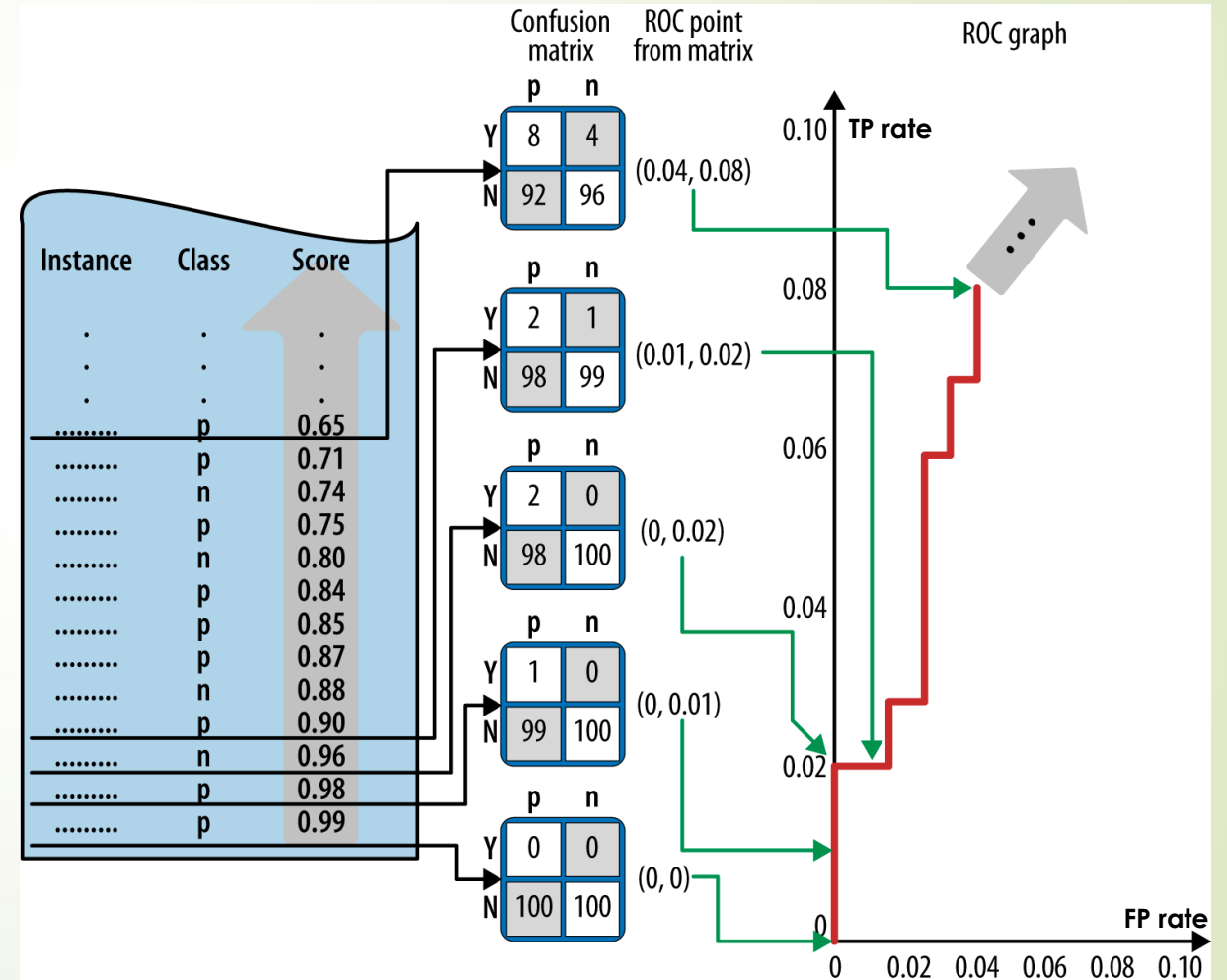
ROC Curves of Ranking Models

- Each threshold value produces a different point in ROC space



ROC Curves of Ranking Models

- The “curve” is actually a step function for a single test
- ✓ Whenever we pass a positive instance, we take a step upward (increasing true positives)
- ✓ Whenever we pass a negative instance, we take a step rightward (increasing false positives)
- ✓ ROC graphs are independent of the class proportions as well as the costs and benefits

