MODEL EVALUATION

盧信銘

Department of Information Management,
National Taiwan University

Overview

- Introduction
- Metrics to evaluate classification, regression, and ranking problems
- Evaluation Methods
- Comparing the Performance of two Classifiers.
- Comparing the Performance of three or more Classifiers.

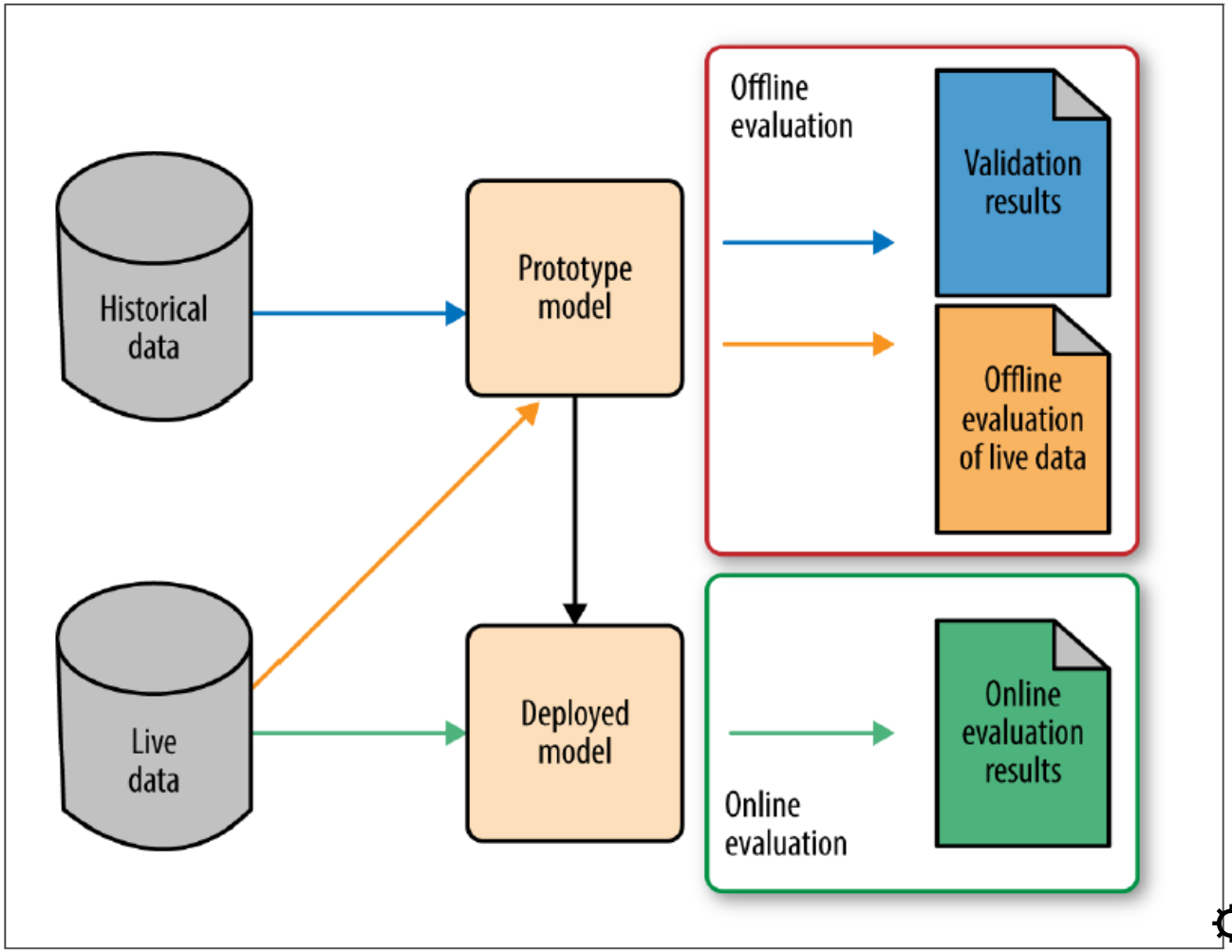
Introduction

- Before attempting any data analytics task, you should ask yourself the following question:
- How can I measure success for this project?

- The question allow us to set realistic goals.
- It also prevent us from working on ill-formulated projects.

Typical Data Analytics Work Flow

- Several stages:
- Prototyping (model selection using historical data)
- Offline evaluation of live data.
- Online evaluation (of deployed model)



(Zhang, 2015)

Performance Evaluation (Offline)

- Using historical data
- Usually adopted mechanisms such as ten-fold cross validation or hold-out testing samples
- Metrics:
  - Classification problem: accuracy, precision, recall, F-measure, area under the curve (AUC)
  - Regression problem: root mean square error (RMSE), mean absolute error (MAE)
  - Ranking problem (think about search engine): normalized discount cumulative gain (NDCG), hitrate@n, mean average precision (MAP).

Reasons for Offline Evaluation

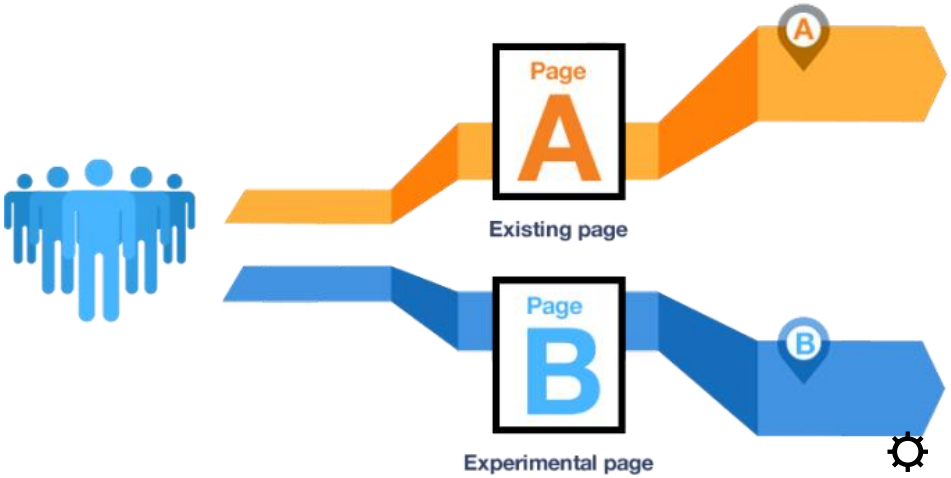
- Determine whether to employ the model;

  (For example: when learning the effectiveness of medical treatments from a limited-size data, it is important to estimate the accuracy of the classifiers.)
- Optimize a model.

  (For example: when post-pruning decision trees we must evaluate the accuracy of the decision trees on each pruning step.)
- Pick the best model.

Performance Evaluation (Online)

- Using live data
- Often adopted mechanisms such as A/B testing
- Metrics:
  - Customer conversion rate.
  - Stickiness.
  - Customer life time value.
  - Also those used in offline stage: NDCG, MAP, RMSE, etc.
- Not the focus of this talk.



How to evaluate the Classifier's Generalization Performance?

- Assume that we test a classifier on some test set and we derive at the end the following *confusion matrix*:

		Predicted class		
		Pos	Neg	
Actual class	Pos	*TP*	*FN*	*P*
	Neg	*FP*	*TN*	*N*

Metrics for Classifier's Evaluation

- Accuracy = $(TP+TN)/(P+N)$
- Error = $(FP+FN)/(P+N)$
- Precision = $TP/(TP+FP)$
- Recall (TP rate) = $TP/P=TP/(TP+FN)$
- FP Rate = $FP/N=FP/(FP+TN)$
- *F-Measure (F1)*= $2\frac{precision \times recall}{precision+recall}$

		Predicted class		
		Pos	Neg	
Actual class	Pos	*TP*	*FN*	*P*
	Neg	*FP*	*TN*	*N*

Example (Confusion Matrix and Performance Metrics)

	Predicted positive	Predicted negative
Labeled positive	80	20
Labeled negative	5	195

- What are the performance metrics?
- Accuracy = $(TP+TN)/(P+N)$=(80+195)/300=91.7%
- Error = $(FP+FN)/(P+N)$=(5+20)/300=8.3%
- Precision = $TP/(TP+FP)$=80/(80+5)=94.1%
- Recall (TP rate) = TP/P=TP/(TP+FN)=80/(80+20)=80%
- FP Rate = $FP/N=FP/(FP+TN)$=5/(5+195)=2.5%
- *F-Measure (F1)*= $2\frac{precision \times recall}{precision+recall}=2\frac{0.8*0.941}{0.8+0.941}=0.865=86.5\%$

Per-Class Accuracy

	Predicted positive	Predicted negative
Labeled positive	80	20
Labeled negative	5	195

- Another way to compute accuracy is to do this class-by-class
- In this example, the accuracy for positive cases is 80/(80+20)=80%
- Accuracy for negative cases is 195/(5+195)=97.5%
- Per-class accuracy= (80%+97.5%)/2 = 88.75
- This method is called macro-averaging.
- The accuracy in the previous slide is call micro-averaging.
- When the positive and negative cases are not balanced, these two accuracy will be different.
- Need to be very careful.

Is this Performance Good Enough?

- Need to compare with other approaches to know whether this level of performance is good.
  - Will discuss later.
- Before doing that, there is a simple approach to give you a rough idea about the performance level.
- Compare with majority class classifier.
- In the previous example, we have 100 positive cases and 200 negative cases.
- The majority classifier simply assign every instance to the majority class (negative class in this example).
- This will give us accuracy=200/300 = 66.7%

Majority Classifier

	Predicted positive	Predicted negative
Labeled positive	0 (TP)	100 (FN)
Labeled negative	0 (FP)	200 (TN)

- Accuracy = $(TP+TN)/(P+N)$=(0+200)/300=66.7% (vs. 91.7%)
- Error = $(FP+FN)/(P+N)$=(100+0)/300=33.3% (vs. 8.3%)
- Precision = $TP/(TP+FP)$=0/(0+0)=NA (vs. 80%)
- Recall (TP rate) = TP/P=TP/(TP+FN)=0/(100)=0% (vs. 91.4%)
- FP Rate = $FP/N=FP/(FP+TN)$=100/(100+200)=33% (vs. 4%)
- *F-Measure (F1)*= $2\frac{precision \times recall}{precision+recall}$=NA (vs. 86.5%)

