Stima della qualità dei classificatori per l' analisi dei dati biomolecolari

Giorgio Valentini

e-mail: valentini@dsi.unimi.it

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

Rischio atteso e rischio empirico

- L` apprendimento di una funzione non nota $f : R^d \rightarrow$ C avviene tramite un algoritmo L che genera un insieme di funzioni g che approssimano f utilizzando solo un training set $D=\{(x_i,t_i)\}_{i=1}^{n}$ distribuito secondo un distribuzione di probabilità non nota $P(\mathbf{x},t)$:

$$g:\Re^d \times \Omega \rightarrow C$$

Ω rappresenta un insieme di parametri della learning machine (ad es., l' insieme dei pesi delle unità di calcolo di una rete neurale).

• Obiettivo dell' apprendimento non è minimizzare il rischio empirico $R_{emp}(\omega)$:

$$R_{emp}(\omega)=\frac{1}{n}\sum_{i=1}^{n} Loss\left(g(x_i,\omega),t_i\right)$$

bensì il rischio atteso R(ω):

$$R(\omega)=\iint Loss\left(g(x,\omega),t\right)\, p(x,t)\, dx\, dt$$

A parte le difficoltà matematiche della minimizzazione del funzionale R(ω), quasi sempre la funzione di densità di probabilità congiunta non è nota ...

Stima del rischio atteso

- Il rischio empirico non sempre converge al rischio atteso.
- La Teoria Statistica dell' Apprendimento di Vapnik ha mostrato che un limite superiore al rischio atteso può essere scomposto in due componenti:

$$R(\omega) \leq R_{emp}(\omega) + \Phi(h/m)$$

dove il primo termine dipende dal rischio empirico, mentre l' intervallo di confidenza Φ dipende principalmente dal rapporto fra la complessità *h* della learning machine e la cardinalità *m* del training set disponibile.

• Per valutare le capacità di generalizzazione delle learning machine è necessario stimare il rischio atteso e non semplicemente il rischio empirico.

• Il problema è: come stimare il rischio atteso ?

Due approcci principali alla stima del rischio atteso

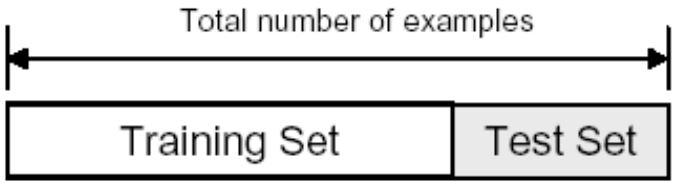
- Stima teorica dei limiti superiori al rischio atteso (basati sull’ errore empirico e sulla stima della complessità della learning machine)
- Stima sperimentale (basata sul campionamento dei dati disponibili)

Metodi di stima sperimentale dell' errore di generalizzazione

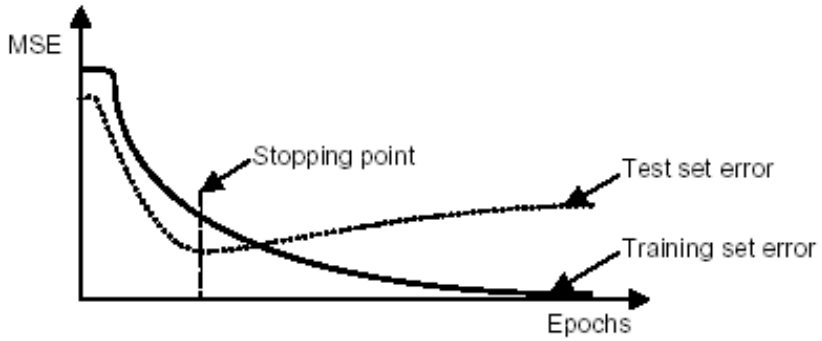
- Holdout
  - Suddivisione dei dati in *training* e *test* set (tipicamente 2/3 ed 1/3)
- Sottocampionamento casuale
  - Holdout ripetuto n volte
- Cross validation
  - Partizione dei dati in k sottoinsiemi disgiunti (fold)
  - k-fold: training con k-1 fold, test sul rimanente; il processo è ripetuto k volte utilizzando ognimvolta come test set un fold differente.
  - Leave-one-out: k = numero dei campioni disponibili
- Bootstrap
  - Campionamento con rimpiazzo
- Metodi out-of-bag
  - Training sui campioni estratti tramite bootstrap e testing sui rimanneti campioni non selezionati. Il proceso è ripetuto n volte.

