Lecture 9

Model Evaluation

Golnaz Taheri, PhD
Senior Lecturer, Stockholm University

Outline

- Model Evaluation
- Accuracy/precision/recall/f-measure
- Cross-Validation
- Bootstrapping
- Area under the ROC

Stockholms universitet

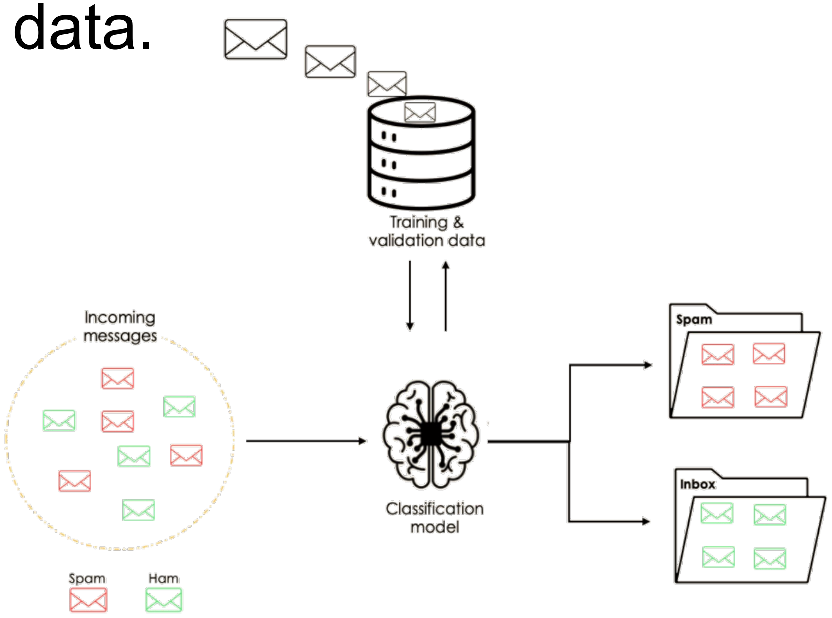
Model Evaluation

- How to evaluate the performance of a model?
  - **Metrics** for Performance Evaluation
  - **Methods** for Performance Evaluation
- How to compare the relative performance of different models?

Classification recap

- Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data.

- In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

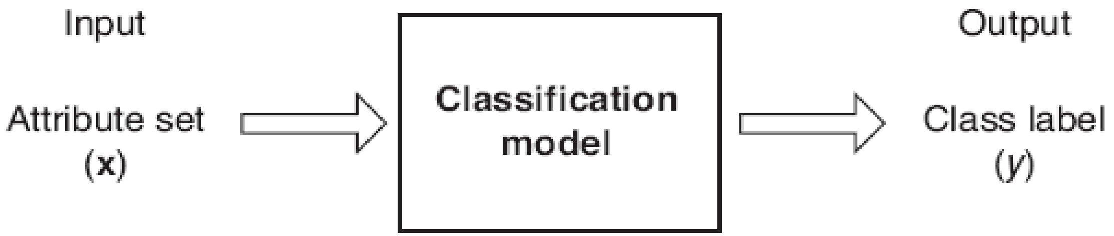


But how good the classification model?

Stockholms universitet

Mean Squared Error

- Suppose we have learned a function $\boldsymbol{f}$ using some training dataset



- Training set $\{x_1, x_2, \ldots, x_n\}$
- Classes of the training set $\{y_1, y_2, \ldots, y_n\}$
- The mean squared error (MSE):

predicted value

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

Stockholms universitet

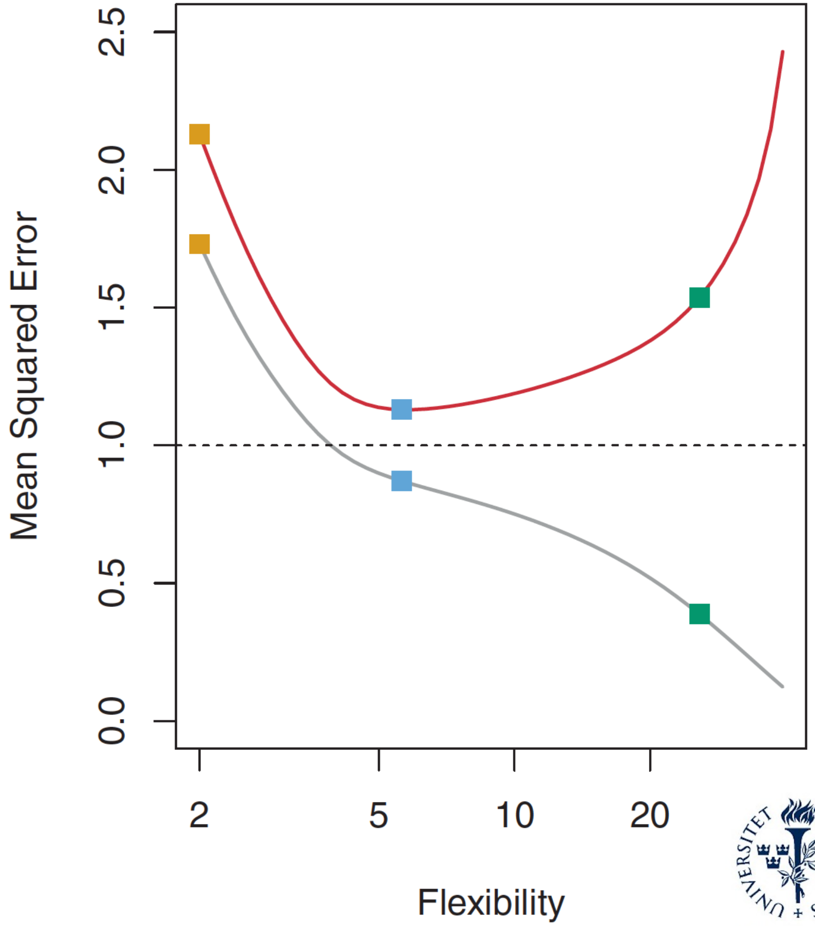
Mean Squared Error

- Is the MSE on the training set good enough?
    - Who cares if we can classify correctly data that we have already seen…
    - Say we build a classifier on seen stock values
    - We would rather want to be able to predict them in the future…

Stockholms universitet

Overfitting and the U-Shape effect

- MSE on the training set:
  - grey line
- MSE on the test set:
  - red line
- Training error always reduces as models become more complex (hence more flexible)
- Test error forms a U-shape



