Data Science

12 – Data Analytic Thinking

Asst. Prof. Dr. Santitham Prom-on

Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

Overview

Learning Outcome

- Evaluate model performance
- Identify appropriate baseline and choose a suitable evaluation metrics
- Calculate cost and benefit
- Create baseline method for comparison
- Perform cross-validation

Agenda

- Model evaluation
- Cost/benefit analysis

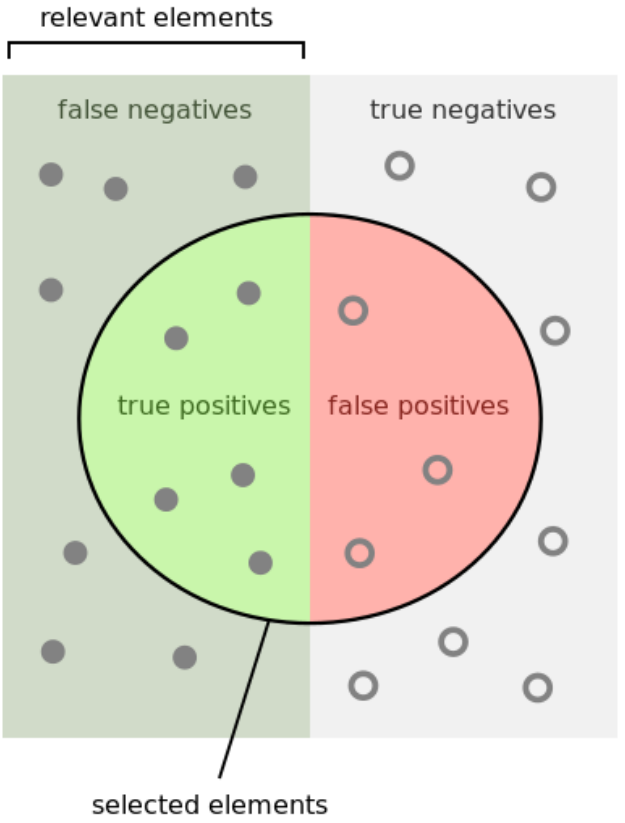
Precision and Recall

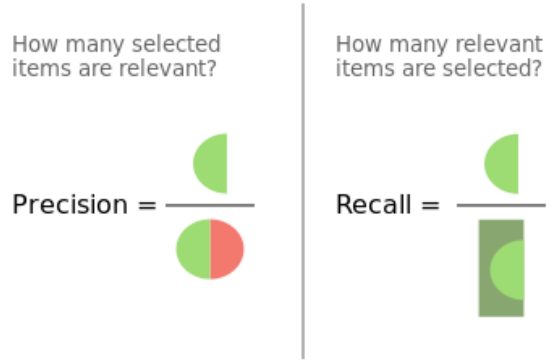
	Actual Positive (p)	Actual Negative (n)
The model says “Yes” = positive (y)	True positives	False positives
The model says “No” = not positive (n)	False negatives	True negatives

- Recall (Completeness) = true positive rate = TP/(TP + FN)
- Precision (Exactness) = the accuracy over the cases predicted to be positive, TP/(TP + FP)
- F-measure = the harmonic mean of precision and recall

  = the balance between recall and precision

  $= 2 \cdot \frac{precision * recall}{precision + recall}$





Baseline

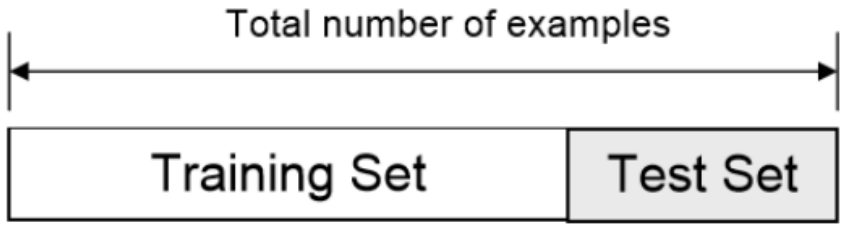
- Data scientists need to know whether they indeed are improving performance. It is then important to know the baseline to compare against.
- The *baseline* values depend on the actual application and coming up with a suitable baselines is one task in the business understanding phase.
- For classification, we may use the ‘majority classifier’, a naïve classifier that always chooses the majority class of the training dataset, as a baseline.
- Chosen baseline should be something the stakeholders find informative

Cross-validation - Why

- One may be tempted to use the entire training data to select the “optimal” classifier, then estimate the error rate
- This naïve approach has two fundamental problems
  - The final model will normally overfit the training data: it will not be able to generalize to new data
    - The problem of overfitting is more pronounced with models that have a large number of parameters
  - The error rate estimate will be overly optimistic (lower than the true error rate)
    - In fact, it is not uncommon to have 100% correct classification on training data