Metodo di hold-out (1)

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



- **A typical application the holdout method is determining a stopping point for the back propagation error**

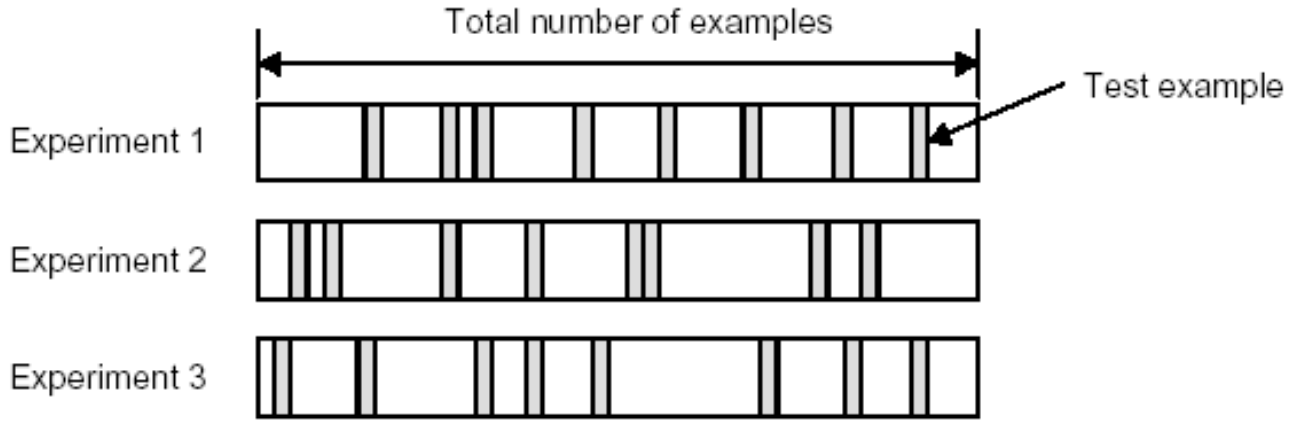


Metodo di hold-out (2)

- **The holdout method has two basic drawbacks**
  - 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
- **The limitations of the holdout can be overcome with a family of resampling methods at the expense of higher computational cost**
  - Cross Validation
    - Random Subsampling
    - K-Fold Cross-Validation
    - Leave-one-out Cross-Validation
  - Bootstrap

Campionamento casuale (holdout ripetuto)