Stockholms universitet

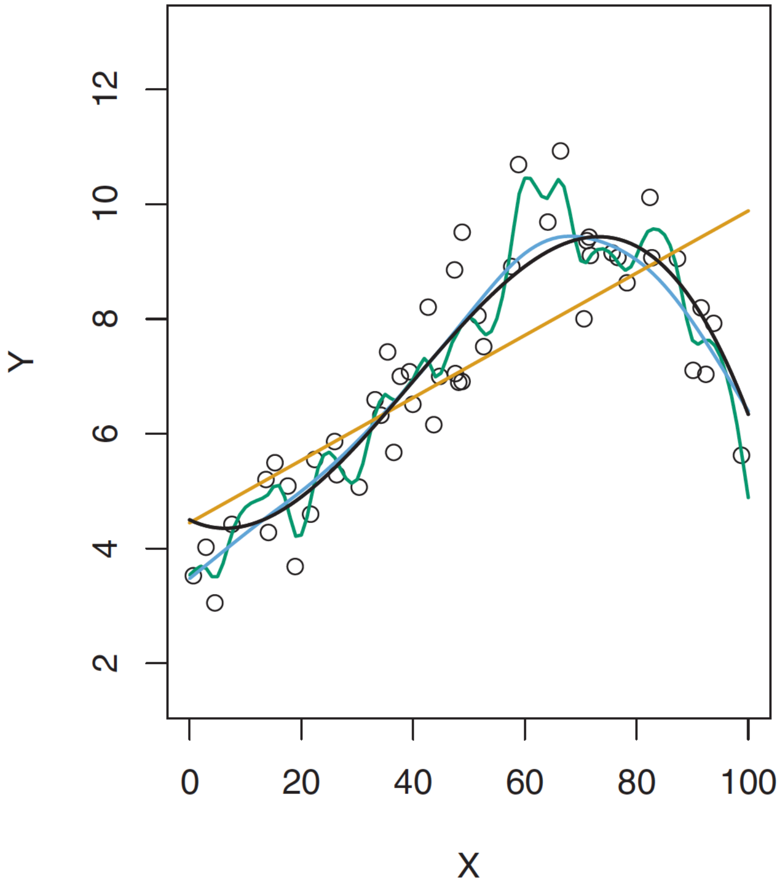
Bias and Variance

- **Variance:** how much would the model change (in performance on a test set) if we used a different training set
- Training sets are used to estimate f, hence a different training set may produce a different f.
- **Ideally:** The estimate of f should not vary too much among different training sets
- Flexible classifiers have higher variance
- High variance may cause the model to learn noise or errors, hence *overfitting*

Stockholms universitet

Bias and Variance

- Flexible classifiers have higher variance
  - Green line: *more flexible*
  - Yellow line: *very inflexible*
  - Which one has a higher variance?



Stockholms universitet

Bias and Variance

- **Bias:** a learner's tendency to "under-learn" the data due to erroneous assumptions
- Error introduced by using a simpler model to learn a complex relation
- High bias may cause the model to miss important relations between the attributes and class labels, hence *underfitting*
- **Ideally:** The estimate of f should have a low bias
- Flexible classifiers have lower bias

Stockholms universitet

Bias and Variance

- Simple models: make many assumptions on the underlying data distribution (high bias), but they are not that sensitive to the training set (low variance)

- Complex models: make few (or no) assumptions on the underlying data distribution (low bias), but they may end up being very sensitive to the training set (high variance)

Bias and Variance

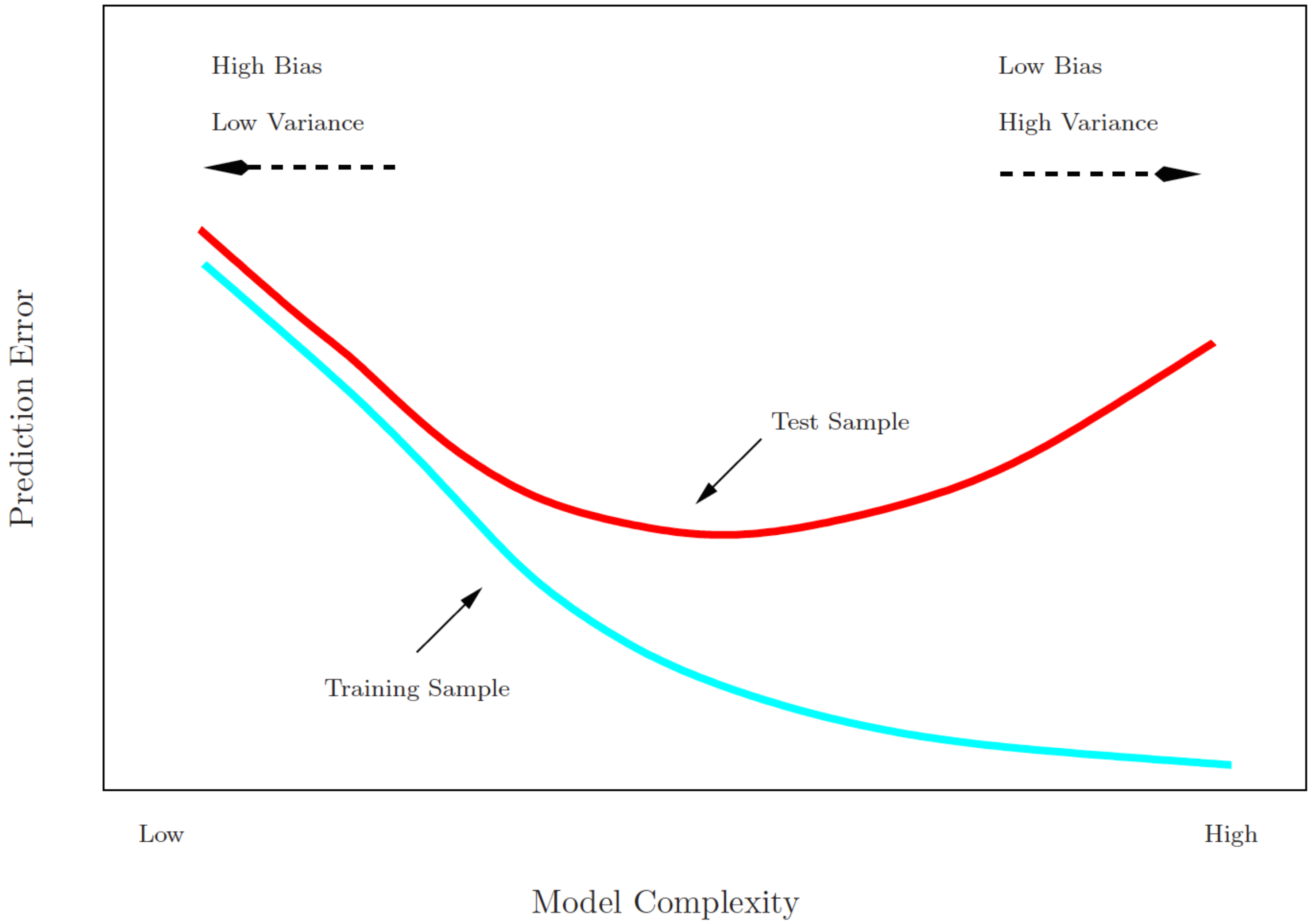
- **Hence, we want models with:**
  - Low variance: they should not be sensitive to the training set, they should not overfit
  - Low bias: they should not make unrealistic assumptions, they should be as simple as possible, they should not under-learn
- It is easy to obtain models that are good at one of these two metrics
- Challenge: obtain models that achieve a good trade-off between variance and bias

Bias and Variance

- Decision trees:
  low bias
  - Can represent any boolean function between the attributes
  - If the training set contains noise then the constructed trees may vary
  high variance
- A linear classifier:
  - Assumes a linear separator
  high bias
  - If the separator of two classes is not a line it won't learn it!
  - Less flexible
  low variance

Stockholms universitet

Bias and Variance



Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - rather than how fast it takes to classify or build models, scalability, etc.