Holdout method

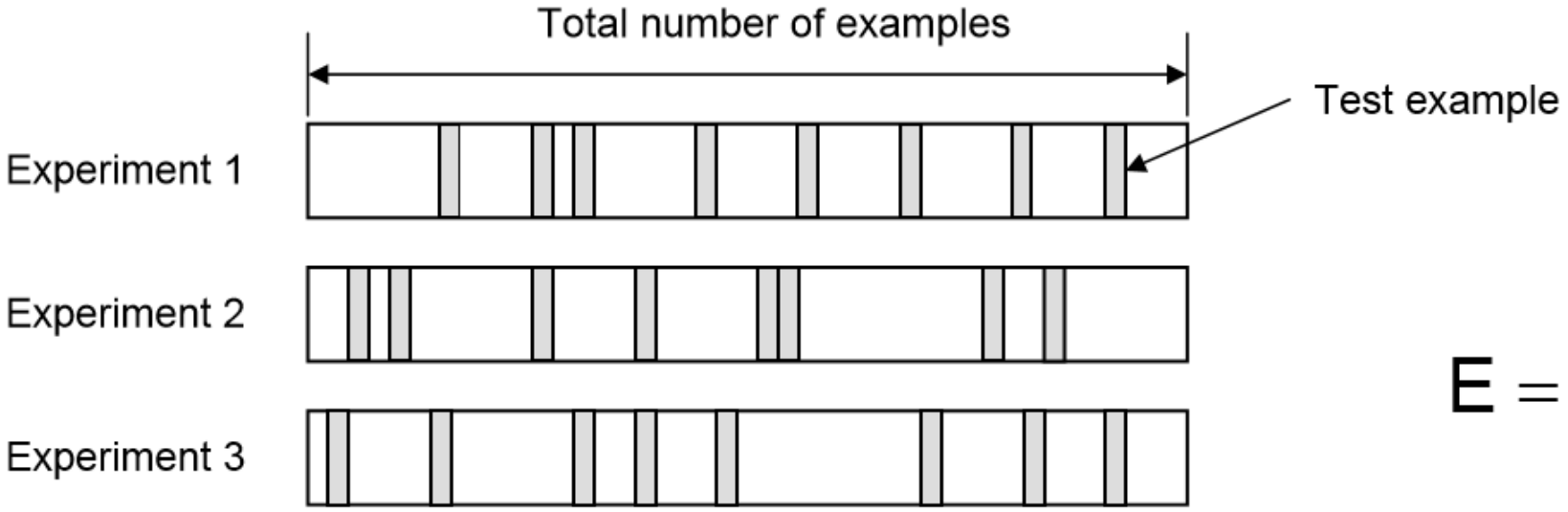
- Split dataset into two groups
  - Training set: used to train the classifier
  - Test set: used to estimate the error rate of the trained classifier



- Drawback
  - In problems where we have a sparse dataset we may not be able to afford the “luxury” of setting aside a portion of the dataset for testing
  - Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an “unfortunate” split