- **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

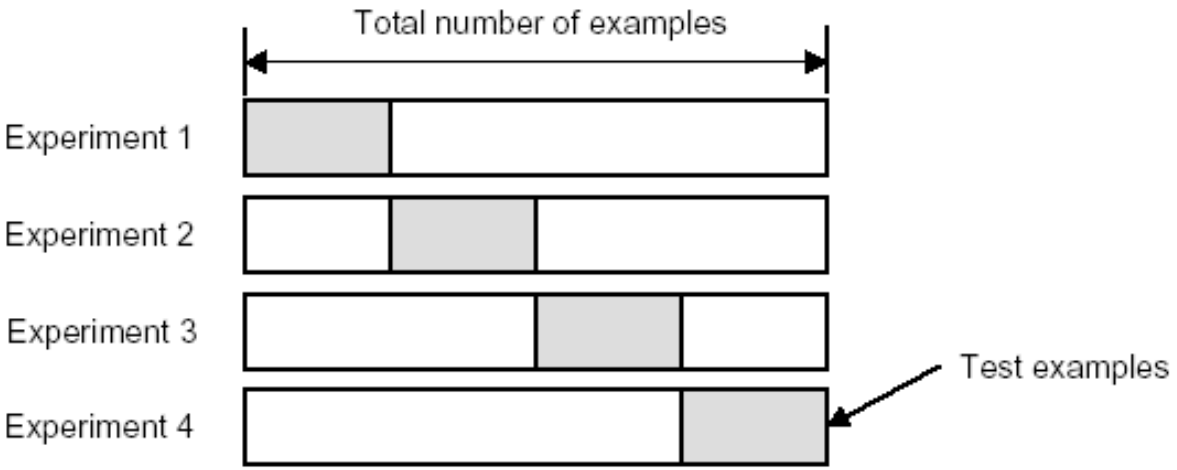


- **The true error estimate is obtained as the average of the separate estimates E_i**
  - This estimate is significantly better than the holdout estimate

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

K-fold cross validation

- **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
  - This procedure is illustrated in the following diagram for K=4

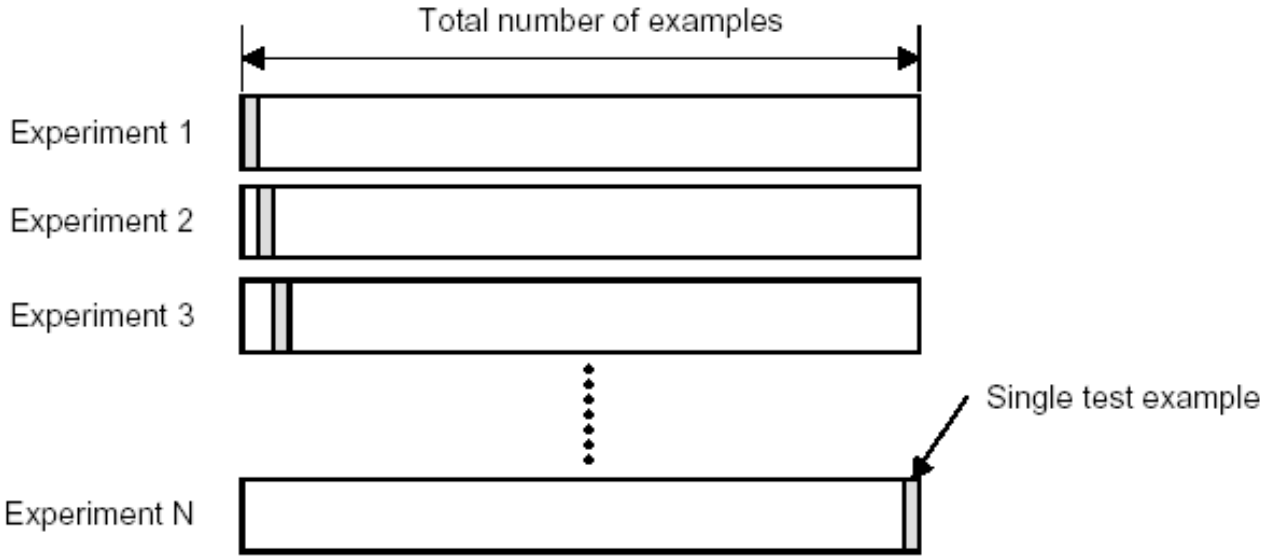


- **K-Fold Cross validation is similar to Random Subsampling**
  - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- **As before, the true error is estimated as the average error rate on test examples**

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

Leave-one-out

- **Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples**
  - For a dataset with N examples, perform N experiments
  - For each experiment use N-1 examples for training and the remaining example for testing



- **As usual, the true error is estimated as the average error rate on test examples**

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

How to measure classifier performances

- Confusion matrix
- Accuracy
- Precision, recall, specificity
- Precision at a given recall
- F-measure
- ROC and AUC

Confusion matrix

Consider a two-class classification problem:

- True positives (TP): positive examples correctly classified as positives
- True Negatives (TN): negative examples correctly classified as negatives
- False positives (FP): negative examples wrongly classified as positives
- False negatives (FN): positive examples wrongly classified as negatives