ROC Curves and Analysis

True	Predicted	
	pos	neg
pos	40	60
neg	30	70

True	Predicted	
	pos	neg
pos	70	30
neg	50	50

True	Predicted	
	pos	neg
pos	60	40
neg	20	80

Classifier 1
TPr = 0.4
FPr = 0.3

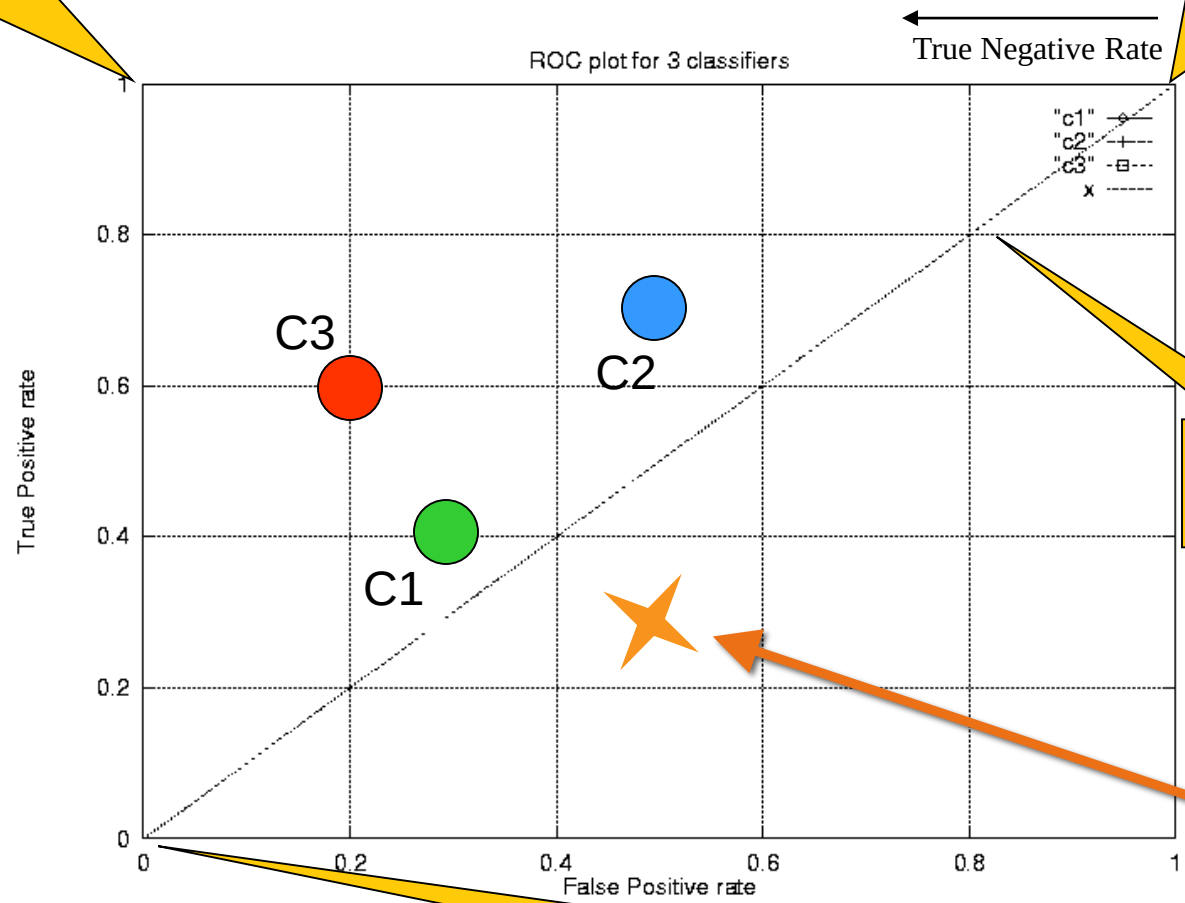
Classifier 2
TPr = 0.7
FPr = 0.5

Classifier 3
TPr = 0.6
FPr = 0.2

Receiver operating characteristic (ROC) Space

Ideal classifier

always positive



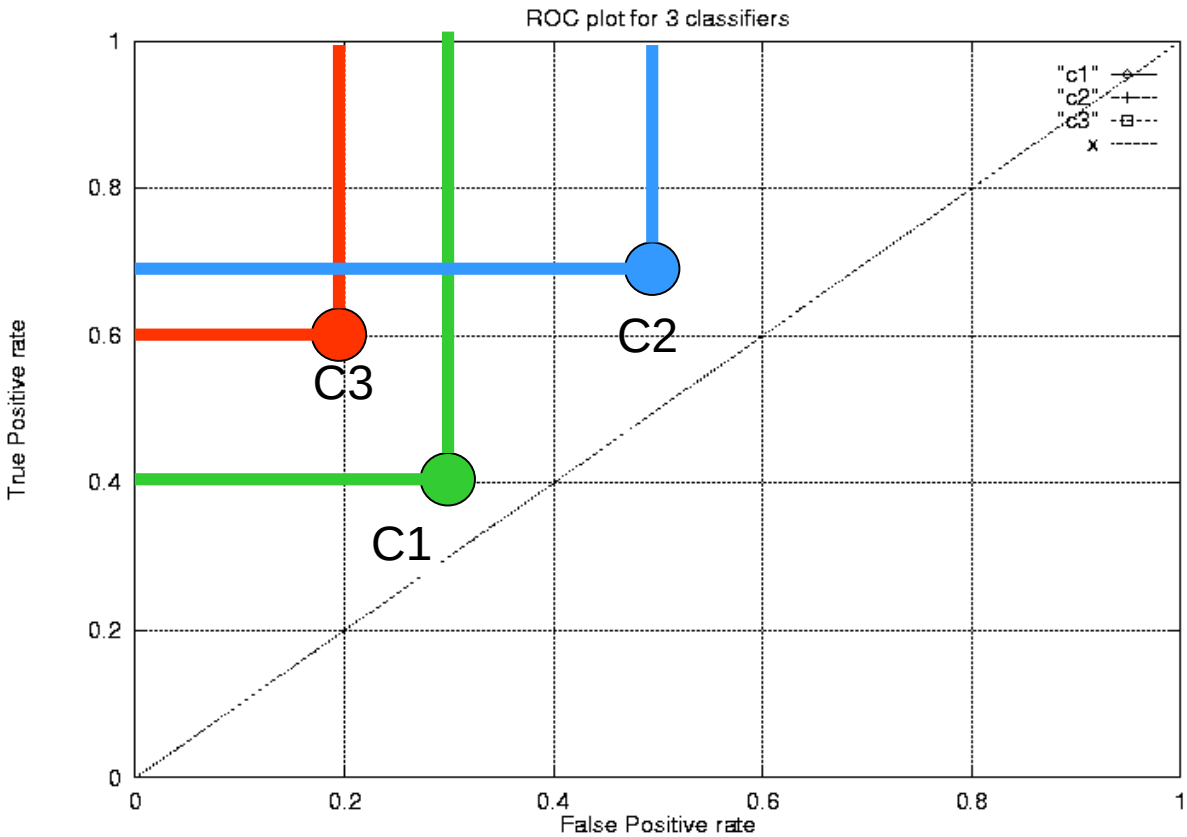
chance

Worse than random; can be improved by switching the prediction

always negative

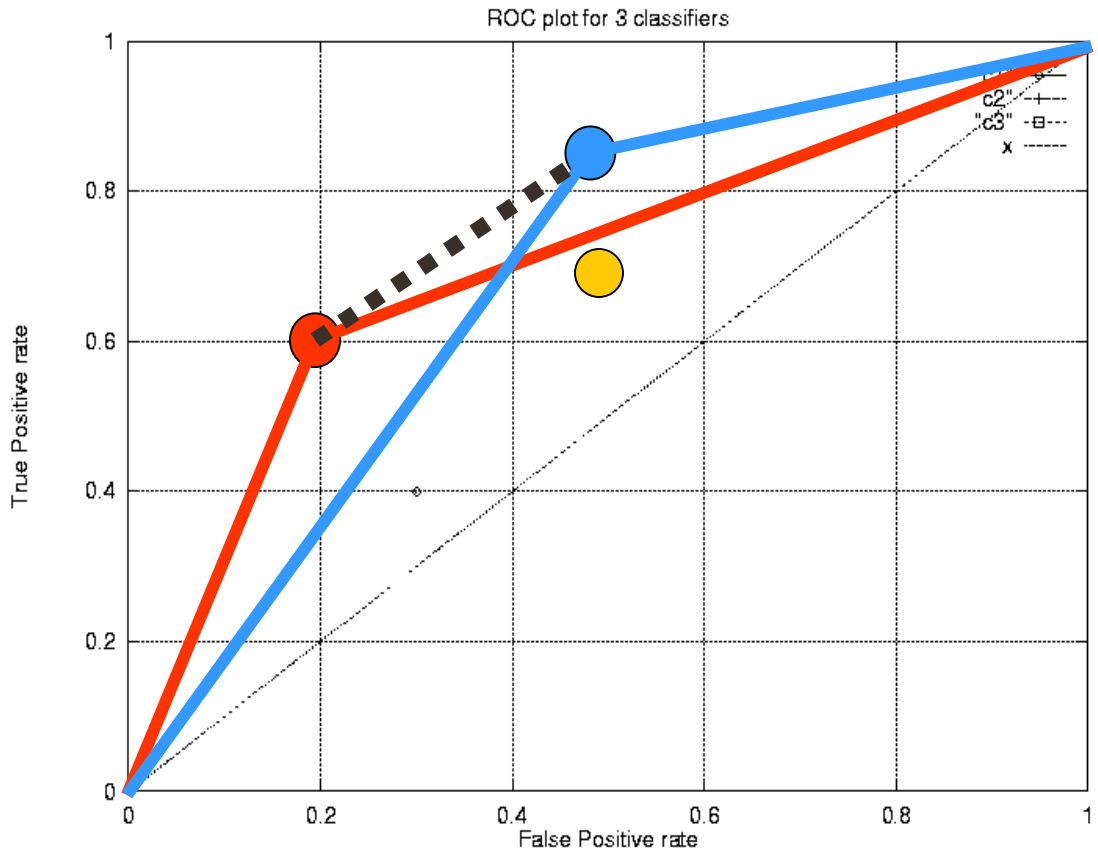
TP rate = *TP/P*
FP Rate = *FP/N*

Dominance in the ROC Space



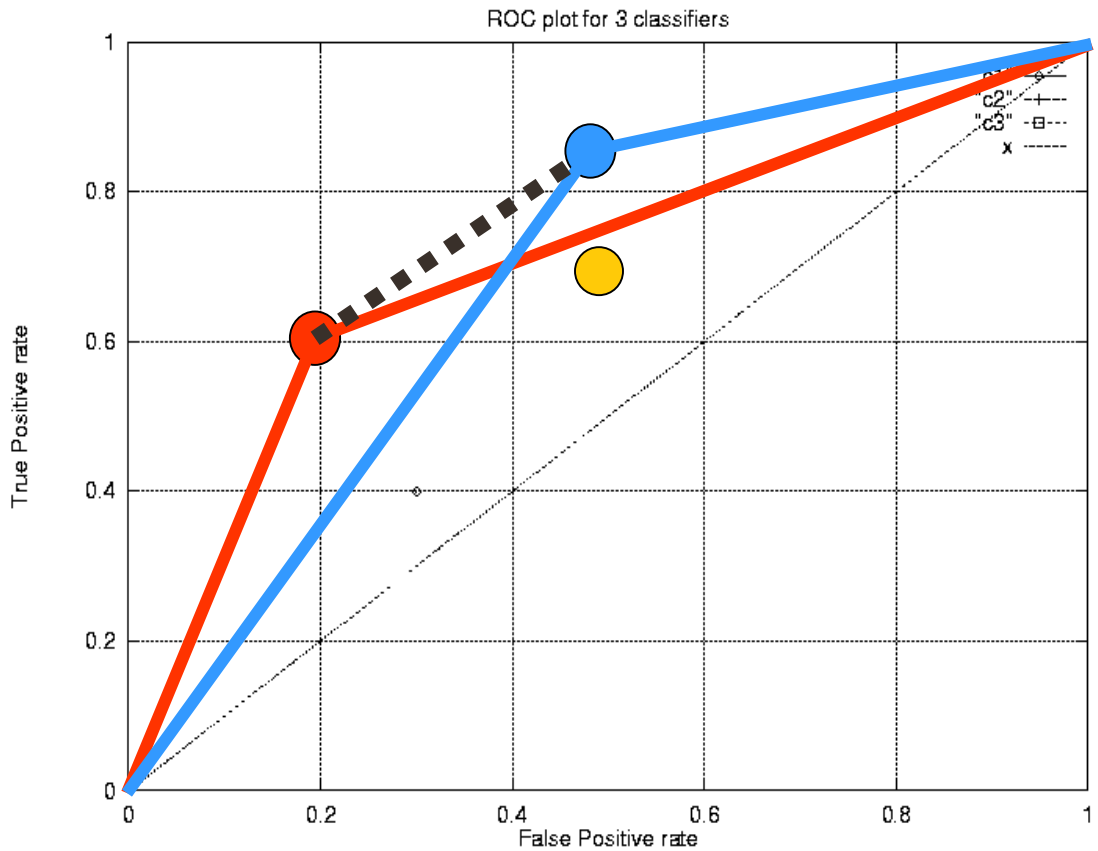
Classifier A dominates classifier B if and only if $TPr_A > TPr_B$ *and* $FPr_A < FPr_B$.

ROC Convex Hull (ROCCH)



- ROCCH is determined by the dominant classifiers;
- Classifiers on ROCCH achieve the best accuracy;
- Classifiers below ROCCH are always sub-optimal.

Convex Hull



- Any performance on a line segment connecting two ROC points can be achieved by randomly choosing between them;