CV: Random subsampling

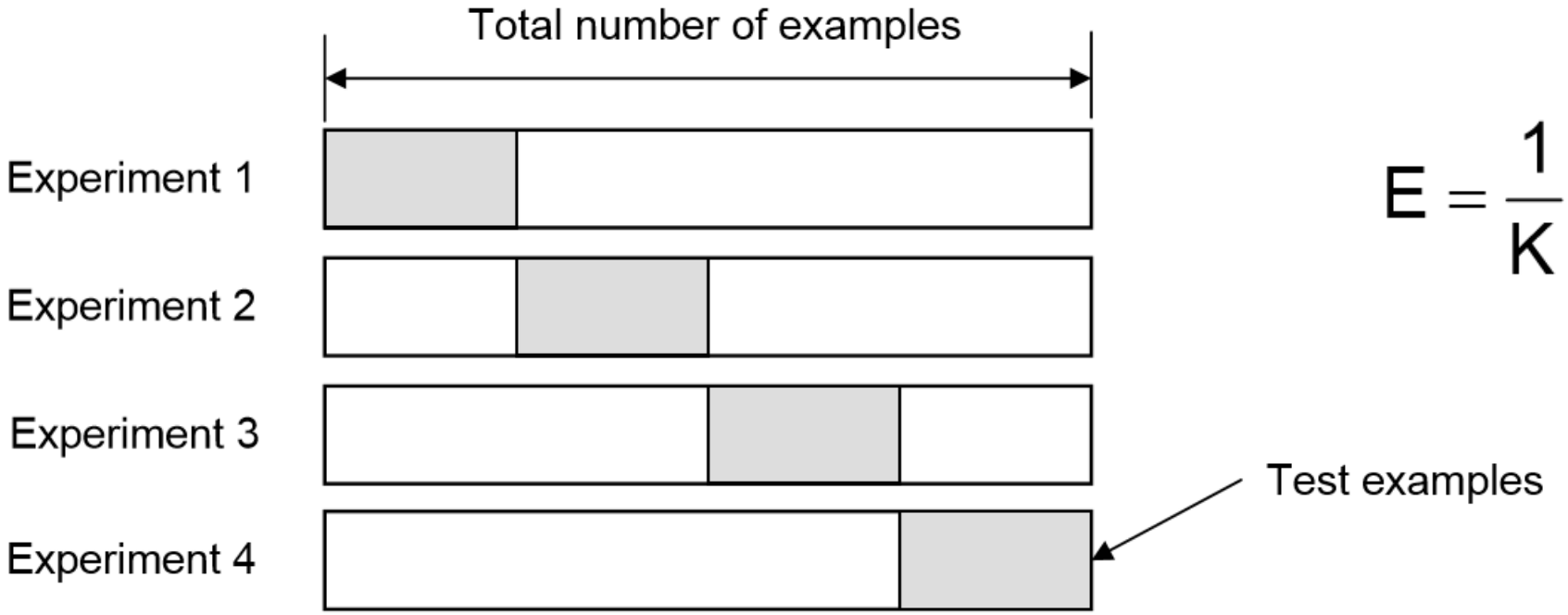
- Performs K data splits of the entire dataset
  - Each data split randomly selects a (fixed) number of examples without replacement
  - For each data split we retrain the classifier from scratch with the training examples and then estimate E_i with the test examples



$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

CV: K-Fold CV

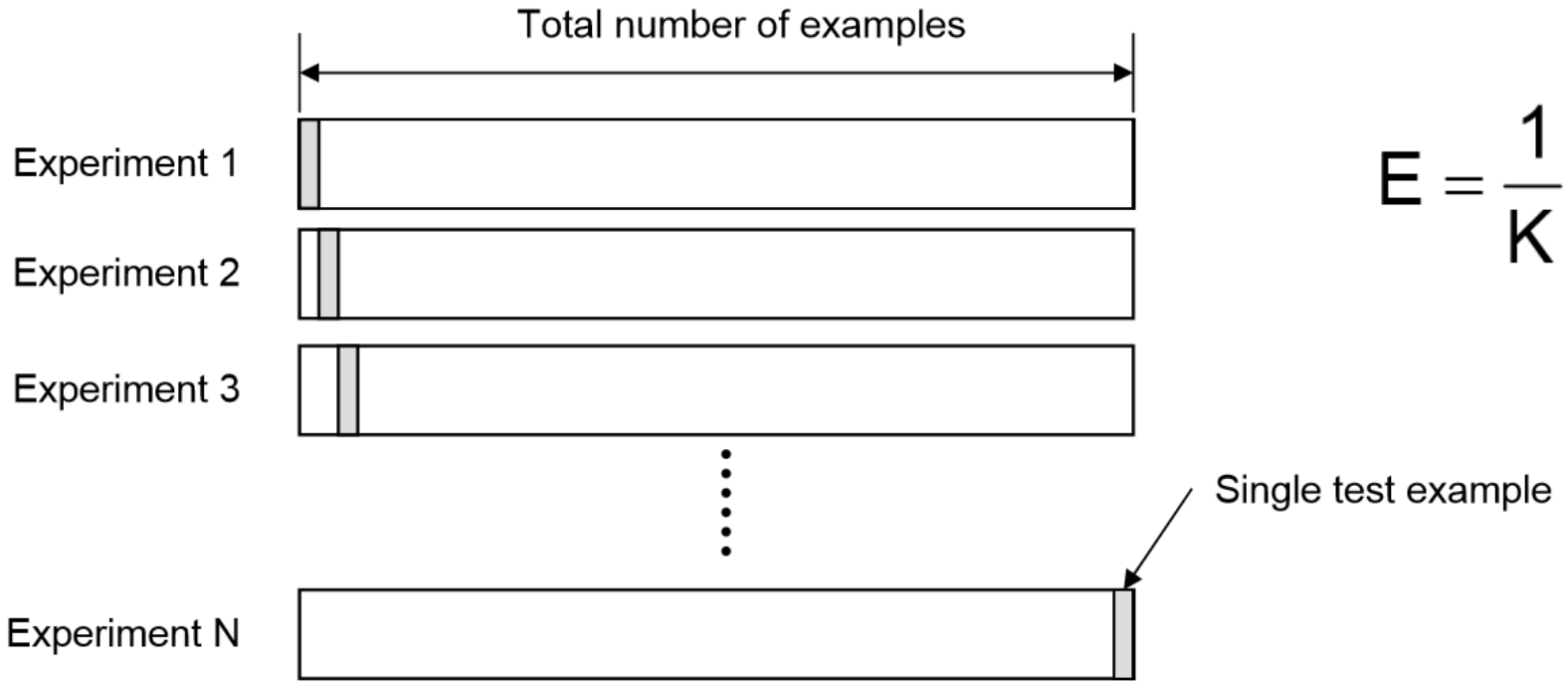
- Create a K-fold partition of the the dataset
  - For each of K experiments, use K-1 folds for training and a different fold for testing