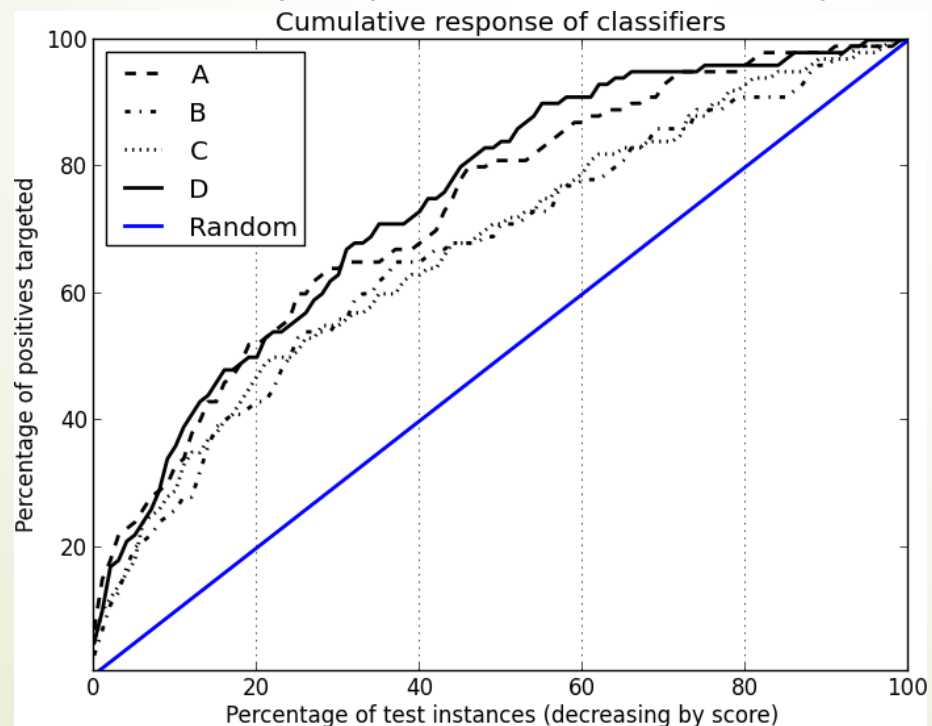


Area Under the ROC Curve (AUC)

- The area under a classifier's curve expressed as a fraction of the unit square
 - ✓ Its value ranges from zero to one
- The AUC is useful when a single number is needed to summarize performance
 - ✓ A ROC curve provides more information than its area
 - ✓ Equivalent to the probability that a randomly chosen positive instance will be ranked ahead of a randomly chosen negative instance (Mann-Whitney-Wilcoxon) measure
- ROC graph is NOT the most intuitive visualization for many business stakeholders

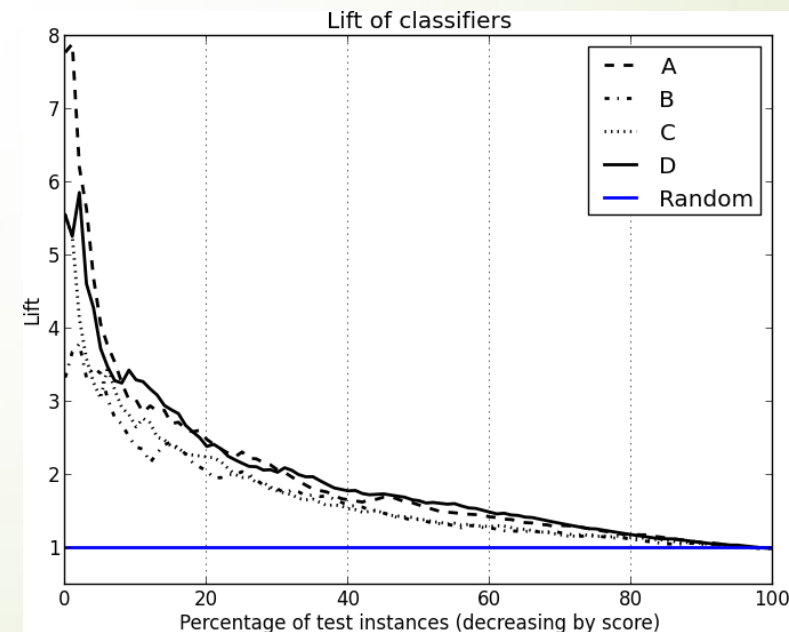
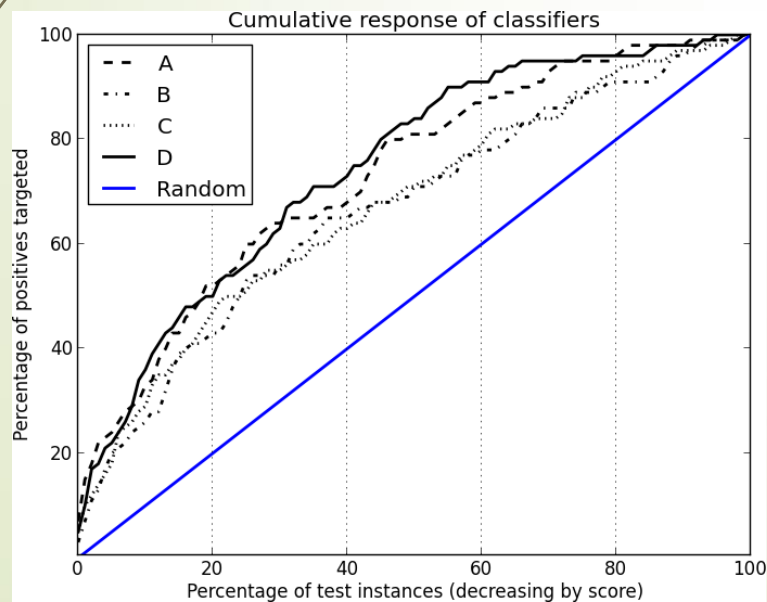
Cumulative Response Curves