- The classifiers on ROCCH can be combined to form a hybrid.

ROC and AUC

- Receiver operating characteristic curve (ROC) and AUC (area under the ROC curve) are common techniques to analyze classifier performance.
- Consider a trained binary classifier applying to a set of unseen data.
- Instances are sorted according to their predicted probability of being a true positive:

Rank	Predicted Prob.	Actual class
1	0.95	Pos
2	0.93	Pos
3	0.93	Neg
4	0.82	Pos
…	…	…

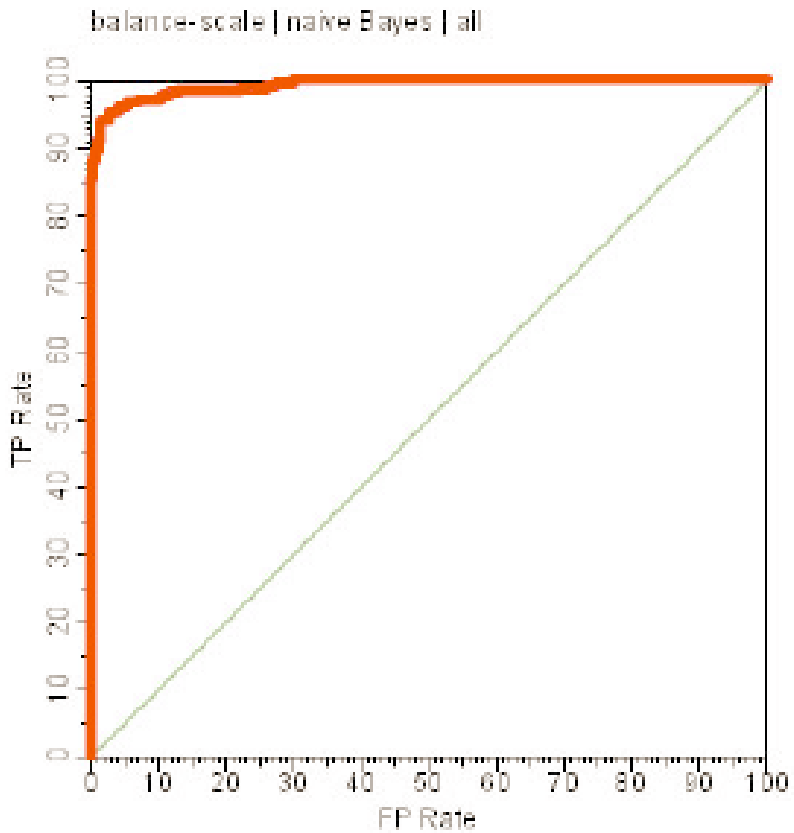
How to Construct ROC Curve for one Classifier

- Sort the instances according to their P_{pos}.
- Move a threshold on the sorted instances.
- For each threshold define a classifier with confusion matrix.
- Plot the TPr and FPr rates of the classfiers.

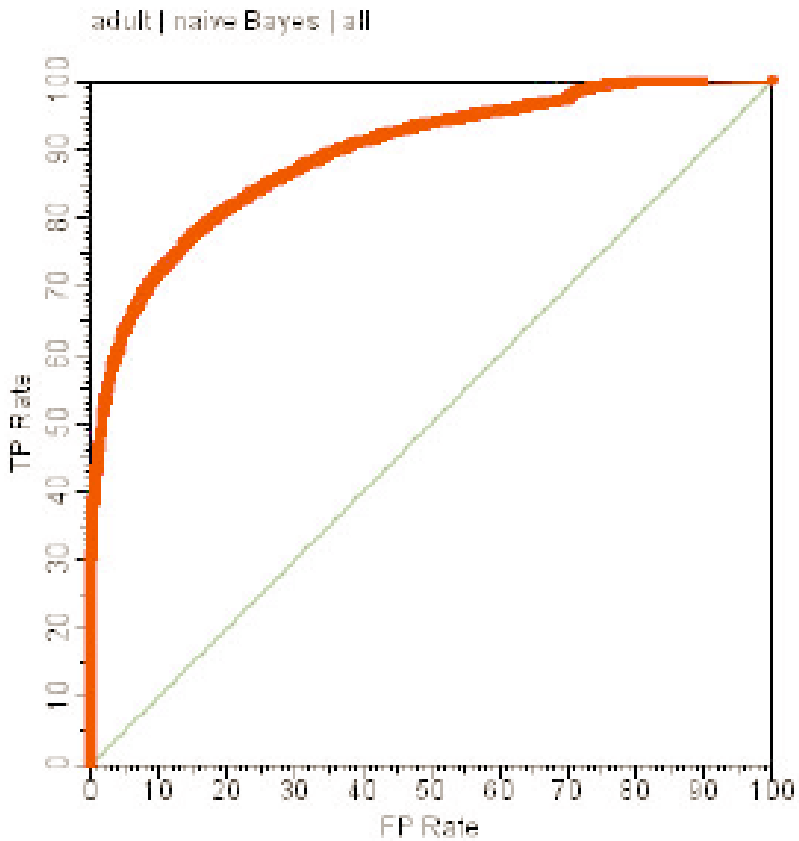
P_{pos}	True Class
0.99	pos
0.98	pos
0.7	neg
0.6	pos
0.43	neg

True	Predicted	
	pos	neg
pos	2	1
neg	1	1

ROC for one Classifier

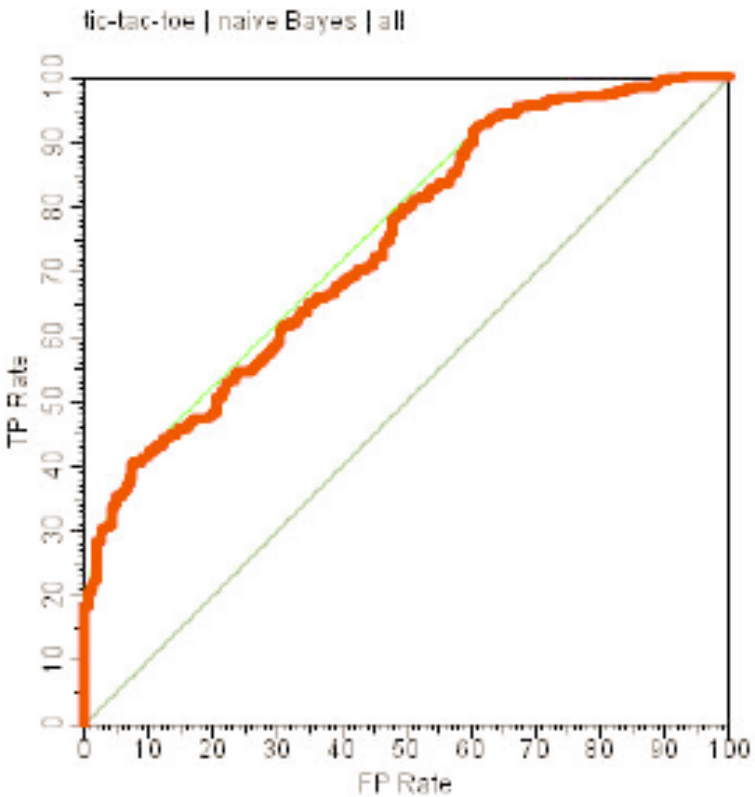


Good separation between the classes, convex curve.