$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

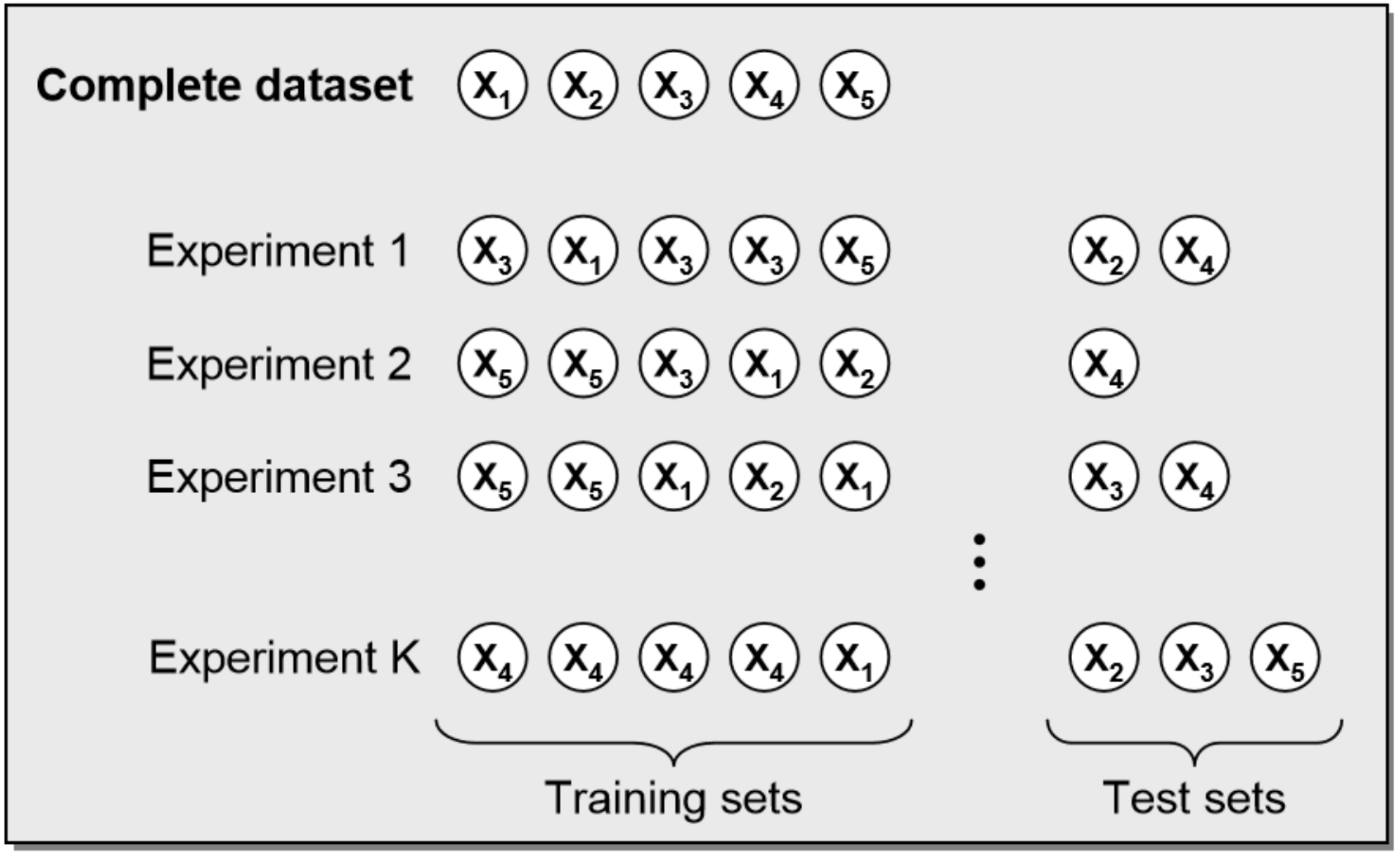
CV: Leave-One-Out CV (LOOCV)

- Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples



$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Bootstrap


Complete dataset
x1 x2 x3 x4 x5
Experiment 1
x3 x1 x3 x3 x5
x2 x4
Experiment 2
x5 x5 x3 x1 x2
x4
Experiment 3
x5 x5 x1 x2 x1
x3 x4
Experiment K
x4 x4 x4 x4 x1
x2 x3 x5
Training sets
Test sets

Evaluating the classifier

- Mainly evaluate using the *error rate* or *classification accuracy*.

$$\text{accuracy} = \frac{\text{Number of correct decision made}}{\text{Total number of decision made}}$$

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

- The accuracy is easy to measure, but sometime not fit to real business problems.
- The accuracy has some well-known problems.
  - The importance of different errors (Unequal Costs and Benefits)
  - Unbalanced class

Confusion Matrix

- Separate out the decisions made by the classifier, making explicit how one class is being confused for another.
- Different errors can then be dealt with separately.
- A 2x2 confusion matrix

	Actual Positive (p)	Actual Negative (n)
The model says "Yes" = positive (y)	True positives	False positives
The model says "No" = not positive (n)	False negatives	True negatives

Counts of the Errors

Counts of the correct decisions

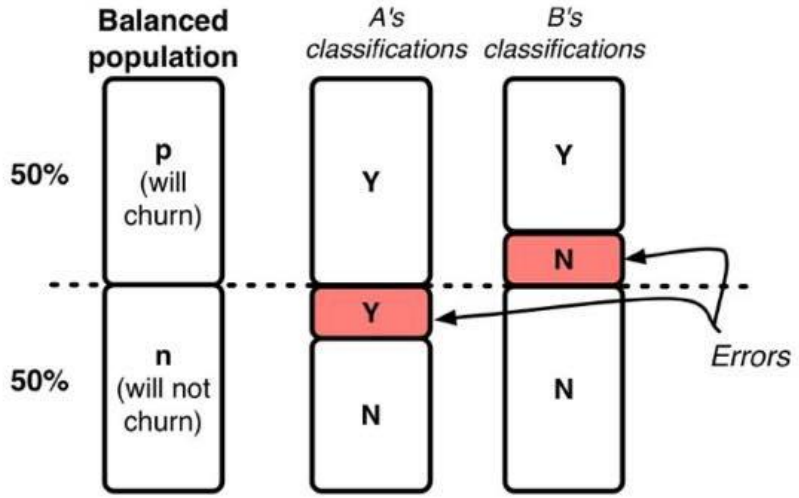
A	Churn	Not churn
Y	500	200
N	0	300

B	Churn	Not churn
Y	500	0
N	200	300

Unbalanced Classes

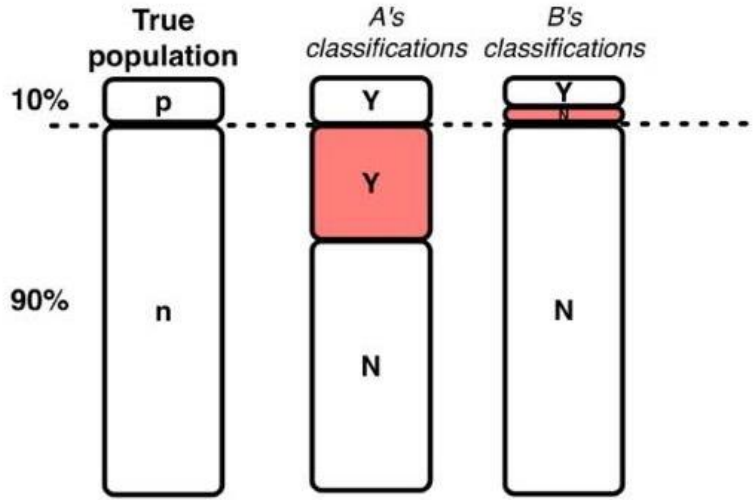
- Classifiers often are used to sift through a large population of normal (uninteresting) entities to find a relatively small number of unusual ones.
  - Checking assembly line for defective parts
  - Targeting customers who actually would respond to an offer.
- Rarity cause the class distribution to be unbalanced (skewed).
- As the distribution becomes more skewed, evaluation based on accuracy break down.
- The accuracy of 99.9% may tell us little about what data mining has really accomplished.
- This example, accuracy is the wrong thing to measure.

Unbalanced Classes (Cont'd)



Artificially balanced datasets for training and testing (50% will churn)

A	Churn	Not churn
Y	500	200
N	0	300



Representative sample from the actual population (only 10% will churn)

B	Churn	Not churn
Y	500	0
N	200	300

Problem with Unequal Costs and Benefits

- Classification accuracy makes no distinction between false positive and false negative errors.
- Counting them together implies that both errors are equally important. In real-world domains, this is rarely the case.
  - False positive: A patient is wrongly informed he has cancer when he does not.
  - False negative: A patient who has cancer but is wrongly told the otherwise.
- Error should be counted separately.
- We should estimate the cost or benefit of each decision a classifier can make.