Predicted	True: Positives	True: Negatives
Positives	TP	FP
Negatives	FN	TN

Accuracy, precision, recall, F-score

Predicted \ True	Positives	Negatives
Positives	TP	FP
Negatives	FN	TN

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Prec = \frac{TP}{TP + FP} \qquad Rec = \frac{TP}{TP + FN}$$

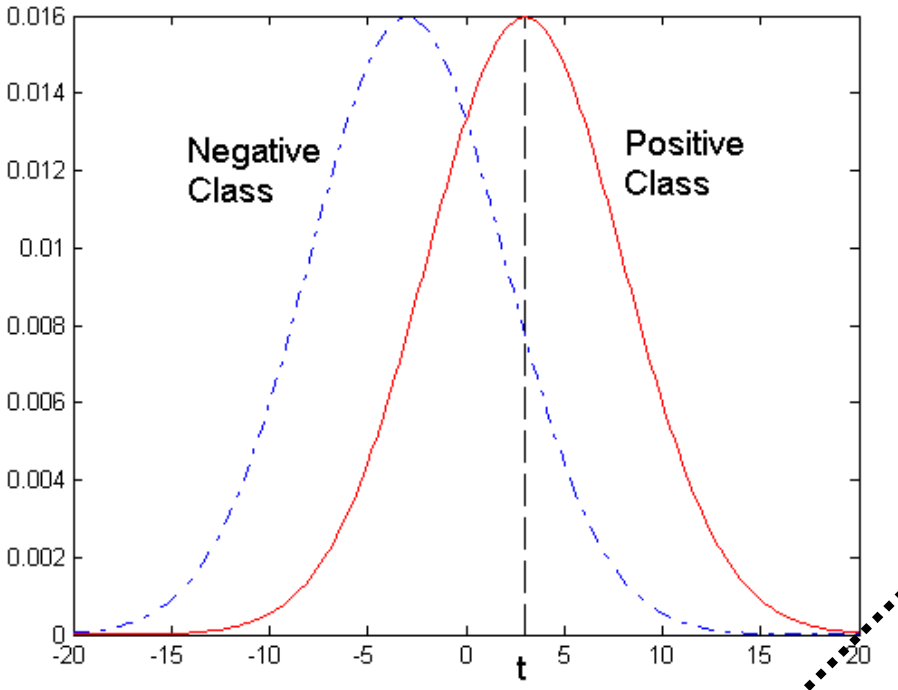
$$F - score = \frac{1}{\frac{1}{2}\left(\frac{1}{Prec} + \frac{1}{Rec}\right)} = \frac{2 * Prec * Rec}{Prec + Rec}$$