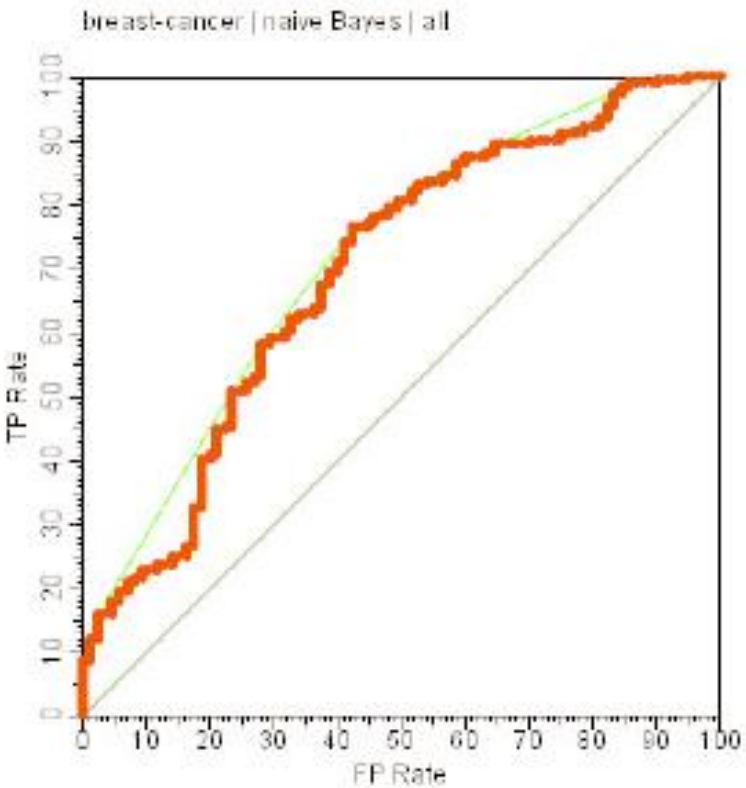


Reasonable separation between the classes, mostly convex.

ROC for one Classifier

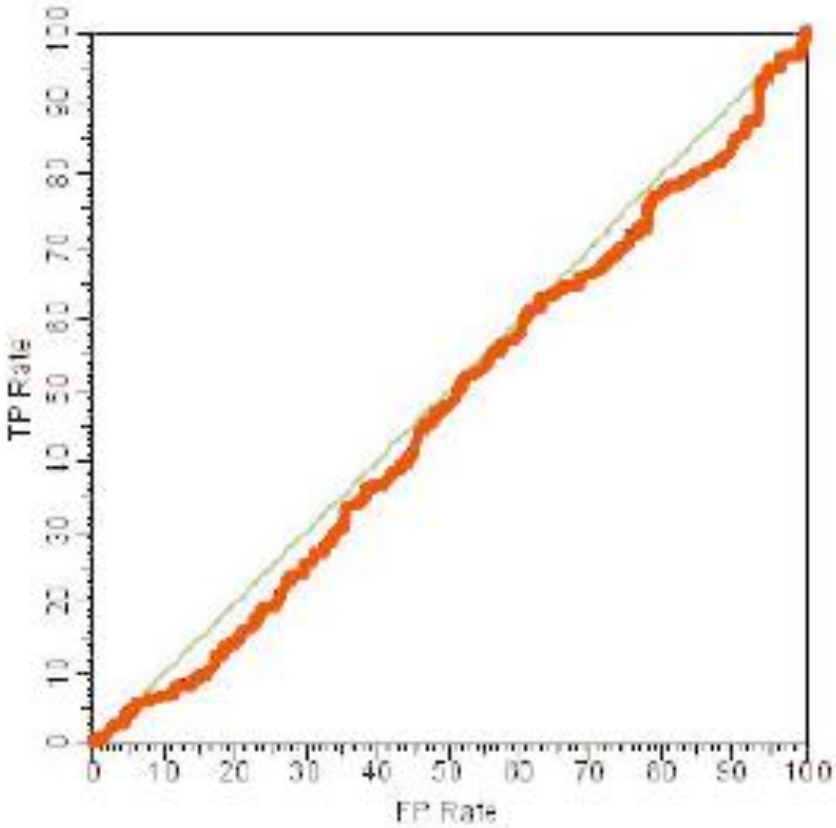


Fairly poor separation between the classes, mostly convex.



Poor separation between the classes, large and small concavities.

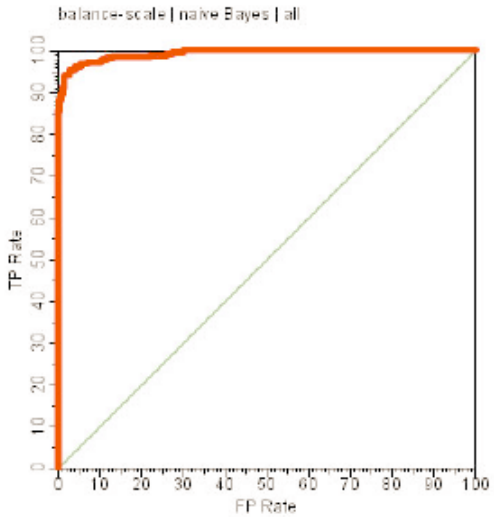
ROC for one Classifier

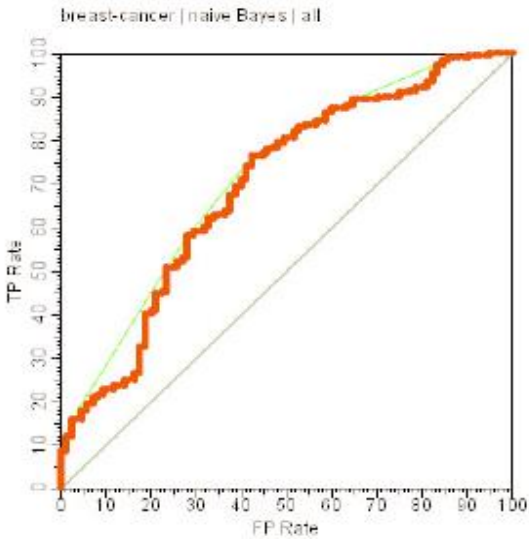


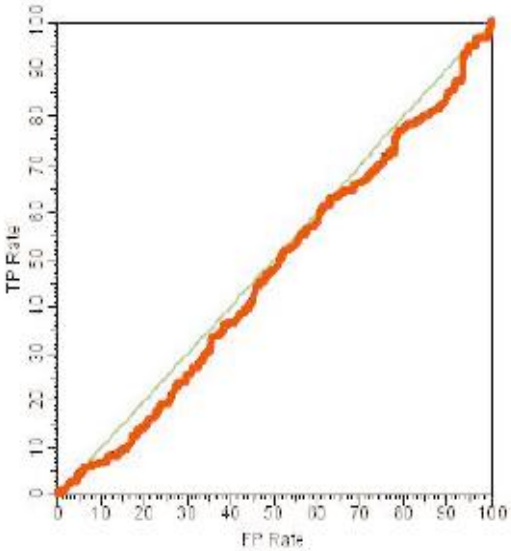
Random performance.

The AUC Metric

- The area under ROC curve (AUC) assesses the ranking in terms of separation of the classes.
- Larger AUC ➔ Better classifier performance
- Best performance: AUC=1
- Random classifier: AUC=0.5







Evaluation of Regression Model

- Root Mean Square Error (RMSE):
- $RMSE = \sqrt{\frac{\sum(Actual - Forecast)^2}{n}} = \sqrt{MSE}$
- Standard error is RMSE.
- Mean Absolute Error (MAE):
- $MAE = \frac{\sum|Actual - Forecast|}{n}$
- Mean Absolute Percentage Error (MAPE):
- $MAPE = \frac{1}{n}\left[\sum\left|\frac{Actual - Foreccast}{Actual}\right|\right]$
- In general, the lower the error measure (RMSE, MAE, MAPE), the better the forecasting model

At a high level,
what is the difference between using RMSE and MAE?

Ranking Performance

- Ranking problem: Consider the situation of developing a recommender system. For each user, we have a list of recommended items.
- This is a ranked list.
- Meaning: if space are limited, we will only show the top N items.
- How good is this the recommendations?

Hitrate@n

- Hitrate@n (n=1, 2, 3, 5)
  - $Hitrate@n = N_c/N$
  - The percentage of testing records that were hit at least once among the top n predictions.

- Example: recommend books given historical purchasing data.
  - Actual purchase (user 1): m1 (rank 1), m5 (rank 2), m3 (rank 10)
  - Actual purchase (user 2): m88 (rank 2) m10 (rank 3) m7 (rank 5)
- Hitrate@1: ½ = 0.5
- Hitrate@3: 2/2 = 1

Discounted Cumulative gain (DCG)

- Discounted cumulative gain (DCG):
- We have higher gain if items ranked at top are more relevant (e.g. been selected or purchased).
- Discount items at lower positions.
- Typical discount rate is 1/log(rank).
- $DCG = rel_1 + \sum_{i=2}^{n} rel_i / \log_2 i$
- rel_i is relevance score of item ranked at position i:
  - $rel_i = (1,0)$ if the outcome is to be selected or not.
  - $rel_i = (2,1,0)$ if a human judge rates the items.
  - Other scoring schema are possible.

Discounted cumulative gain (DCG)

- To compute DCG at position n (using log base 2):
- $DCG = rel_1 + rel_2 + \frac{rel_3}{\log 3} + \frac{\mathrm{rel}_4}{\log 4} + \cdots + \frac{\mathrm{rel}_\mathrm{n}}{\log n}$
- Example:
- 10 ranked documents judged on 0-3 relevance scale: 3, 2, 3, 0, 0, 1, 2, 2, 3, 0
- Discounted gain:
- 3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0
- = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0
- DCG@n:
- 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

Normalized Discounted cumulative gain (NDCG)

- Ideal Discounted Cumulative Gain (IDCG): What would be the perfect score?
- The search engine returned the most relevant page at top, then less relevant page, etc.
- The all top recommendations are purchased.
- The score corresponding to the ideal situation is IDCG.

- Normalized discounted cumulative gain (NDCG)
  - NDCG = $\frac{DCG}{IDCG}$, $DCG = rel_1 + \sum_{i=2}^{n} rel_i / \log_2 i$
  - The discounted score compared to the best possible one.
  - Between 0 and 1.