Data-Analytic thinking

- The general principle of thinking here is beyond just classification. You can apply it to regression as well.
- Regression: A movie recommendation model
  - Predict how much a given customer will like a given movie (Stars).
  - A mean square error (R^2) can be used to report model performance.
  - The real question is, “Why is the mean squared error on the predicted number of stars an appropriate metric for our recommendation problem ? Is it meaningful ? Is there a better matric ?
- So….. the analyst should decompose data-analytic thinking into
  - The structure of the problem
  - The elements of the analysis that can be extracted from the data
  - The elements of the analysis that need to be acquired from other sources (business knowledge or subject matter experts)
  - The expected outcomes

Expected Value

- In expected value (EV) calculation, the possible outcomes of a situation are enumerated.
- EV is the weighted average of the values of the different possible outcomes, where
  - The weight given to each value is its probability of occurrence.
  - $Ev = p(o_1).v(o_1) + p(o_2).v(o_2) + p(o_3).v(o_3) + \ldots\ldots$
  - Each o_i is a possible decision outcome; $P(o_i)$ is its probability and $v(o_i)$ is its value.
  - Probabilities, $P(o_i)$, can be estimated from the data.
  - However, sometime business values, $v(o_i)$, must come from external domain knowledge.

Using EV in classification: Targeted market

- Assign each consumer a class of likely responder versus not likely responder. Then, target the likely responders.
- Assume that customers can **only** buy product by responding to the offer.
- In real life, the probability of response is very low (1-2%). If we choose a “common sense” threshold of 50% to decide the likely responder class, we probably target no one.

Target Market (cont'd)

- Construct a model (From historical data), that gives an estimated probability of response ($P_R(x)$)for any customer.
- Feature vector description x is given as input.
- Decide whether to target a particular consumer described by feature vector x
- Expected benefit of targeting consumer x:
  - $Expected\ benefit = p_R(x).v_{R+}[1 - p_R(x)].v_{NR}$
  - v_R = value we get from a response and v_{NR} is the value we get from no response. The benefit v_R and v_{NR} need to be determined separately.
  - If customer does not respond, the expected benefit is v_{NR} = zero.
  - The probabilities come from the historical data.
  - Assume that a product cost is $200 and our product-related cost is $100 + cost incur from marketing of $1. The value of v_R will be $99.
  - If we mail marketing material and a customer do not respond then v_{NR} becomes $-1

Target Market (cont'd)

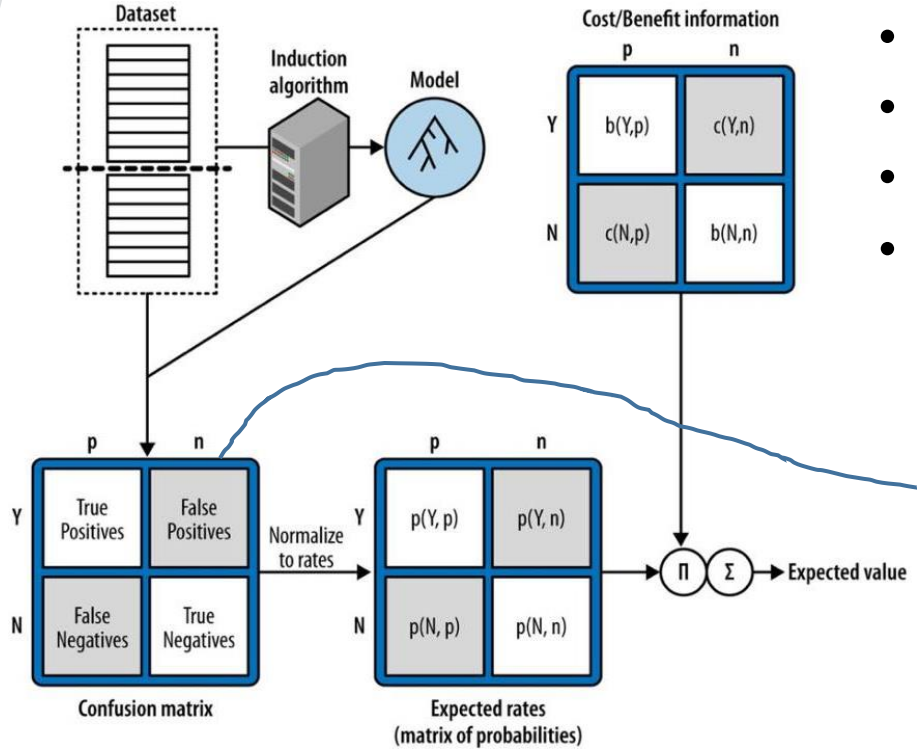
- EV should be great than 0 (EV > 0)
- $p_R(x).99 - [1 - p_R(x)].1 > 0$
- Mathematically, A decision rule can be "Target a given customer x only if, $p_R(x).99 > [1 - p_R(x)].1$
- From the equation, we got $p_R(x) > 0.01$, which means that we should target the consumer as long as the estimated probability of responding is greater than 1%.
- This shows how EV calculation can express how we will use the model.