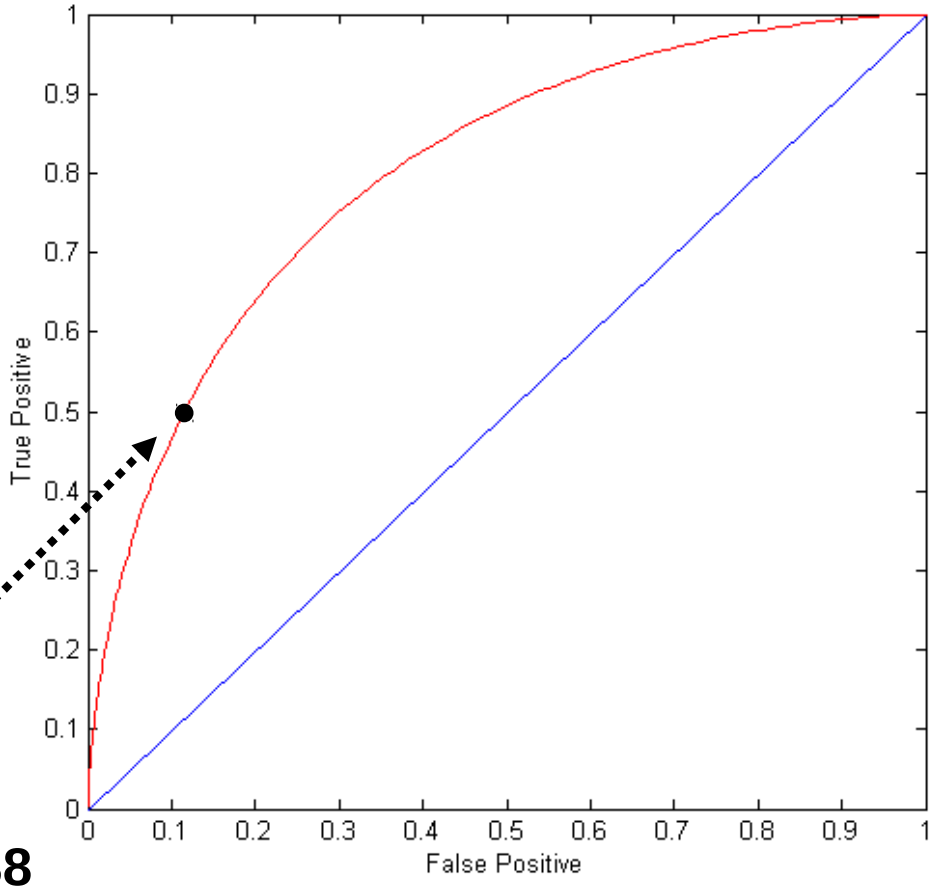
ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TP (on the y-axis) against FP (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
  - changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point

ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)
- any points located at x > t is classified as positive





At threshold t:

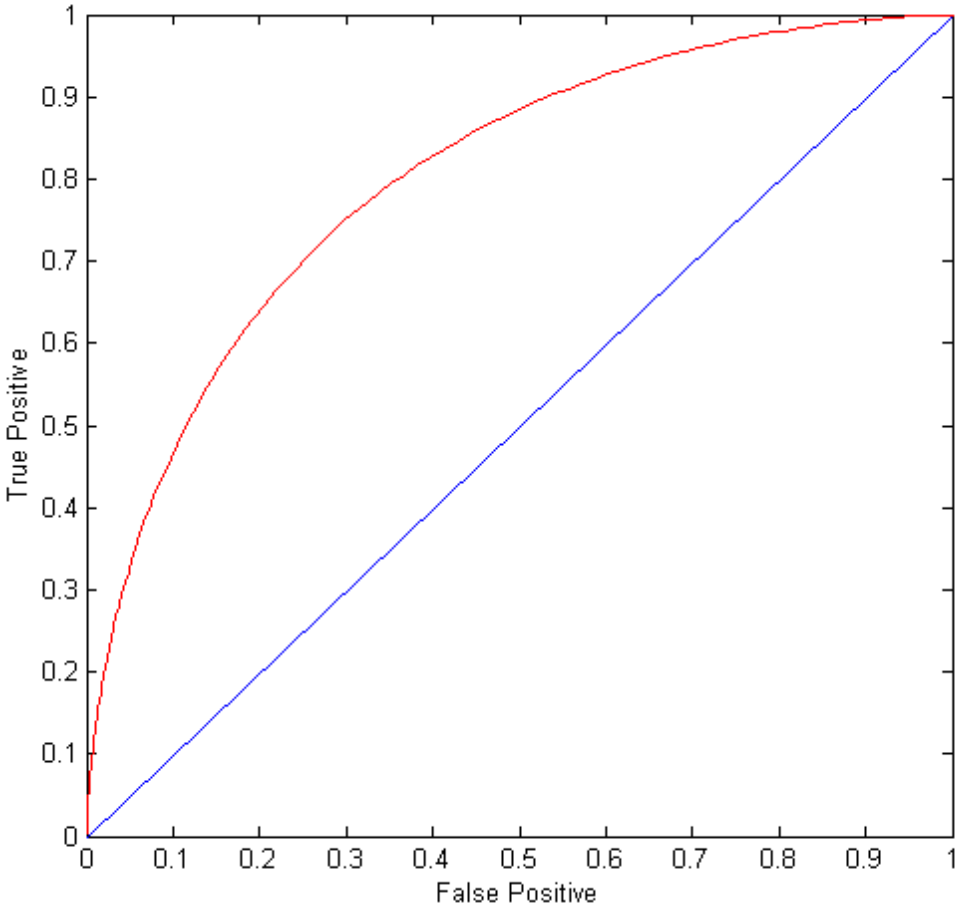
TP=0.5, FN=0.5, FP=0.12, FN=0.88

ROC Curve

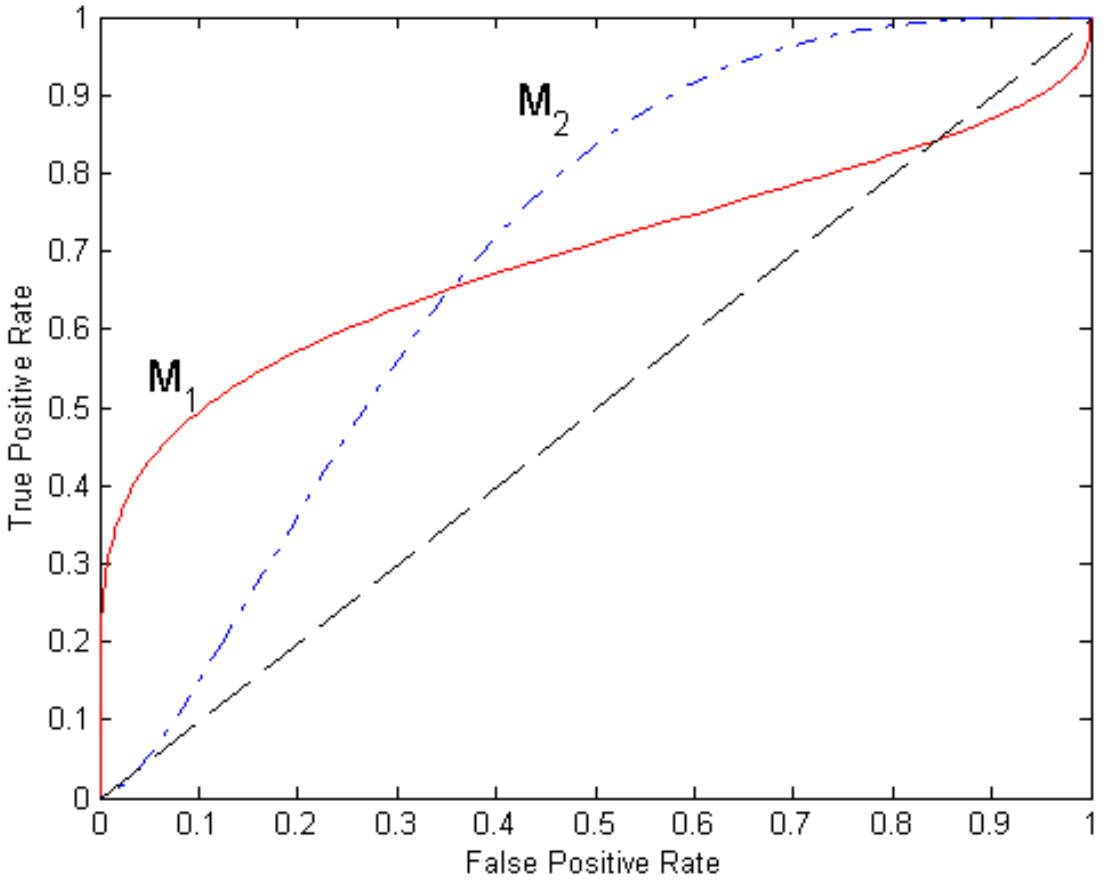
(TP,FP):

- (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



Using ROC for Model Comparison



- No model consistently outperform the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR
- Area Under the ROC curve
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5