NDCG Example

- Example: recommend books given historical purchasing data.
  - Actual purchase (user 1): m1 (rank 1), m5 (rank 2), m3 (rank 10)
  - Actual purchase (user 2): m88 (rank 2) m10 (rank 3) m7 (rank 5)
- DCG@10 for record 1: $1 + 1/\log_2 2 + 1/\log_2 10 = 2.30$
- IDCG@10 for record 1: $1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$
- NDCG@10 for record 1 = DCG/IDCG = 2.30/2.63=0.87
  - Need to average across all records to obtain NDCG of an experiment

- DCG@5 for record 1: $1 + 1/\log_2 2 = 2$
- IDCG@5 for record 1: $1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$
- NDCG@5 for record 1: DCG/IDCG = 2/2.63=0.76

Precision@K

- Compute the precision at a given position K over a ranked output.
- Ignore results ranked lower than K.
- For example, using O and X denote whether a recommended item is purchased:
- Position: 1 2 3 4 5 6 7 8 9 10
- Result: O X O X X X X X O X
- Precision@1=1/1=100%
- Precision@3=2/3=66%
- Precision@5=2/5=40%
- Precision@10=3/10=30%

Mean Average Precision (MAP)

- Consider the rank position of all relevant document (or all purchased items)
- Namely: k1, k2, k3, …, kR
- Compute precision@K for each k1, k2, k3, …, kR.
- Average all these precision values.
  - Average precision: $AP(a) = \frac{1}{m_a}\sum_{k=1}^{m_a} precision(rank_k)$
- MAP is the average precision of all users (or queries)
  - $MAP = \frac{1}{|T|}\sum_{a \in T} AP(a)$

Mean Average Precision (MAP)

- Example: recommend books given historical purchasing data.
  - Actual purchase (user 1): m1 (rank 1), m5 (rank 2), m3 (rank 10)
  - Actual purchase (user 2): m88 (rank 2) m10 (rank 3) m7 (rank 5)

- Average precision for record 1: (1/1 + 2/2 + 3/10)/3=0.76
- Average precision for record 2: (1/2 + 2/3 + 3/5)/3 = 0.588
- Mean average precision (rec 1 and 2) = (0.76 + 0.588)/2 = 0.647

Perplexity

- Perplexity (exponential of negative averaged log likelihood)
  - $\exp\left(-Log(W_{test})/C_{W_{test}}\right)$

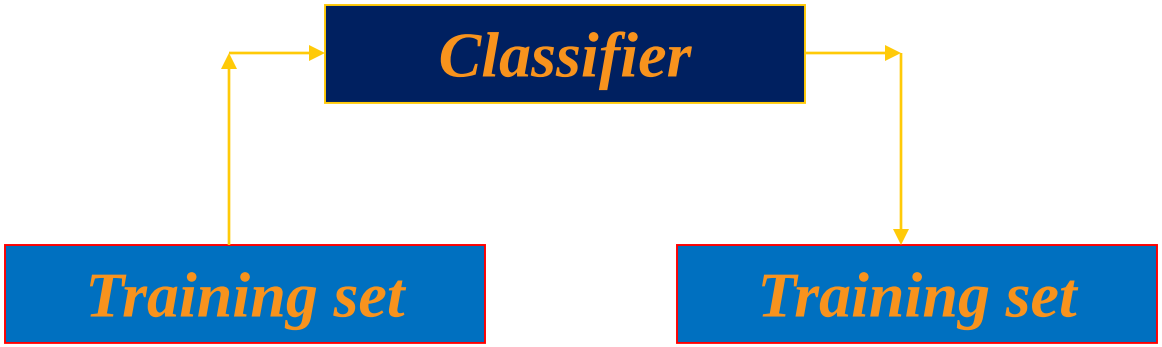
- Task: predicting book purchases given historical records
  - User 1: m1 (0.4), m5 (0.15), m3 (0.03)
  - User 2: m5 (0.3) m10 (0.1) m7 (0.01)
- Perplexity:
- $\exp(-[\log 0.4 + \log 0.15 + \log 0.03 + \log 0.3 + \log 0.1 +$

How to Estimate the Metrics?

- We can use:
  - Training data;
  - Independent test data;
  - Hold-out method;
  - *k*-fold cross-validation method;
  - Leave-one-out method;
  - Bootstrap method (omitted);
  - And many more…

Estimation with Training Data

- The accuracy/error estimates on the training data are *not* good indicators of performance on future data.



  - **Q: Why?**
  - **A:** Because new data will probably not be **exactly** the same as the training data!
- The accuracy/error estimates on the training data measure the degree of classifier's overfitting.

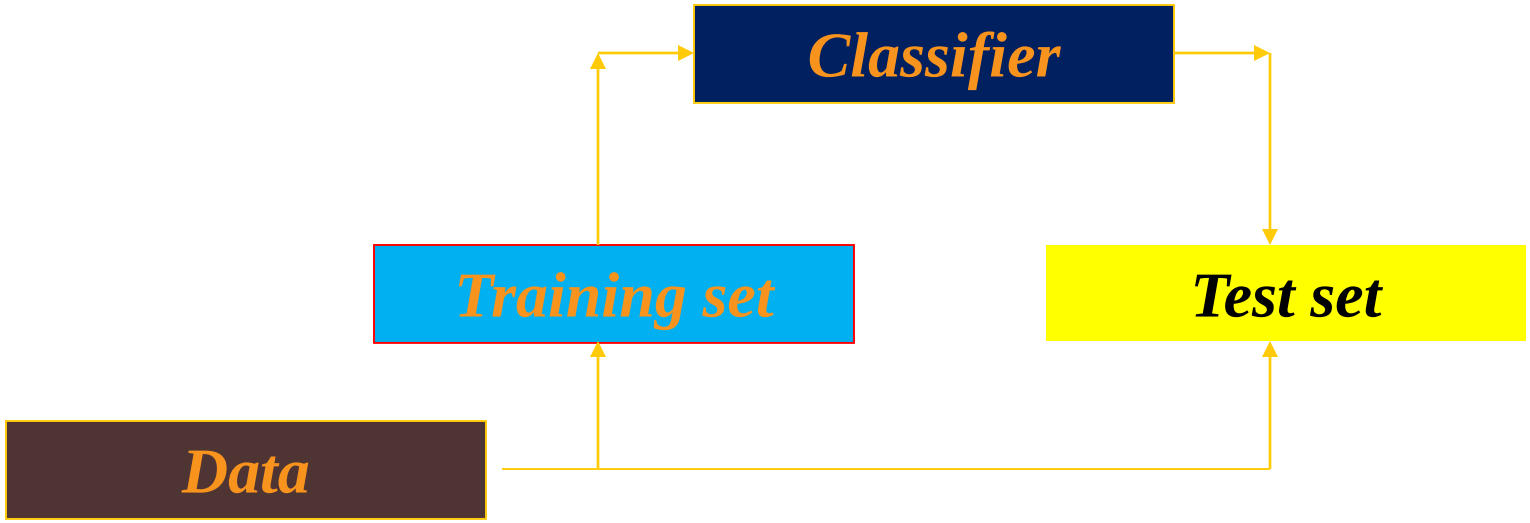
Estimation with Independent Test Data

- Estimation with independent test data is used when we have plenty of data and there is a natural way to forming training and test data.

- *For example: Quinlan in 1987 reported experiments in a medical domain for which the classifiers were trained on data from 1985 and tested on data from 1986.*

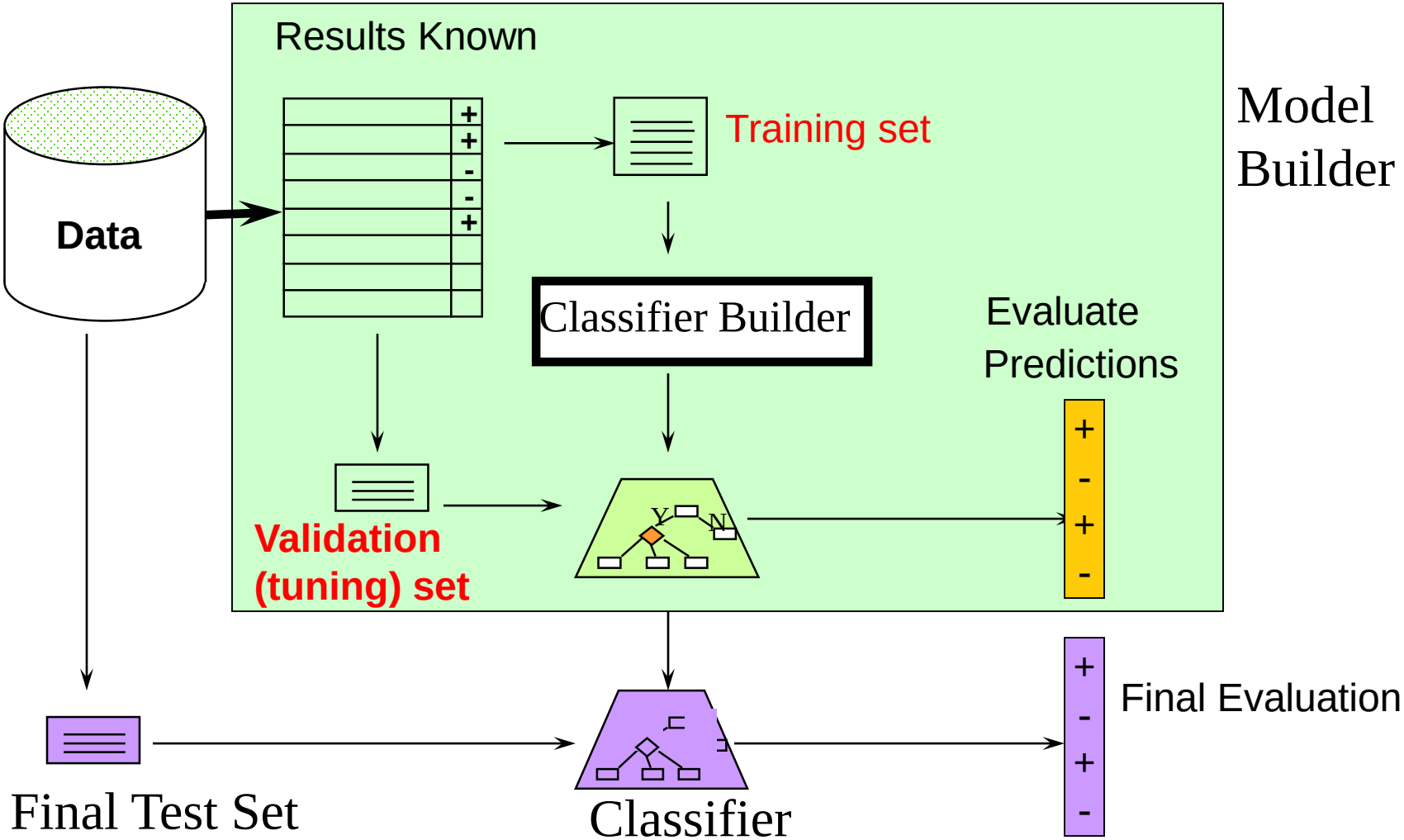
Hold-out Method

- The hold-out method splits the data into training data and test data (usually 2/3 for train, 1/3 for test). Then we build a classifier using the train data and test it using the test data.



- The hold-out method is usually used when we have thousands of instances, including several hundred instances from each class.

Classification: Train, Validation, Test Split



The test data can't be used for parameter tuning!

Making the Most of the Data

- Once evaluation is complete, *all the data* can be used to build the final classifier.
- The final classifier may be for production system.
- Generally, the larger the training data the better the classifier (but returns diminish).
- The larger the test data the more accurate the error estimate.

