# Machine learning

## Supervised learning

1

## Supervised Learning: Important Concepts

- Data: labeled instances <$x$, $y$>, e.g. emails marked spam/not spam
  - Training Set
  - Held-out Set
  - Test Set
- Features: attribute-value pairs which characterize each $x$
- Experimentation cycle
  - Learn parameters (e.g. model probabilities) on training set
  - (Tune hyper-parameters on held-out set)
  - Compute accuracy of test set
  - Very important: never "peek" at the test set!
- Evaluation
  - Accuracy: fraction of instances predicted correctly
- Overfitting and generalization
  - Want a classifier which does well on test data
  - Overfitting: fitting the training data very closely, but not generalizing well

2

# Example: Spam Filter

Input: email
Output: spam/ham
Setup:
- Get a large collection of example emails, each labeled "spam" or "ham"
- Note: someone has to hand label all this data!
- Want to learn to predict labels of new, future emails

Features: The attributes used to make the ham / spam decision
- Words: FREE!
- Text Patterns: $dd, CAPS
- Non-text: SenderInContacts
- …

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

# Example: Digit Recognition

Input: images / pixel grids
Output: a digit 0-9
Setup:
- Get a large collection of example images, each labeled with a digit
- Note: someone has to hand label all this data!
- Want to learn to predict labels of new, future digit images

Features: The attributes used to make the digit decision
- Pixels: (6,8)=ON
- Shape Patterns: NumComponents, AspectRatio, NumLoops
- …


0


1


2


1


??

# Classification Examples

- In classification, we predict labels *y* (classes) for inputs *x*
- Examples:
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Fraud detection (input: account activity, classes: fraud / no fraud)
  - Customer service email routing
  - Recommended articles in a newspaper, recommended books
  - DNA and protein sequence identification
  - Categorization and identification of astronomical images
  - Financial investments
  - …

5

# Supervised Learning

- Learning a discrete function: **Classification**
  - Boolean classification:
    - Each example is classified as true(positive) or false(negative).
- Learning a continuous function: **Regression**

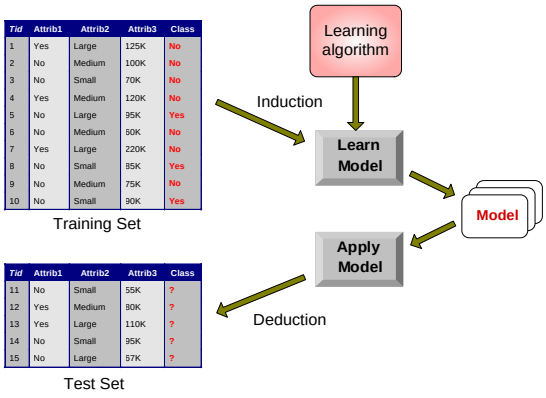
6

# Classification – A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label
  - The set of tuples used for model construction is training set
  - The model is represented as **classification rules**, **decision trees**, or **mathematical formulae**
- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - **Test set is independent of training set**, otherwise over-fitting will occur
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

7

# Illustrating Classification Task




| Tid | Attrib1 | Attrib2 | Attrib3 | Class |
|---|---|---|---|---|
| 1 | Yes | Large | 125K | No |
| 2 | No | Medium | 100K | No |
| 3 | No | Small | 70K | No |
| 4 | Yes | Medium | 120K | No |
| 5 | No | Large | 95K | Yes |
| 6 | No | Medium | 60K | No |
| 7 | Yes | Large | 220K | No |
| 8 | No | Small | 85K | Yes |
| 9 | No | Medium | 75K | No |
| 10 | No | Small | 90K | Yes |

Training Set

| Tid | Attrib1 | Attrib2 | Attrib3 | Class |
|---|---|---|---|---|
| 11 | No | Small | 55K | ? |
| 12 | Yes | Medium | 80K | ? |
| 13 | Yes | Large | 110K | ? |
| 14 | No | Small | 95K | ? |
| 15 | No | Large | 67K | ? |