- Confusion Matrix:

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)

Stockholms universitet

Accuracy

- Most widely-used metric:

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)

Accuracy

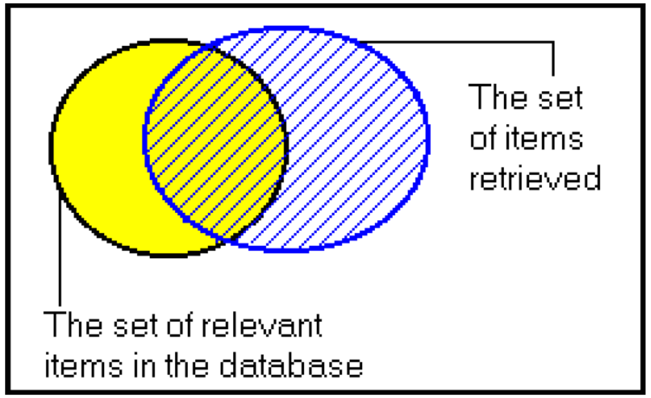
$$\frac{TP + TN}{(TP + TN + FP + FN)}$$

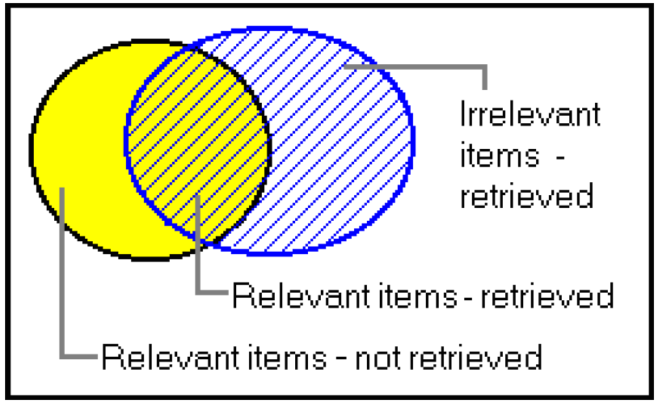
Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 1 examples = 999
  - Number of Class 0 examples = 1

- One solution:
  - A model that predicts everything to be Class 1
  - Accuracy: 999/1000 = 99.9 % ☺
  - Anything went wrong?
  - Accuracy is misleading because the model does not detect any class 0 example

Precision and Recall

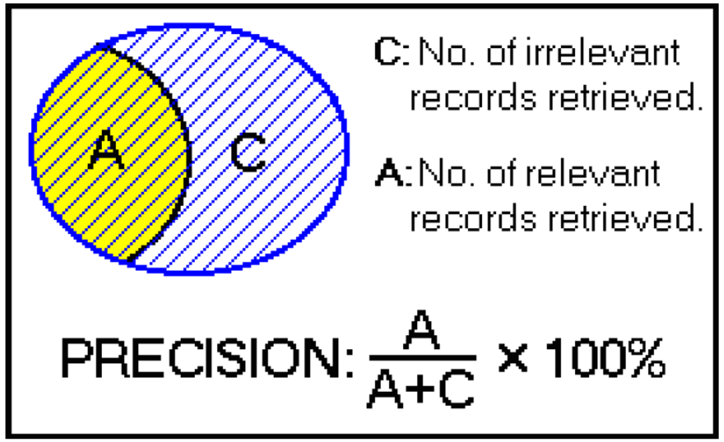


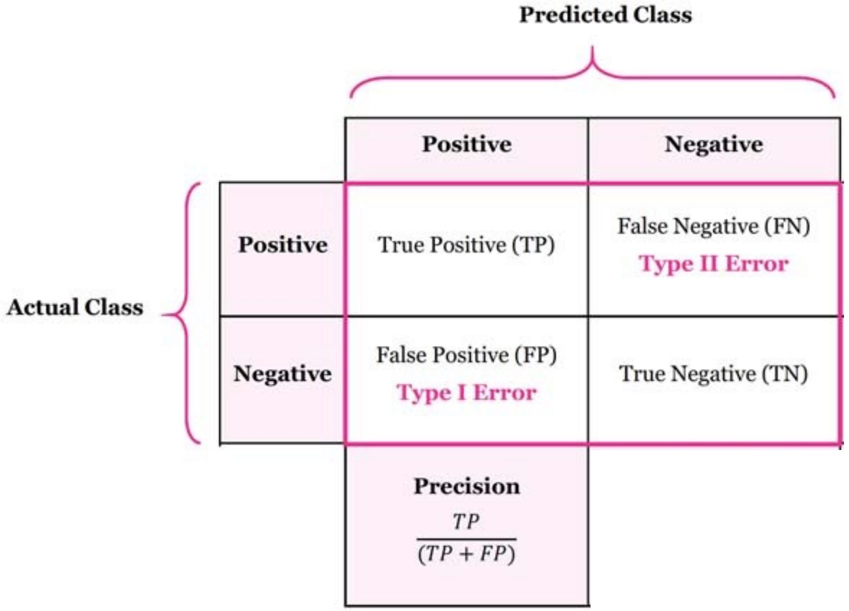


- There are relevant items in the database that need to be retrieved
- There are items that are retrieved
- There are relevant items that are retrieved
- There are some irrelevant items that are retrieved
- There are items that are relevant but not retrieved

Stockholms universitet

Precision



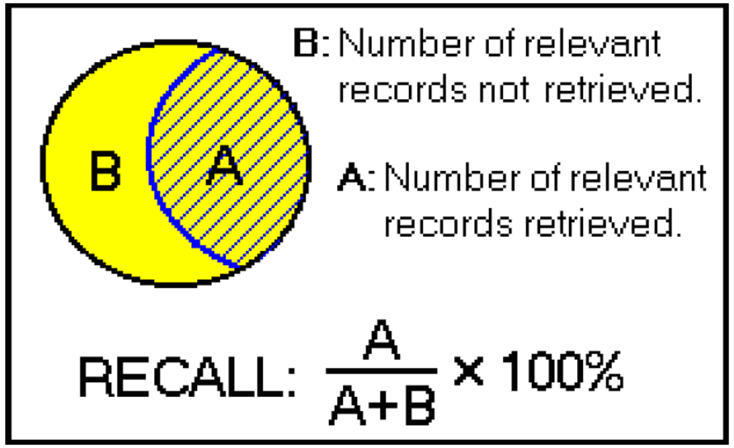


- If we claim an example belong to a class, what is the chance we are correct?