Stratification

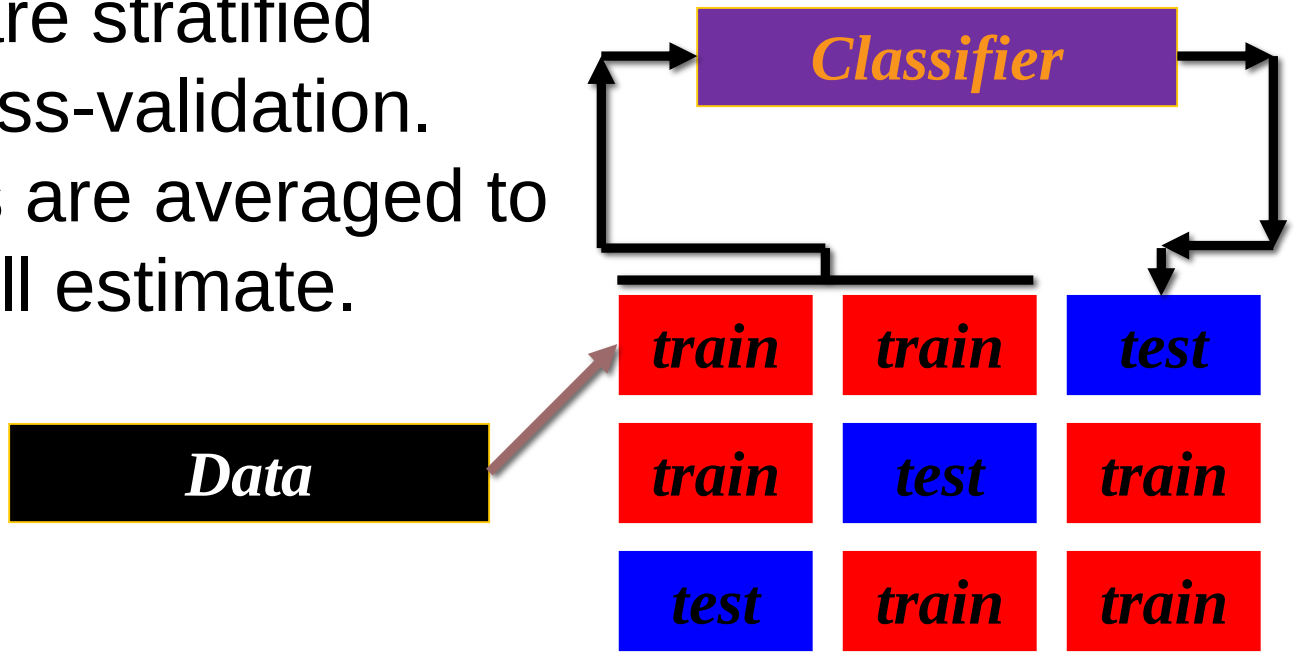
- The *holdout* method reserves a certain amount for testing and uses the remainder for training.
  - *Usually: one third for testing, the rest for training.*
- For “unbalanced” datasets, a sample may contain on minority class.
- To address this problem, we can perform stratification
- Procedure:
- *For each class:*
  - *Random sample α% for training and the remaining for testing.*
- *Combine all training and testing samples from all class to form the final training and testing dataset.*

Repeated Holdout Method

- Holdout estimate can be made more reliable by repeating the process with different subsamples.
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratification).
  - The error rates on the different iterations are averaged to yield an overall error rate.
- This is called the *repeated holdout* method.

k-Fold Cross-Validation

- *k-fold cross-validation* avoids overlapping test sets:
  - *First step*: data is split into *k* subsets of equal size;
  - *Second step*: each subset in turn is used for testing and the remainder for training.
- The subsets are stratified before the cross-validation.
- The estimates are averaged to yield an overall estimate.



More on Cross-Validation

- Standard method for evaluation: stratified 10-fold cross-validation.
- Why 10? Extensive experiments have shown that this is the best choice to get an accurate estimate.
- Stratification reduces the estimate's variance.
- Even better: repeated stratified cross-validation:
  - E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance).

Leave-One-Out Cross-Validation

- Leave-One-Out is a particular form of cross-validation:
  - Set number of folds to number of training instances;
  - I.e., for *n* training instances, build classifier *n* times.
- Makes best use of the data.
- Involves no random sub-sampling.
- Very computationally expensive.
- May be problematic in some situation. (see next slide)

Leave-One-Out Cross-Validation and Stratification

- Disadvantage: stratification not possible:
  - It *guarantees* a non-stratified sample because there is only one instance in the test set!
- Extreme example - random dataset split equally into two classes:
  - Consider a simple classifier that predicts majority class in the training;
    - 50% accuracy on fresh data;
    - Leave-One-Out-CV estimate is 100% error!

Confidence Interval and Hypothesis Testing on Performance Metrics

- Need to know “how reliable” a performance metrics is.
- Meaning (roughly, not very precise): what will the result become if we redo the experiment using another dataset sampled from the same population?
- Meaning (in a more “statistically correct” sense): What is the width of the confidence interval of the performance metric?
- Recall: A 95% confidence interval is an interval that can cover the true parameter (e.g. the true accuracy) 95% of the time.

Two Scenarios to Discuss (Single Classifier)

- We have discuss several mechanisms to estimate performance:
  - Hold-out Method (Scenario 1)
  - (Repeated) Cross-Validation (Scenario 2)
  - Repeated Hold-out Method (Scenario 2)

- Need to use different statistical procedures to estimate confidence interval or conduct hypothesis testing.

Confidence Interval for Hold-out Methods

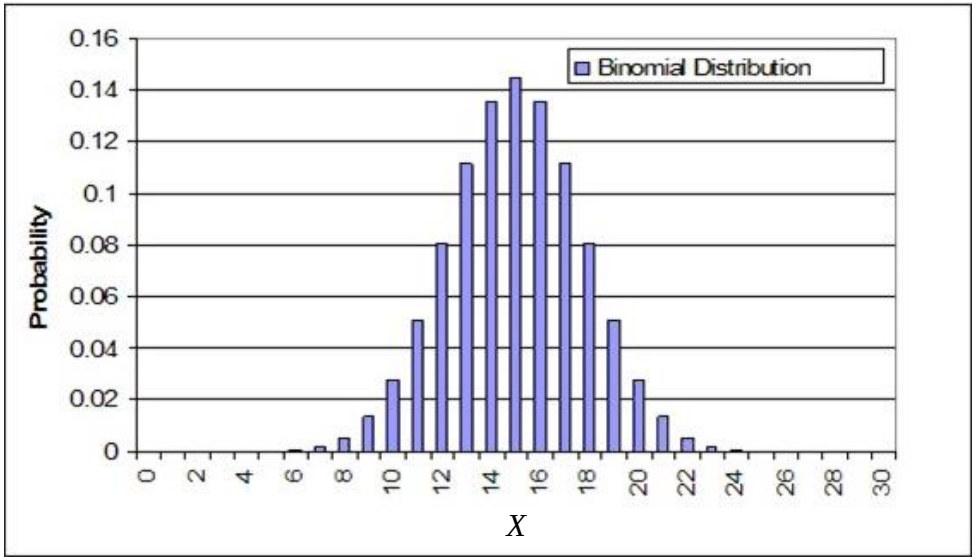
- Hold-out Method: A convenient approach to use.
- In some situation, hold-out method is the only choice.
- For example, to evaluate a rule-based system that are constructed manually from inspecting a training dataset, we usually use hold-out method.
- Reason: very time-consuming to manually construct rules.

Confidence Interval for Hold-out Methods

- Assume that the estimated accuracy $acc_S(h)$ of classifier h is 75%.
- How close is the estimated accuracy $acc_S(h)$ to the true accuracy $acc_D(h)$?

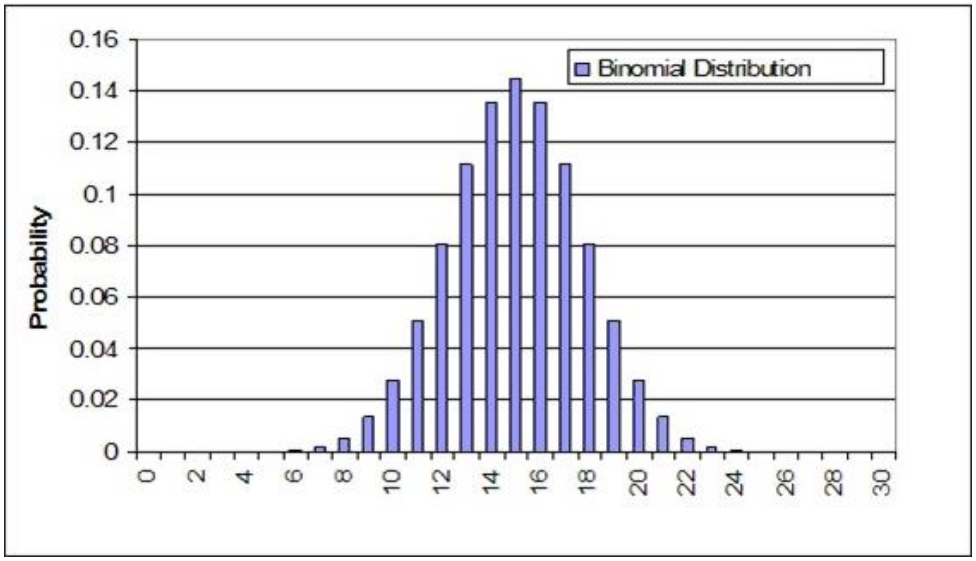
Confidence Intervals for Accuracy

- Classification of an instance is a Bernoulli trial.
  - A Bernoulli trial has 2 outcomes: correct and wrong;
  - The random variable X, the number of correct outcomes of N Bernoulli trials, has a Binomial distribution $b(x; N, acc_D)$;
  - Example: If we have a classifier with true accuracy acc_D equal to 50%, then to classify 30 randomly chosen instances we receive the following Binomial Distribution:



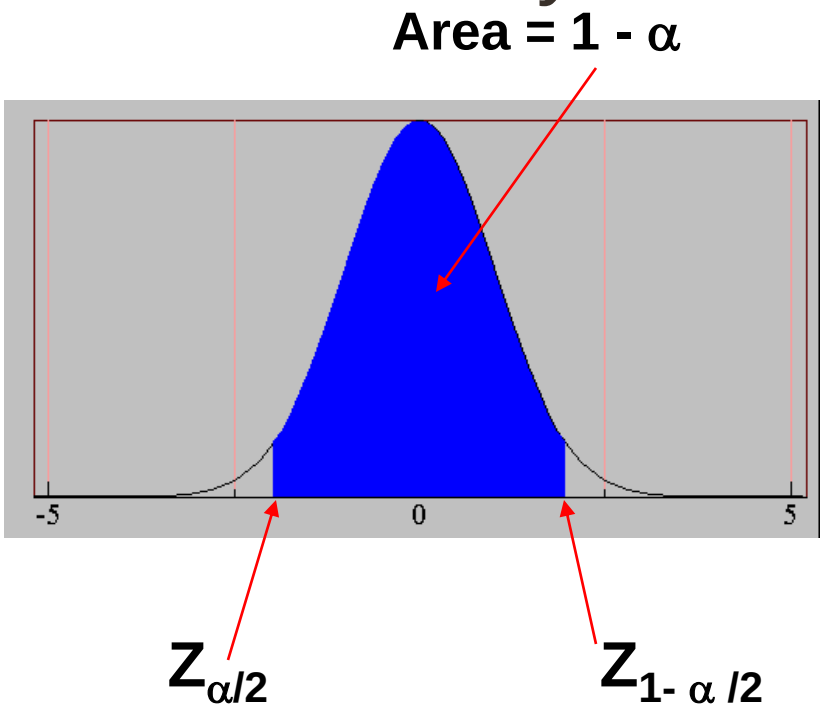
Scenario 1: Confidence Intervals for Accuracy

- The binomial distribution of X has mean equal to $N\ acc_D$ and variance $N\ acc_D(1- acc_D)$.
- It can be shown that the empirical accuracy $acc_S= X / N$ follows also a binomial distribution with mean equal to acc_D and variance $acc_D(1- acc_D)/ N$.