- Cumulative response curves plot the **hit rate (tp rate; y axis)**, i.e., the percentage of positives correctly classified, as a function of the percentage of the population that is targeted (x axis).
- ✓ The diagonal line $x=y$ represents random performance



Lift Curves

- The lift curve is essentially the value of the cumulative response curve at a given x point **divided by the diagonal** line ($y=x$) value at that point
- ✓ The diagonal line of a cumulative response curve becomes a horizontal line at $y=1$ on the lift curve.
- ✓ The diagonal line $x=y$ represents random performance



Summary of Evaluation Curves

- Both lift curves and cumulative response curves must be used with care if the **exact proportion of positives** in the population is unknown or is not represented accurately in the test data.
- Unlike for ROC curves, these curves assume that the **test set has exactly the same target class priors** as the population to which the model will be applied.

Example of Churn Prediction

Model	Accuracy
Classification tree	95%
Logistic regression	93%
k-Nearest Neighbor	100%
Naive Bayes	76%

Train Error

Model	Accuracy (%)	AUC
Classification Tree	91.8 ± 0.0	0.614 ± 0.014
Logistic Regression	93.0 ± 0.1	0.574 ± 0.023
k-Nearest Neighbor	93.0 ± 0.0	0.537 ± 0.015
Naive Bayes	76.5 ± 0.6	0.632 ± 0.019

Ten-fold cross-validation

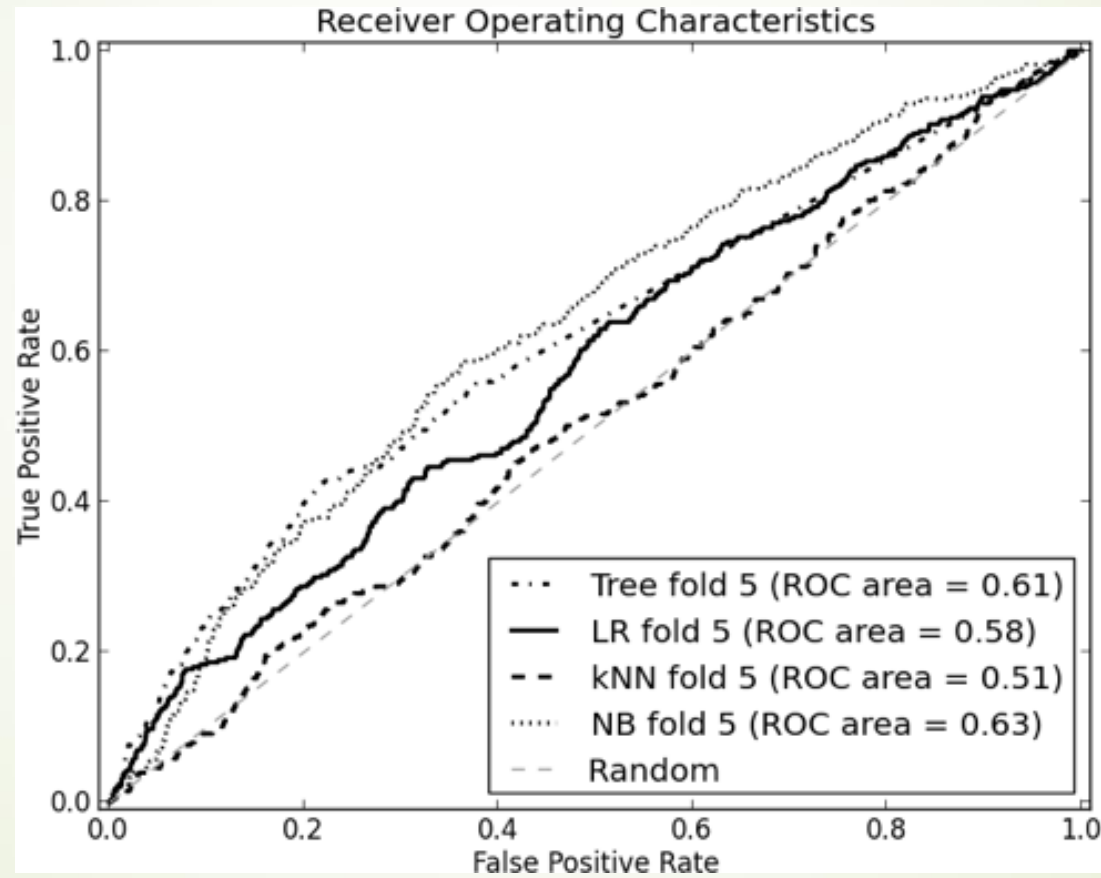
	p	n
Y	127 (3%)	848 (18%)
N	200 (4%)	3518 (75%)