Precision = 1?

- All records that are retrieved are relevant
- May be missing some relevant records that are not retrieved

Recall



- What proportion of the the class did we correctly retrieve?

Recall = 1?

- All relevant records are retrieved
- We may have some retrieved records that are irrelevant

Evaluation Measures

$$\text{Precision (class=YES)} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

$$\text{Precision (class=NO)} = \frac{d}{b+d} = \frac{TN}{FN+TN}$$

$$\text{Recall (class=YES)} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

$$\text{Recall (class=NO)} = \frac{d}{c+d} = \frac{TN}{FP+TN}$$

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a: TP	b: FN
	Class=No	c: FP	d: TN

$$\text{F-measure (class=YES)} = \frac{2 recall * precision}{recall + precision} = \frac{2a}{2a+b+c} = \frac{2TP}{2TP+FP+FN}$$

F-measure is the harmonic mean of precision and recall

Precision and recall are computed for each class!

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

Generalized F-measure: Recall is β times more important than precision

Stockholms universitet

When is precision more important?

Precision more useful when we want to confirm correctness of model

- Precision is more useful when we want to confirm the correctness of our model.

- YouTube recommendations

- Reducing the number of FP is of most importance.

- FP means videos that the user does not like, but YouTube is still recommending them.

- FN are of lesser importance here since the YouTube recommendations should only contain videos that the user is more likely to click on.

Stockholms universitet

When is recall more important?

- COVID-19 detection

- We want to avoid FN as much as possible.

- A FN case means that a COVID-positive patient is assessed to not have the disease, which is harmful.

- In this use case, FP (a healthy patient diagnosed as COVID-positive) are not as important as preventing a contagious patient from spreading the disease.

- In high-risk disease detection cases like cancer, recall is a more important evaluation metric than precision.

Stockholms universitet

Methods for Performance Evaluation

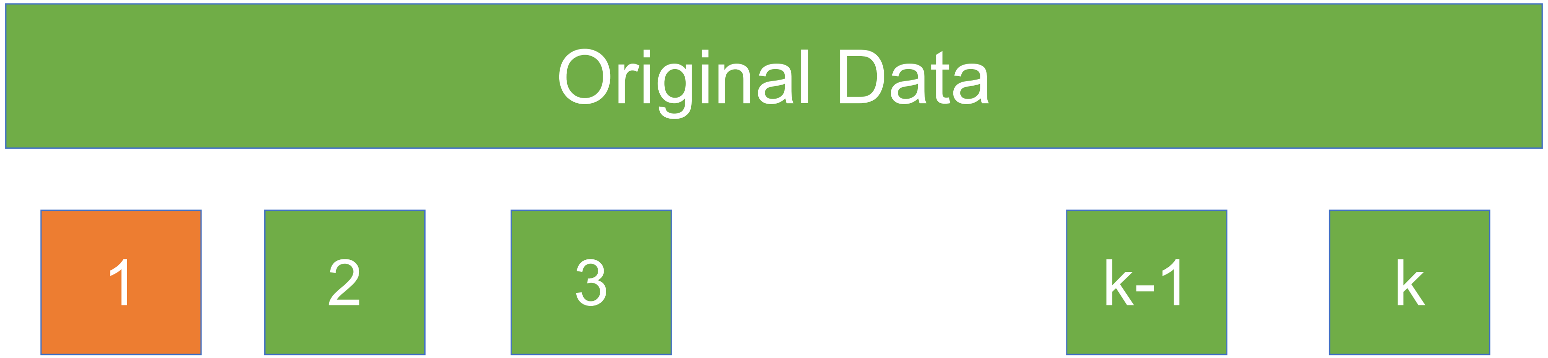
- How to obtain a reliable estimate of performance?
- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Size of training and test sets
  - Structure of the training and test sets

Methods of Estimation

- **Holdout (or using a validation set)**
  - Reserve **2/3** for training and **1/3** for testing
- **Random subsampling**
  - Repeated holdout
- **Stratified subsampling**
  - Preserve the class balance in the samples
- This procedure:
  - can have **high variance**
  - may depend heavily on which data points end up in the training set and which end up in the test set
  - **High bias:** The error rate will tend to over-estimate the error rate of the model that would be trained on the whole dataset

Cross Validation

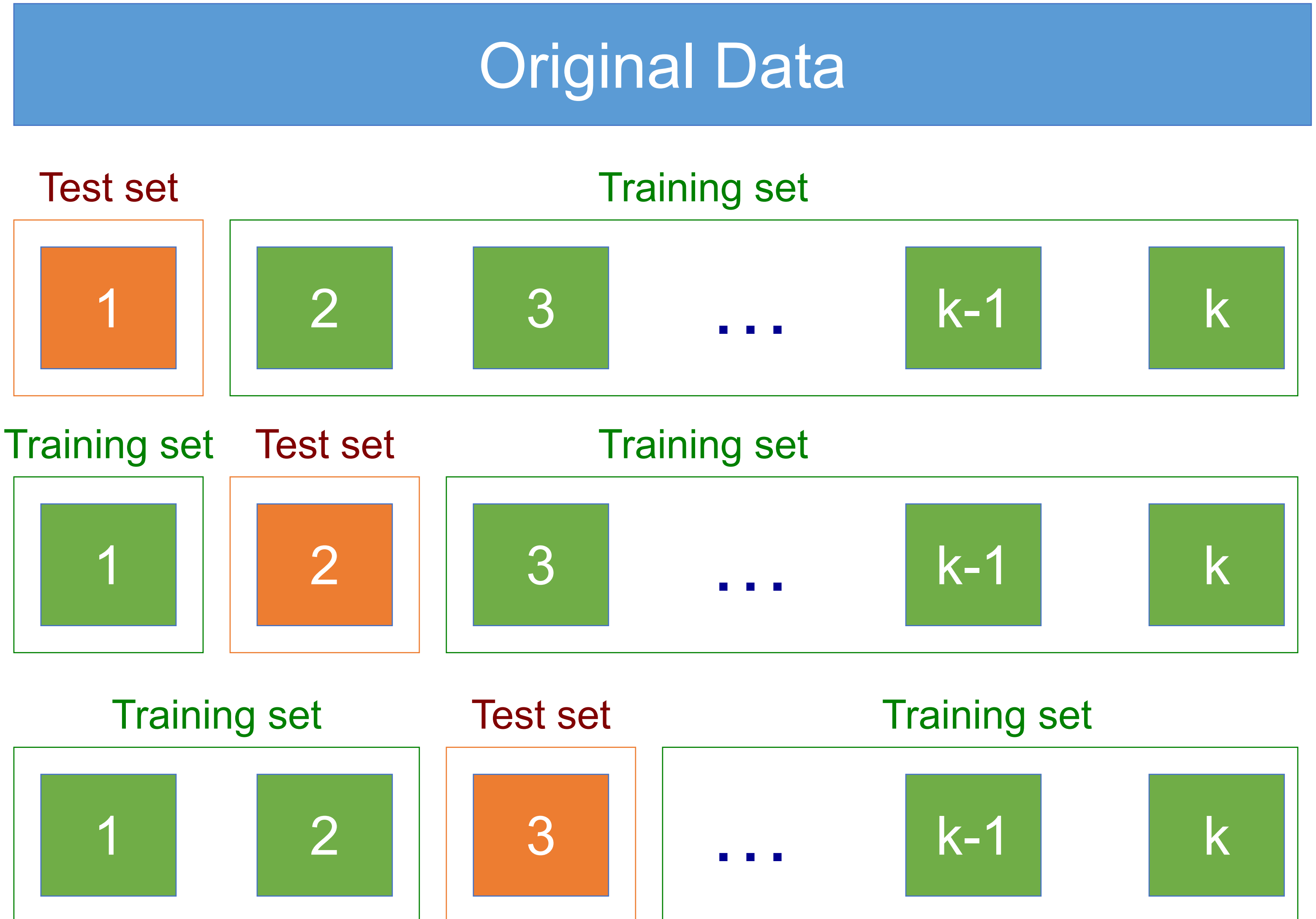
- Partition data into k disjoint subsets
- k-fold:
  - **Train** on $k-1$ partitions
  - **Test** on the remaining one