Scenario 1: Confidence Intervals for Accuracy

- For large test sets (N > 30), a binomial distribution is approximated by a normal one with mean acc_D and variance $acc_D(1- acc_D)/N$.



- Thus,

$$P(Z_{\alpha/2} < \frac{acc_S - acc_D}{\sqrt{acc_D(1-acc_D)/N}} < Z_{1-\alpha/2}) = 1-\alpha$$

- Confidence Interval (Wilson score interval) for acc_D:

$$\frac{2\times N\times acc_S + Z_{\alpha/2}^2 \pm \sqrt{Z_{\alpha/2}^2 + 4\times N\times acc_S - 4\times N\times acc_S^{\ 2}}}{2(N+Z_{\alpha/2}^2)}$$

Scenario 1: Confidence Intervals for Accuracy

- Confidence Interval for acc_D: $\frac{2\times N\times acc_S + Z^2_{\alpha/2} \pm \sqrt{Z^2_{\alpha/2} + 4\times N\times acc_S - 4\times N\times acc_S^{\ 2}}}{2(N+Z^2_{\alpha/2})}$
- The confidence intervals shrink when we decrease confidence:

1- α	0.99	0.98	0.95	0.9	0.8	0.7	0.5
$Z_{\alpha/2}$	2.58	2.33	1.96	1.65	1.28	1.04	0.67

- The confidence intervals become tighter when the number *N* of test instances increases. See below the evolution of the intervals for confidence level 95% for a classifier with accuracy 80% on 100 test instances.

N	20	50	100	500	1000	5000
Confidence Interval	[0.58, 0.92]	[0.67, 0.89]	[0.71, 0.87]	[0.76, 0.83]	[0.77, 0.82]	[0.78, 0.81]

Scenario 2: Accuracy from k-fold cross validation

- Consider: apply 10-fold cross validation and obtained 10 accuracy values $acc_1, acc_2, \ldots, acc_{10}$.
- What is the confidence interval for true accuracy (a^*)?
- Assume that the averaged accuracy can be reasonably approximated by a normal distribution:
  - $\bar{a} = \frac{1}{10}\sum acc_i, \;\; \sigma_a = \sqrt{\frac{1}{10}\sum_i (acc_i - \bar{a})}$
  - $\sigma_{\bar{a}} = \frac{1}{\sqrt{10}}\sigma_a$
  - $\bar{a} \sim N(a^*, \sigma_{\bar{a}})$.
  - Confidence interval: $\bar{a} \pm z_{\alpha/2}\sigma_{\bar{a}}$

Comparing Performance of Two Classifiers

- Two classifiers, M_1 and M_2, want to know which one is better?
- Prepare K training-testing datasets (e.g. K-fold cross validation).

Dataset	Classifier A Performance	Classifier B Performance	Difference
Train1-Test1	A1	B1	d1=A1-B1
Train2-Test2	A2	B2	d2=A2-B2
…	…	…	…
TrainK-TestK	Ak	Bk	dk=Ak-Bk

- Question: Is the performance difference statistically significant?

Comparing Performance of Two Classifiers

- Note that the two performance metrics are **paired**.
- A1 and B1 are trained and tested on the same training and testing dataset.
- If the true accuracy of Classifier A and B are μ_A and μ_B.
- What to show that the two performance is different
- $H_0: \mu_A - \mu_B = 0;$
- $H_A: \mu_A - \mu_B \neq 0.$

Dataset	Classifier A Performance	Classifier B Performance	Difference
Train1-Test1	A1	B1	d1=A1-B1
Train2-Test2	A2	B2	d2=A2-B2
…	…	…	…
TrainK-TestK	Ak	Bk	dk=Ak-Bk

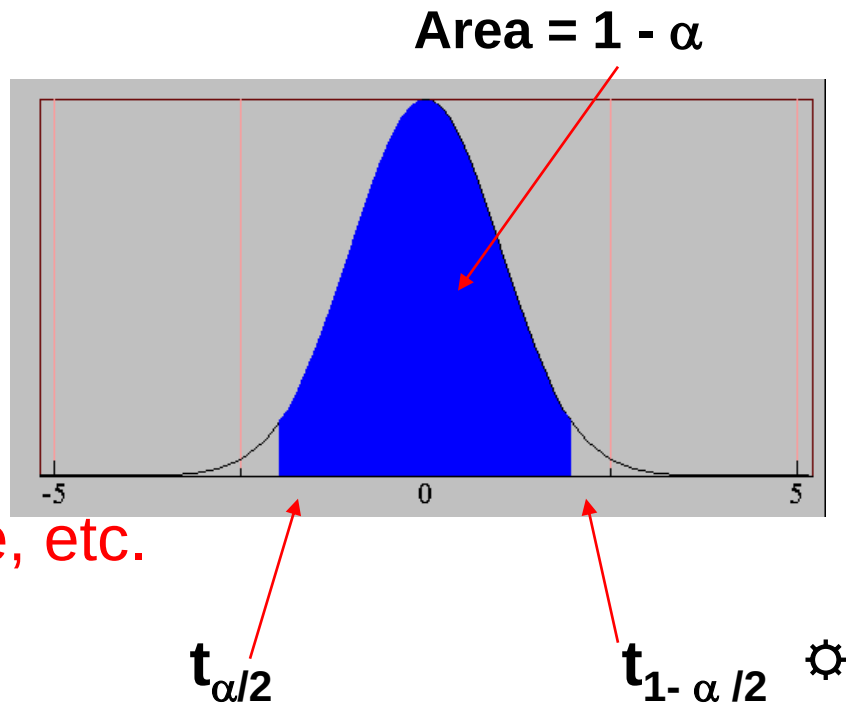
Comparing Performance of Two Classifiers

- Under the null hypothesis, the performance difference $d_1, d_2, \ldots, d_k$ has a mean 0.
- The standard deviation is $s_d = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}\left(d_i - \bar{d}\right)^2}$
- When k is large enough (e.g. $k > 30$), $\bar{d}$ will be approximately normal.
- We can test the null hypothesis via $t = \frac{\bar{d}-0}{s_d/\sqrt{k}}$.

Dataset	Classifier A Performance	Classifier B Performance	Difference
Train1-Test1	A1	B1	d1=A1-B1
Train2-Test2	A2	B2	d2=A2-B2
…	…	…	…
TrainK-TestK	Ak	Bk	dk=Ak-Bk

Comparing Performance of Two Classifiers

- The t statistics is governed by t-distribution with *k - 1* degrees of freedom.
- Confidence interval for performance difference
- $\bar{d} - t_{\frac{\alpha}{2}} \frac{s_d}{\sqrt{n}} < \mu_A - \mu_B < \bar{d} + t_{\frac{\alpha}{2}} \frac{s_d}{\sqrt{n}}$
- $t_{\alpha/2}$ is the *t*-value with v = k -1 degrees of freedom, leaving an area of α/2 to the right.
- Note: The procedure can be applied for various classification, regression, ranking measures such as RMSE, MAE, recall, F-measure, etc.



Comparing Performance of Many Classifiers

- When comparing the performance of three or more classifiers, it is not a good idea to perform t-test on all pairs of classifiers.
- Why?
- Because of the inflated alpha problem!

- Consider the case of comparing the performance of 6 classifiers.
- We need to consider $C_2^6 = \frac{6\times5}{2} = 15$ pairs.
- If using $\alpha = 0.05$ for each t-test, then the chance of finding at least one significant difference (given that all performance are the same) is:

$$1 - 0.95^{15} = 0.563$$

- Meaning: the alpha is inflated.

Comparing Performance of Many Classifiers

- Performance metric: hitrate@3
- Six models, 10 training-testing datasets (10 fold cv)

test_file	Coco	knn	logic	sexy	sexy2	pop
fold1	58.69	57.25	50.36	58.71	59.47	30.23
fold2	58.47	57.27	50.09	58.37	59.23	30.28
fold3	58.4	57.3	50.14	58.49	59.33	30.28
fold4	58.77	57.28	50.46	58.67	59.76	30.59
fold5	58.67	57.41	50.44	58.86	59.74	30.49
fold6	58.87	57.81	50.56	58.89	59.69	30.62
fold7	58.64	57.58	50.49	58.71	59.6	30.52
fold8	58.67	57.71	50.44	58.6	59.37	30.7
fold9	58.6	57.49	50.12	58.7	59.4	30.18
fold10	58.84	57.79	50.36	59.04	59.41	30.49

Questions

- Q1: Does the six models differ in terms of hitrate@3?
- Q2: Does one model outperformed another model?

- This dataset is generated through so called “repeated measure experiments”
- Q1 can be analyzed by two-way ANOVA.
- Q2 can be analyzed by Tukey’s Test.