Comparing Models

- Compare if the data-driven model perform better than the hand-crafted model suggested by the marketing group ?
- Compare whether the decision tree would work better than a linear discriminant model for a particular problem ?

Comparing Models using EV

In aggregation, how well does each model do: What is its expected value?

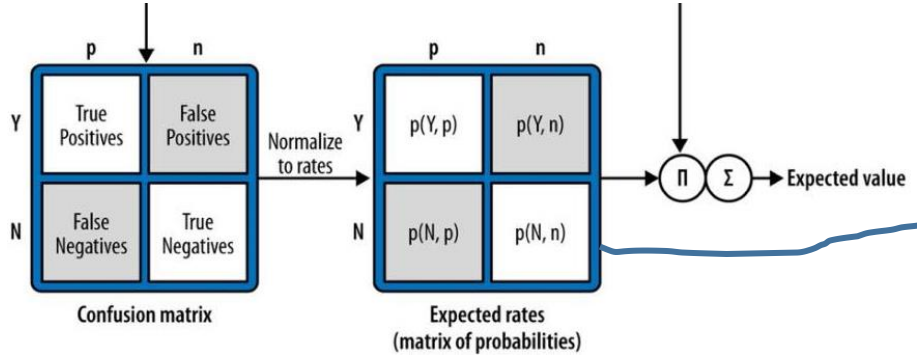


- From the expected value equation:
- $Ev = p(o_1).v(o_1) + p(o_2).v(o_2) + p(o_3).v(o_3) + \ldots$
- Each o_i corresponds to one cell of the confusion matrix.
- Let's consider a concrete example of a classifier confusion matrix

	P	n
Y	56	7
N	5	42

Probability estimation

- Each cell of the confusion matrix contains a count of the number of decisions corresponding to the combination of *predicted* and *actual*, the count will be expressed as *count(h, a)*.
- EV -> reduce the counts to rates or estimated probabilities, *p(h, a)*
- $p(h, a) = count(h, a)/T$; T = the total number of instances.