- Each subset is given a chance to be in the test set
- Performance measure is averaged over the k iterations

Stockholms universitet

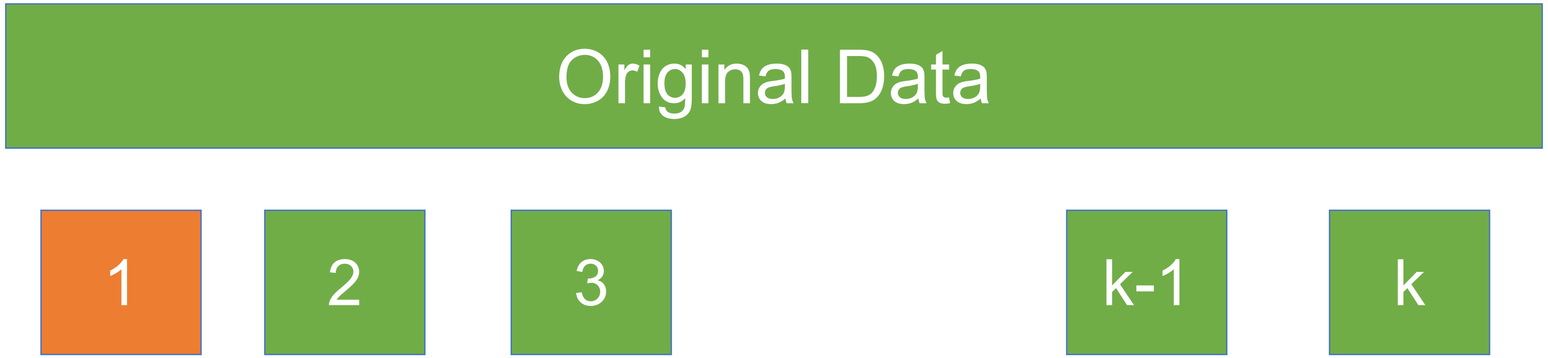
Cross Validation



Stockholms universitet

Cross Validation

- Partition data into k disjoint subsets
- k-fold:
  - **Train** on $k-1$ partitions
  - **Test** on the remaining one

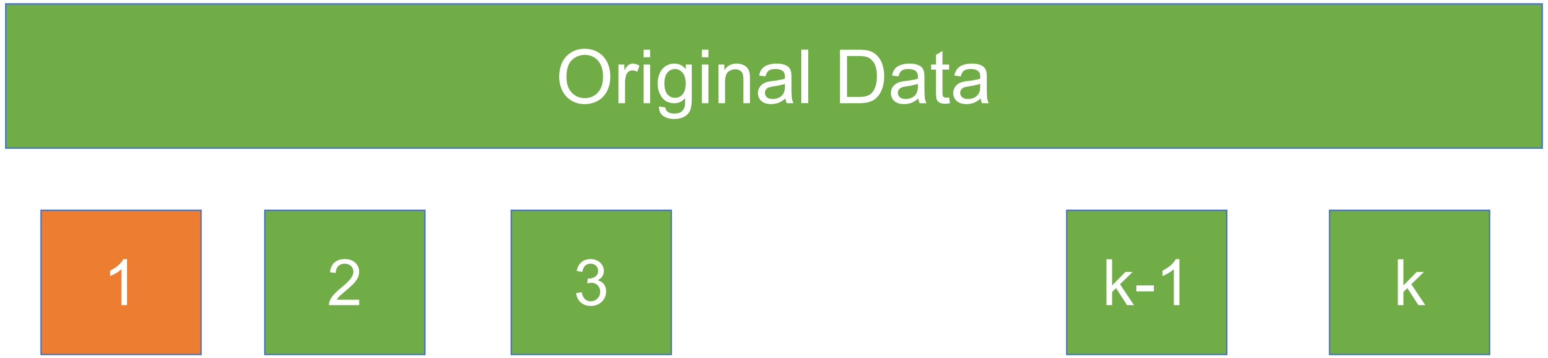


- Compute the average metric (accuracy, precision, recall, etc) for all rounds

Stockholms universitet

Cross Validation

- Partition data into k disjoint subsets
- k-fold:
  - **Train** on $k-1$ partitions
  - **Test** on the remaining one



- **Leave-one-out Cross validation**
  - Special case where $k = n$ **(n is the size of the dataset)**

k-fold vs. leave-one-out?