Two-Way ANOVA

- “Repeated measure experiments” and “randomized block design” can be analyzed using the same techniques: two-way ANOVA.
- We will use these terms interchangeably in the following discussion.
- Each classifier is a “treatment.”
- Each train-test split is a “block.”

test_file	Coco	knn	logic	sexy	sexy2	pop
fold1	58.69	57.25	50.36	58.71	59.47	30.23
fold2	58.47	57.27	50.09	58.37	59.23	30.28
fold3	58.4	57.3	50.14	58.49	59.33	30.28
fold4	58.77	57.28	50.46	58.67	59.76	30.59
fold5	58.67	57.41	50.44	58.86	59.74	30.49

Randomized Blocks

In addition to ***k*** treatments, we introduce notation for ***b*** blocks in our experimental design…

mean of the observations of the 1st treatment

	Treatments				
Block	1	2	…	k	Block Mean
1	x_{11}	x_{12}	…	x_{1k}	$\bar{x}[B]_1$
2	x_{21}	x_{22}	…	x_{2k}	$\bar{x}[B]_2$
:	:	:		:	:
b	x_{b1}	x_{b2}	…	x_{bk}	$\bar{x}[B]_b$
Treatment Mean	$\bar{x}[T]_1$	$\bar{x}[T]_2$	…	$\bar{x}[T]_k$	

mean of the observations of the 2nd treatment

Sum of Squares : Randomized Block

Squaring the 'distance' from the grand mean, leads to the following set of formulae…

$$SS(Total) = \sum_{j=1}^{k} \sum_{i=1}^{b} (x_{ij} - \bar{\bar{x}})^2 \qquad SSB = \sum_{i=1}^{b} k(\bar{x}[B]_i - \bar{\bar{x}})^2$$

$$SST = \sum_{j=1}^{k} b(\bar{x}[T]_j - \bar{\bar{x}})^2 \qquad SSE = \sum_{j=1}^{k} \sum_{i=1}^{b} (x_{ij} - \bar{x}[T]_j - \bar{x}[B]_i + \bar{\bar{x}})^2$$

$$MST = \frac{SST}{k-1}$$

$$MSB = \frac{SSB}{b-1}$$

$$MSE = \frac{SSE}{n-k-b+1}$$

test statistic for ***treatments***

$$F = \frac{MST}{MSE}, \ \nu_1 = k-1 \ \& \ \nu_2 = n-k-b+1 \ d.f.$$

$$F = \frac{MSB}{MSE}, \ \nu_1 = b-1 \ \& \ \nu_2 = n-k-b+1 \ d.f.$$

test statistic for ***blocks***

ANOVA Table

We can summarize this new information in an **an**alysis **o**f **va**riance (ANOVA) table for the randomized block analysis of variance as follows…

Source of Variation	d.f.:	Sum of Squares	Mean Square	***F*** Statistic
Treatments	k–1	SST	MST=SST/(k–1)	F=MST/MSE
Blocks	b–1	SSB	MSB=SSB/(b-1)	F=MSB/MSE
Error	n–k–b+1	SSE	MSE=SSE/(n–k–b+1)	
Total	n–1	SS(Total)		

Example

```
> head(df1)
  Method test_file hitrate.top3
1   Coco     fold1        58.69
2   Coco     fold2        58.47
3   Coco     fold3        58.40
…
```

```
setwd('your_folder')
df1=read.csv('hitrate@3.csv')
#convert string to factor before ANOVA
df1$Method = factor(df1$Method)
df1$test_file = factor(df1$test_file)
#Two-way ANOVA
out1=aov(hitrate.top3~Method+test_file,
data=df1); print(summary(out1))
```

```
            Df Sum Sq Mean Sq   F value   Pr(>F)
Method       5   6417  1283.4 86782.758  < 2e-16 ***
test_file    9      1     0.1     8.843 1.64e-07 ***
Residuals   45      1     0.0
---
```

Q1: Does the six models differ in terms of hitrate@3?

- MST/MSE has an F-value of 86782.758
- The p-value is extremely small < 1e-5
- We can reject H0 (all models have the same performance)
- ➔ At least two models have different hitrate@3.
- Note that block effect is also significant.

```
            Df Sum Sq Mean Sq   F value   Pr(>F)
Method       5   6417  1283.4 86782.758  < 2e-16 ***
test_file    9      1     0.1     8.843 1.64e-07 ***
Residuals   45      1     0.0
---
```

Multiple Comparison Procedure

- Q2: Does one model outperformed another model?
- Recall that conducting t-test on all pairs of classifiers will lead to inflated p-value problem.
- This is called multiple comparison problem in statistical tests.
- A good statistical procedure is Tukey's Multiple Comparison (a.k.a. Tukey Honest Significant Differences)

Tukey's Procedure

- Multiple comparison in randomized block design.
- ω =Critical range = $Q_\alpha \sqrt{\frac{MSE}{b}}$
- MSE: from Two-way ANOVA
- *b:* number of blocks (# of train-test split)
- *k:* number of treatments (# of classifiers)
- Q_α*:* the upper-tail critical value from a Studentized range distribution having k and (b-1)(k-1) degree of freedom (R: qtukey()).
- α: significant level (e.g. 0.05)

Tukey Procedure

- Select a pair of means. Calculate $\left|\bar{x}_i - \bar{x}_j\right|$.
- If $\left|\bar{x}_i - \bar{x}_j\right| > \boldsymbol{\omega}$, there is sufficient evidence to conclude that $\mu_i \neq \mu_j$
- Repeat this procedure for each pair of samples. Rank the means if possible.

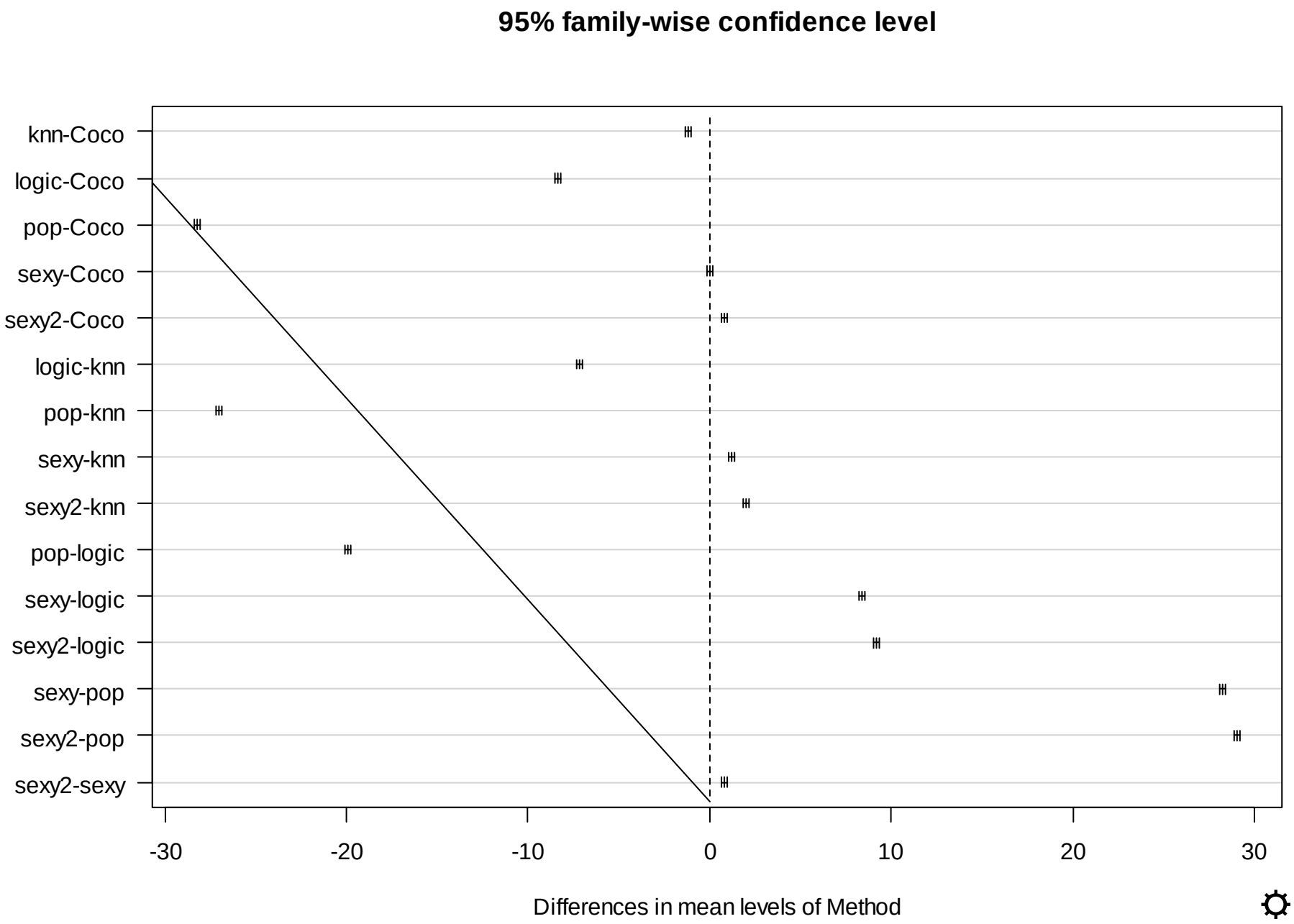
Example (Q2)

- (… continue from the previous R example)
- `posthoc <- TukeyHSD(x=out1, 'Method', conf.level=0.95)`
- `print(posthoc)`
- `#plot results`
- `par(las=1, mar=c(5,7,5,2))`
- `plot(posthoc)`

```
 Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = hitrate.top3 ~ Method + test_file, data = df1)

$Method
                diff        lwr        upr     p adj
knn-Coco      -1.173  -1.334851  -1.011149 0.0000000
logic-Coco    -8.316  -8.477851  -8.154149 0.0000000
pop-Coco     -28.224 -28.385851 -28.062149 0.0000000
sexy-Coco      0.042  -0.119851   0.203851 0.9708136
sexy2-Coco     0.838   0.676149   0.999851 0.0000000
logic-knn     -7.143  -7.304851  -6.981149 0.0000000
pop-knn      -27.051 -27.212851 -26.889149 0.0000000
sexy-knn       1.215   1.053149   1.376851 0.0000000
sexy2-knn      2.011   1.849149   2.172851 0.0000000
pop-logic    -19.908 -20.069851 -19.746149 0.0000000
sexy-logic     8.358   8.196149   8.519851 0.0000000
sexy2-logic    9.154   8.992149   9.315851 0.0000000
sexy-pop      28.266  28.104149  28.427851 0.0000000
sexy2-pop     29.062  28.900149  29.223851 0.0000000
sexy2-sexy     0.796   0.634149   0.957851 0.0000000
```


95% family-wise confidence level
knn-Coco
logic-Coco
pop-Coco
sexy-Coco
sexy2-Coco
logic-knn
pop-knn
sexy-knn
sexy2-knn
pop-logic
sexy-logic
sexy2-logic
sexy-pop
sexy2-pop
sexy2-sexy
-30
-20
-10
0
10
20
30
Differences in mean levels of Method

Q2: Does one model outperformed another model?

- Based on Tukey's Procedure, we can conclude that:

- All pairs of differences are significant at a 95% confidence level, except for `sexy-Coco`.

Metric Evaluation Summary:

- Use test sets and the hold-out method for “large” data;
- Use the cross-validation method for “middle-sized” data;
- Use the leave-one-out methods for small data;
- Don’t use test data for parameter tuning - use separate validation data.
- Use pairwise t-test to compare two models with **synced** training-testing splits.
- Use Two-way ANOVA and Tukey’s Procedure to analyze three or more classifiers.