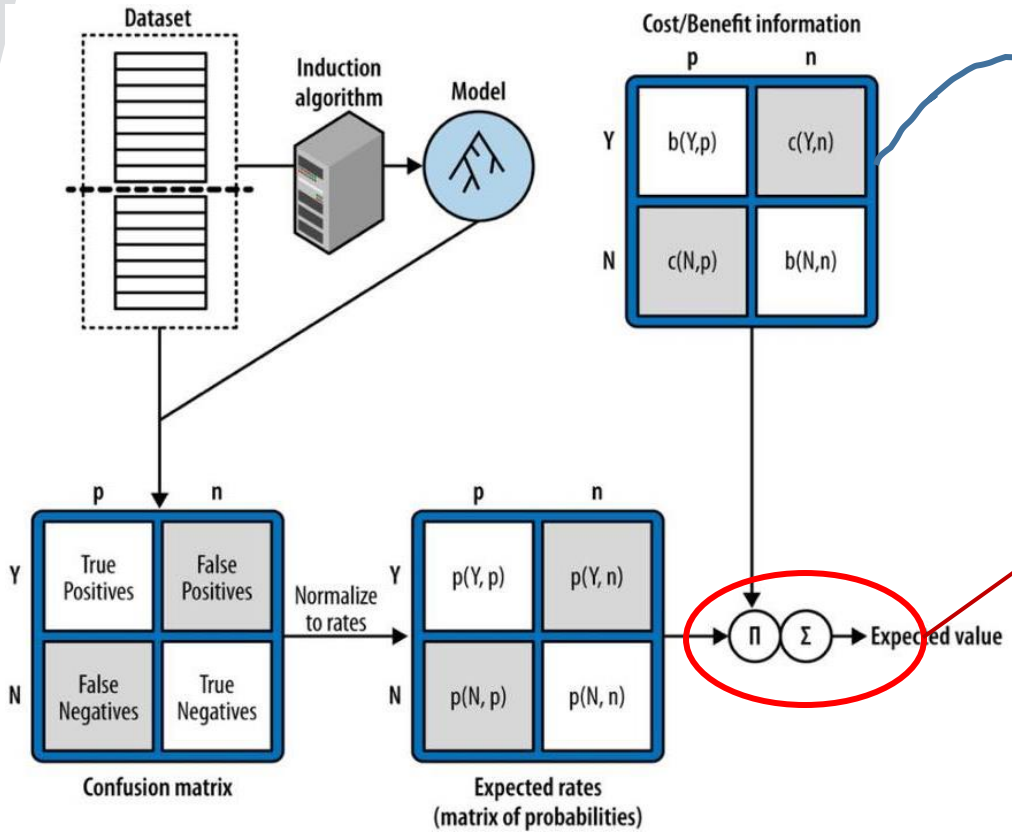


T = 110	
P(Y, p) = 56/110 = 0.51	P(Y, n) = 7/100 = 0.06
P(N, p) = 5/110 = 0.05	P(N, p) = 42/110 = 0.38

Cost/Benefit Values (1)

- For each (predicted, actual) pair, estimate the cost/benefit of making such a decision.
  - Correct classification (true positives and negatives) correspond to the 'benefit' b(Y,p) and b(N, n).
  - Incorrect classification correspond to the 'benefit' b(Y, n) and b(N,p) – negative benefits can be thought of as 'cost' and can be referred to as costs c(Y,n) and c(N,p).
- Let's expressed all numbers in a dollar unit
  - False positive (predict to respond, but did not), the mailing cost is \$1/customer so b(Y,n) = \$-1
  - False negative (predict to not respond and did), no money was spend nothing was gain, b(n,p) = \$0
  - True positive (offers and buy), so b(Y,p) = \$99
  - True negative (not offered and cannot buy), so b(N,n) = \$0

Cost/Benefit Values (2)



	P	n
Y	b(Y,p) = 99	b(Y,n) = -1
N	b(n,p) = 0	b(N,n) = 0

- $EV = p(Y,p).b(Y,p) + p(N,p).b(N,p) + p(N,n).b(N,n) + p(Y,n).b(Y,n)$
- Using this value, we can now compare the expected profits for various models and other targeting strategies.

EV for Unbalanced Population

- Factor our the probabilities of seeing each class (*class priors)*
- Class priors, p(p) and p(n) specify the likelihood of seeing positive and negative instances.
- Factoring these values out allows us to separate the influence of class imbalance from the predictive power.
- From the Basic: $p(x, y) = p(y).p(x|y)$
- We can change the equation into
- $EV = p(Y|p).p(p).b(Y, p) + p(N|p).p(p).b(N, p) + p(N|n).p(n).b(N, n) + p(Y|n).p(n).b(Y, n)$

EV for Unbalanced Population (Cont'd)

- Then factor out the class priors
- $EV = p(p).[p(Y|p).b(Y,p) + p(N|p).b(N,p)] + p(n).[p(N|n).b(N,n) + p(Y|n).b(Y,n)]$
  - The first component corresponds to the expected profit from the positive examples
  - The second component corresponds to the expected profit from the negative examples.
  - Each is weighted by the probability that we see that sort of example.

Example, if the positive examples are very rare,
their contribution to the overall expected profit will be small.

From the previous example

	P	n
Y	56	7
N	5	42

	P	n
Y	b(Y,p) = 99	b(Y,n) = -1
N	b(n,p) = 0	b(N,n) = 0

Total sample (T) = 110	
Positive P = 56+5 = 61	Negative N = 7+42 = 49
P(p) = 61/110 = 0.55	P(n) = 49/110 = 0.45
TP rate = 56/61 = 0.92	FP rate = 7/49 = 0.14
FN rate = 5/61 = 0.08	TN rate = 42/49 = 0.86

Probability of predicting Y when the instance is actually p, p(Y | p) can be estimated by using a true positive rate

- $EV = p(p).[p(Y|p).b(Y,p) + p(N|p).b(N,p)] + p(n).[p(N|n).b(N,n) + p(Y|n).b(Y,n)]$
  - 0.55[0.92*b(Y,p) + 0.08*b(N,p)] + 0.45[0.86*b(N,n) + 0.14*b(Y,n)]
  - 0.55[0.92*99 + 0.08*0] + 0.45.[0.86*0 + 0.14*-1]
  - 50.1 – 0.063
  - 50.04

Summary

- Data scientists must also design a proper evaluation measure for models.
- Simple measures, such as classification accuracy, sometime do not fit with the real-world domain. It rarely captures what is actually important for the problems.
- The expected value calculation is a good framework for organizing the analytic thinking.

BIG DATA
EXPERIENCE

End

Question?

KMUTT