- Leave-one-out

- **Advantage:**
  - The test error is very close to the true error since the training set is "almost" stable

- **Disadvantage:**
  - Training on almost identical training sets produces models that are highly correlated

- The computational time will be very large: the training algorithm has to be rerun from scratch n times

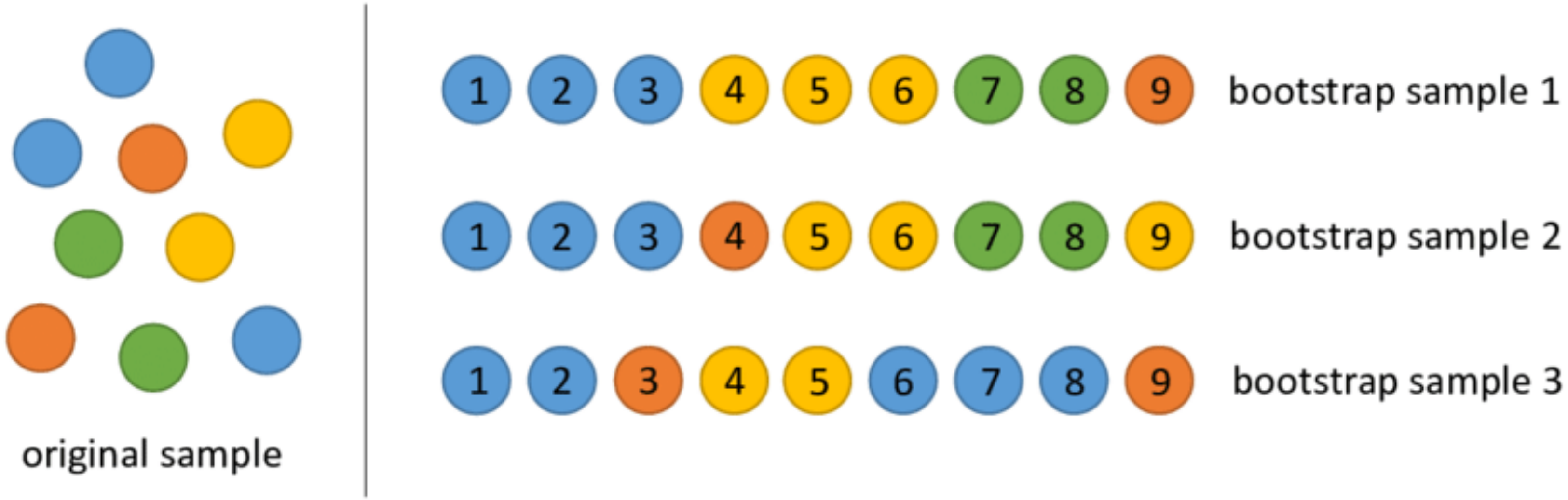
Stockholms universitet

k-fold vs. leave-one-out?

- k-fold
- **Advantage:**
  - Less correlated training sets → less correlated models
  - The computational time is lower
- **Disadvantage:**
  - The bias of the procedure increases: the training set will contain fewer examples
  - The lower the k the higher the bias!
  - In the extreme case it will become equivalent to holdout

Bootstrapping

- Samples the given training examples uniformly with replacement
  - each time an example is selected, it is equally likely to be selected again and re-added to the training set



Stockholms universitet

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the *trade-off* between positive hits and false alarms
- **ROC** curve plots True Positive Rate (**TPR**) **(or sensitivity)** (on the **y**-axis) against False Positive Rate (**FPR**) **(or 1-specificity)** (on the **x**-axis)
- **Specificity** = True Negative Rate (**TNR**)

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

	PREDICTED CLASS		
Actual Class		Yes	No
	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

Stockholms universitet

ROC (Receiver Operating Characteristic)

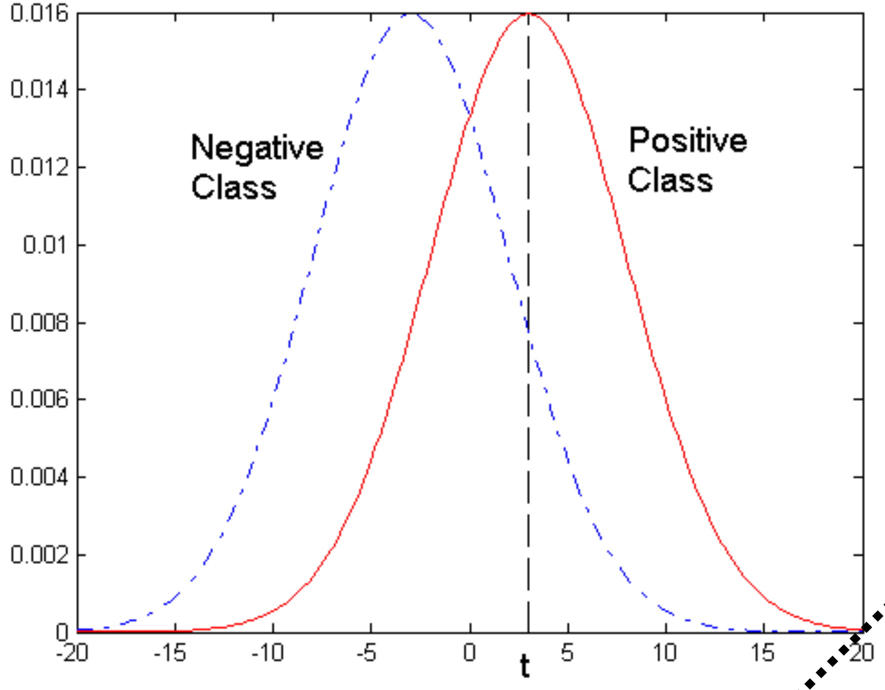
- Performance of each classifier represented as a point on the **ROC** curve
  - changing some threshold of the algorithm
  - or the sample distribution
  - or cost matrix

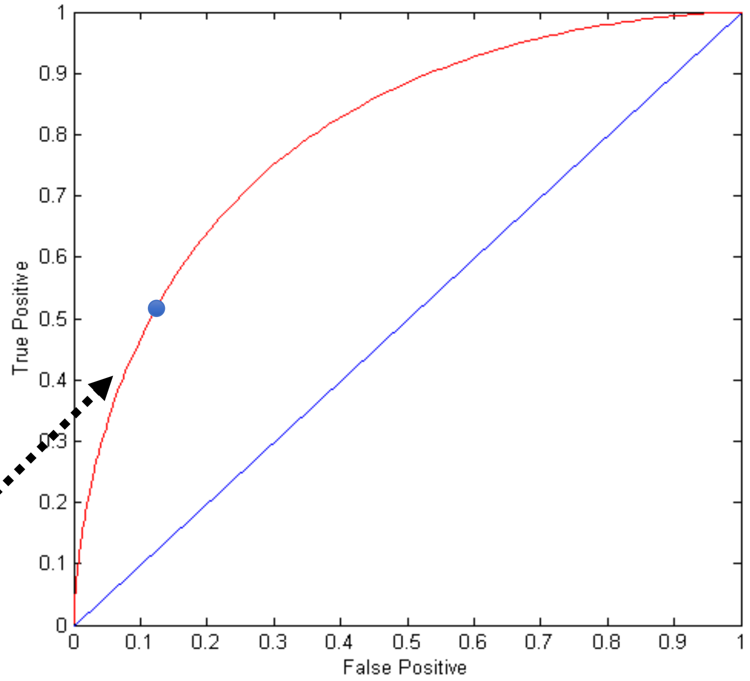
  => changes the location of the point

ROC Curve

- **1**-dimensional data set containing **2** classes (***positive*** and ***negative***)

- **Classifier:** any point located at **x > t** is classified as ***positive***





At threshold **t**:

Rates TP=0.5, FN=0.5, FP=0.12, TN=0.88

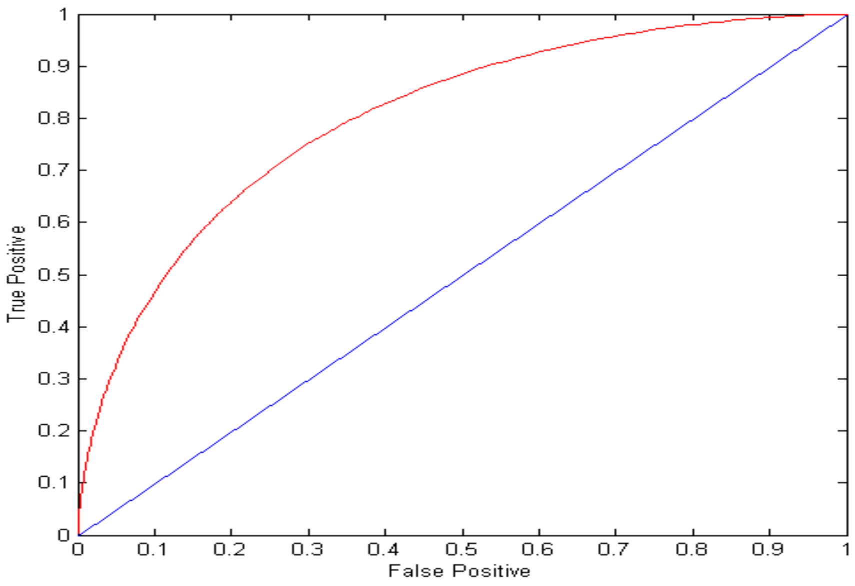
ROC Curve

(TPR, FPR):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal

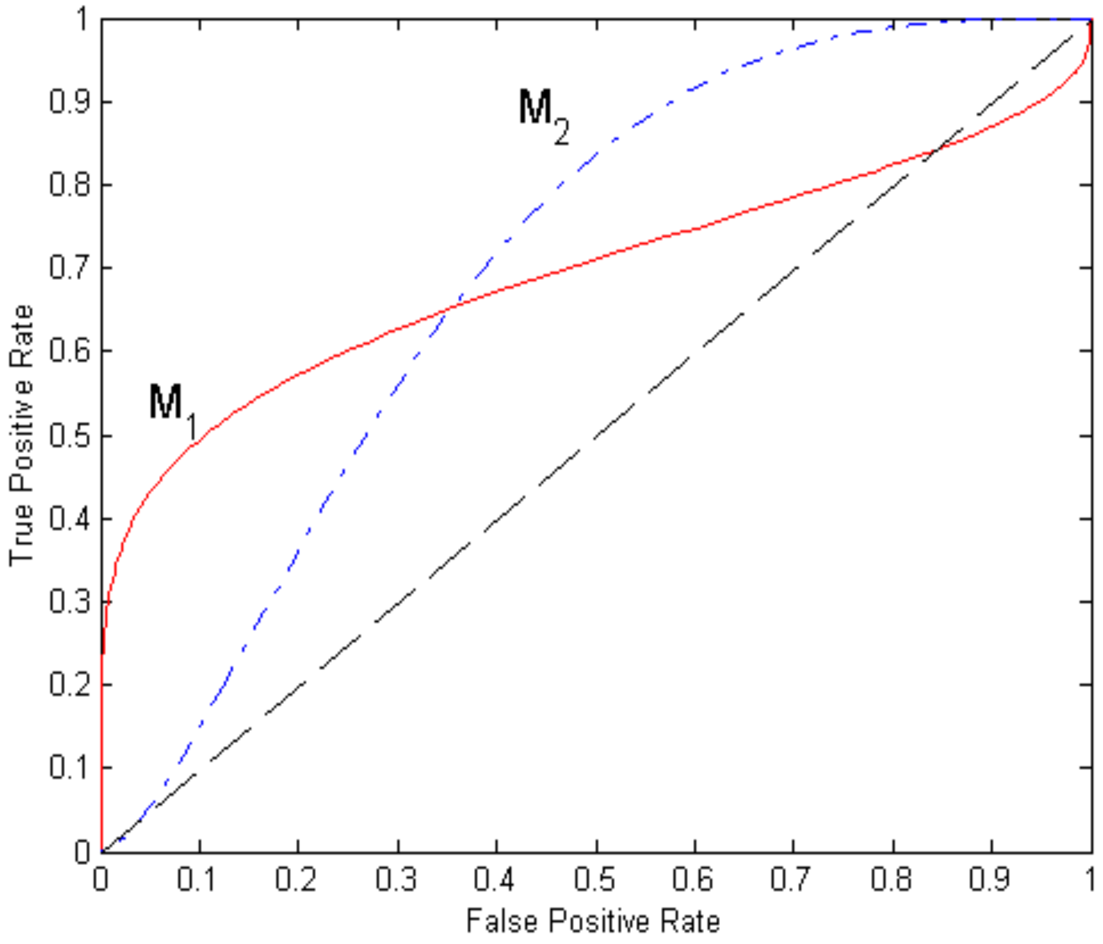
- Diagonal line:
  - random guessing
- Below diagonal line:
  - prediction is opposite of the true class

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$



	PREDICTED CLASS		
		Yes	No
Actual Class	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

Using ROC for Model Comparison



- Area Under the ROC Curve
  - Ideal: Area = 1
  - Random guess: Area = 0.5

- **AUC of a classifier** is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example = **classifier's skill**

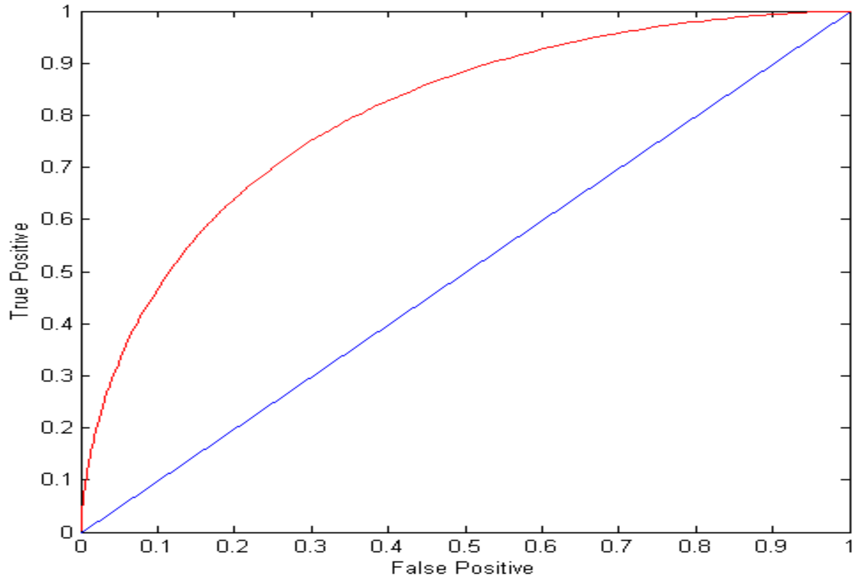
- **AUC vs. Accuracy:** a random guessing classifier may achieve high accuracy if "lucky" or the test set is highly imbalanced; in terms of AUC it will score 0.5