Naïve Bayes—Confusion Matrix

	p	n
Y	3 (0%)	15 (0%)
N	324 (7%)	4351 (93%)

KNN—Confusion Matrix

Example of Churn Prediction as Ranking



Performance Metrics in Sklearn

<https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics>

<code>metrics.accuracy_score(y_true, y_pred[, ...])</code>	Accuracy classification score.
<code>metrics.auc(x, y)</code>	Compute Area Under the Curve (AUC) using the trapezoidal rule
<code>metrics.average_precision_score(y_true, y_score)</code>	Compute average precision (AP) from prediction scores
<code>metrics.balanced_accuracy_score(y_true, y_pred)</code>	Compute the balanced accuracy
⋮	
<code>metrics.precision_recall_curve(y_true, ...)</code>	Compute precision-recall pairs for different probability thresholds
<code>metrics.precision_recall_fscore_support(...)</code>	Compute precision, recall, F-measure and support for each class
<code>metrics.precision_score(y_true, y_pred[, ...])</code>	Compute the precision
<code>metrics.recall_score(y_true, y_pred[, ...])</code>	Compute the recall
<code>metrics.roc_auc_score(y_true, y_score[, ...])</code>	Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.
<code>metrics.roc_curve(y_true, y_score[, ...])</code>	Compute Receiver operating characteristic (ROC)

F1-Score in Sklearn

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

In the **multi-class and multi-label** case, this is the average of the F1 score of each class with weighting depending on the average parameter.

pos_label : str or int, 1 by default

The class to report if `average='binary'` and the data is binary. If the data are multiclass or multilabel, this will be ignored; setting `labels=[pos_label]` and `average != 'binary'` will report scores for that label only.

average : string, [None, 'binary' (default), 'micro', 'macro', 'samples', 'weighted']

This parameter is required for multiclass/multilabel targets. If `None`, the scores for each class are returned. Otherwise, this determines the type of averaging performed on the data:

'binary':

Only report results for the class specified by `pos_label`. This is applicable only if targets (`y_{true,pred}`) are binary.

'micro':

Calculate metrics globally by counting the total true positives, false negatives and false positives.

'macro':

Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

'weighted':

Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance; it can result in an F-score that is not between precision and recall.

'samples':

Calculate metrics for each instance, and find their average (only meaningful for multilabel classification where this differs from `accuracy_score`).

Micro Average vs. Macro-Average

Class 1			Class 2			Micro Average		
	Correct Yes	Correct No		Correct Yes	Correct No		Correct Yes	Correct No
Predicted Yes	10	10	Predicted Yes	90	10	Predicted Yes	100	20
Predicted No	10	970	Predicted No	10	890	Predicted No	20	1860

■ **Macro Averaged Precision :** $((10/(10+10) + (90/(90+10)))/2=0.7$

■ **Micro Averaged Precision :** $100/(20+100)=0.83$

$$\text{Macro - Precision} = \frac{\text{Precision1} + \text{Precision2}}{2}$$

$$\text{Macro - Recall} = \frac{\text{Recall1} + \text{Recall2}}{2}$$

$$\text{Macro - F - Score} = 2 \cdot \frac{\text{Macro - Precision} \cdot \text{Macro - Recall}}{\text{Macro - Precision} + \text{Macro - Recall}}$$

Outline (Summary)

- Accuracy
- Confusion matrix
- Expected value (Profit)
- Profit graph
- ROC curve
- Cumulative response/lift curves