Stockholms universitet

ROC Curve

- Assume: random classifier that 90% of the time predicts the positive class
- **TPR: 0.9**
  - 90% of the positive class examples will be predicted correctly
- **FPR: 0.9**
  - 90% of the negative class examples will be predicted incorrectly
- Falls on the blue line always!
- **WHY?**

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$



	PREDICTED CLASS		
		Yes	No
Actual Class	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

ROC Curve

- Assume: random classifier that 90% of the time predicts the positive class
- **TPR: 0.9**
  - 90% of the positive class examples will be predicted correctly
- **FPR: 0.9**
  - 90% of the negative class examples will be predicted incorrectly
- Falls on the blue line always!
- **WHY?**

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

Suppose you have:

- P positives
- N negatives

10% of time predicts negative class

	PREDICTED CLASS	
Actual Class	Yes	No
Yes	a (TP)	b (FN)
No	c (FP)	d (TN)

Then:

- TP = 0.9P
- FN = 0.1P
- FP = 0.9N
- TN = 0.1N

- TPR = 0.9P / (0.9P + 0.1P) = 0.9
- FPR = 0.9N / (0.9N + 0.1N) = 0.9

Stockholms universitet

ROC Curve

- Assume: random classifier that 90% of the time predicts the positive class
- **TPR: 0.9**
  - 90% of the positive class examples will be predicted correctly
- **FPR: 0.9**
  - 90% of the negative class examples will be predicted incorrectly
- Falls on the blue line always!
- **WHY?**

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

Suppose you have:

- P positives
- N negatives

	PREDICTED CLASS		
Actual Class		Yes	No
	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

Then:

- TP = αP
- FN = (1- α)P
- FP = αN
- TN = (1- α)N

- TPR = α P / (α P + (1- α) P) = α
- FPR = α N / (α N + (1- α) N) = α

How to Construct a ROC curve

Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces posterior probability for each test instance P(+|A)
- Sort the instances according to P(+|A) in decreasing order
- Apply threshold at each unique value of P(+|A)
- Count the number of TP, FP, TN, FN at each threshold
- TPR = TP/(TP+FN)
- FPR = FP/(FP+TN)

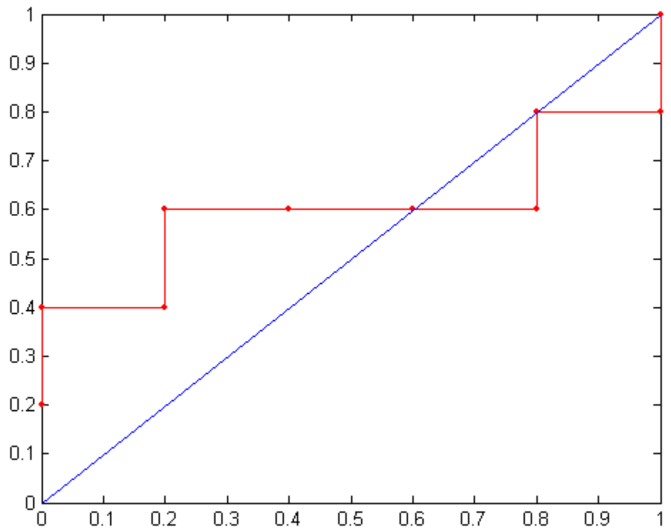
How to construct a ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

TPR = TP/(TP+FN)

FPR = FP/(FP+TN)

Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

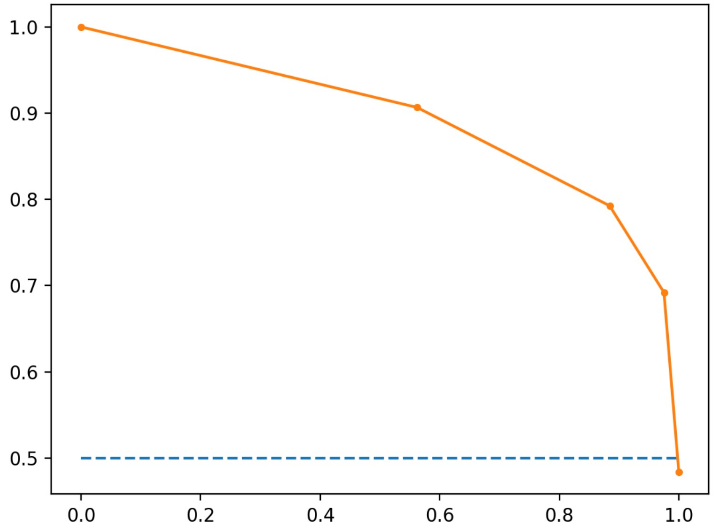


Recall - fraction of relevant examples that are retrieved

Precision-Recall Curve (AUPRC)

- Precision vs Recall trade-offs

$$\frac{TP}{TP+FP} \quad \text{vs} \quad \frac{TP}{TP+FN}$$



Useful in cases of class imbalance

Many examples of class 0 and only a few examples of class 1

Less interested in the skill of predicting examples of class 0 than rare examples of class 1

Less interested in True Negatives

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4349800/

AUC vs AUPRC

Suppose we have 1 million examples (e.g., patients) out of which 100 are of class A (e.g., sick)

- **M1:** 100 predicted sick, 90 correctly sick
- **M2:** 2000 predicted sick, 90 correctly sick

- **M1:** TPR = 0.9, FPR = 0.00001 (10/million)
- **M2:** TPR = 0.9, FPR = 0.00191 (1910/million)

FPR difference of 0.0019

- **M1:** prec = 0.9, recall = 0.9
- **M2:** prec = 0.045, recall = 0.9

PR difference of 0.855

AUC

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

AUPRC

$$\frac{TP}{TP + FP}$$

$$\frac{TP}{TP + FN}$$

Precision and recall do not consider true negatives; thus not affected by the relative imbalance

Stockholms universitet

AUC vs AUPRC

- **ROC AUC:** summarizes trade-offs between TPR and FPR for a classifier at different thresholds
- **PR AUC:** summarizes trade-offs between TPR and the positive predictive value of a classifier for different thresholds
- **ROC AUC:** appropriate when we are interested in classifiers equally ”skilled” for the positive and negative classes + better suited for balanced datasets
- **PR AUC:** suitable for imbalanced datasets

Stockholms universitet

TODOs

Reading:

Main course book chapter:

18

Lab 3

Quiz 4

Coming up next

Tuesday

Lecture 10 : Advance Topics | : Neural Net

Lecture 11: Advanced Topics || : Graph Mining

Monday

Stockholms universitet

Thanks!

golnaz.taheri@dsv.su.se