Test Set

8

# Model validation



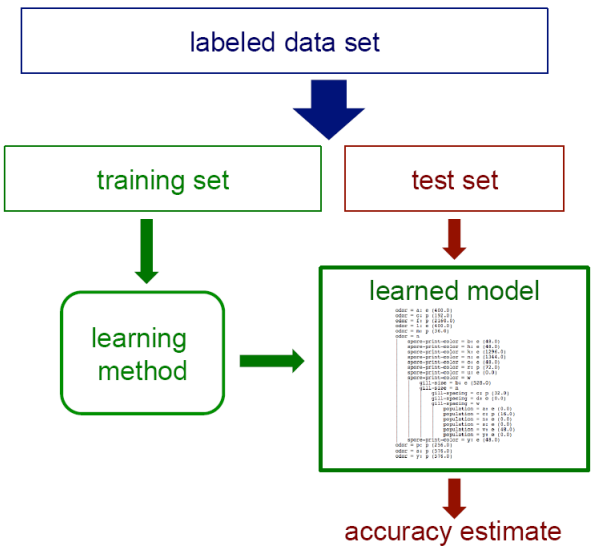
# Model validation

- How can we get an unbiased estimate of the accuracy of a learned model?
  - when learning a model, you should pretend that you don't have the test data
  - if the test set labels influence the learned model in any way, accuracy estimates will be biased

# Learning and testing

- When dealing with supervised learning algorithm, the key concept for validation is to build learning and test samples
- Classical methods will split the dataset with an 80% / 20% sampling after permutations

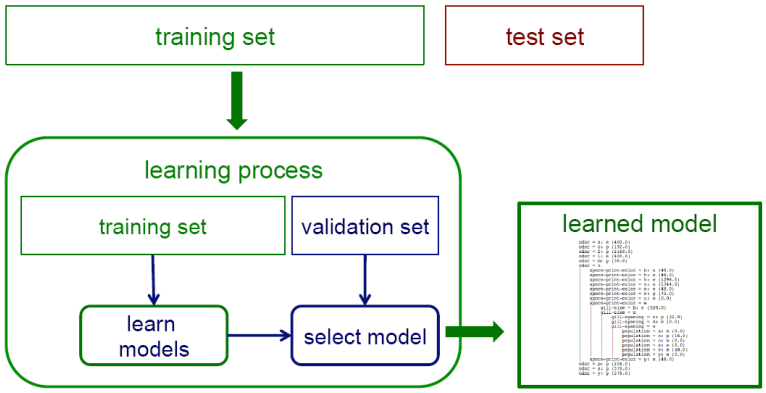
11




12

# Training / validation / test

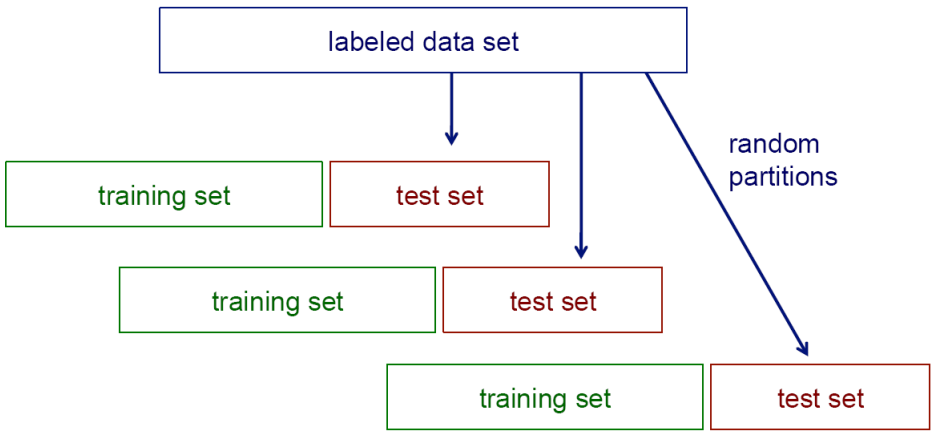
- We could need a supplementary validation set for estimating values of hyperparameters
- In that case, data are split in 3 sub-datasets




13

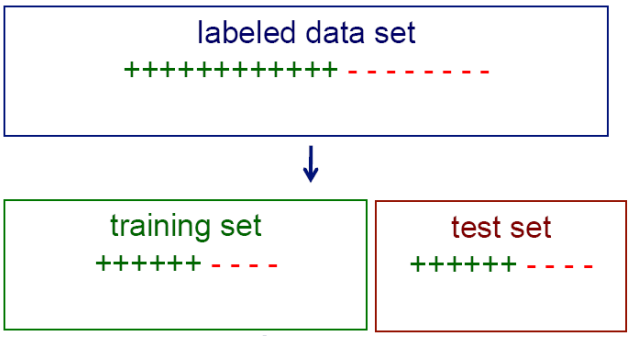
# Random sampling

- Another solution is random sampling, we can randomly split our data into training and test sets
- We would have many partitions



# Stratified sampling

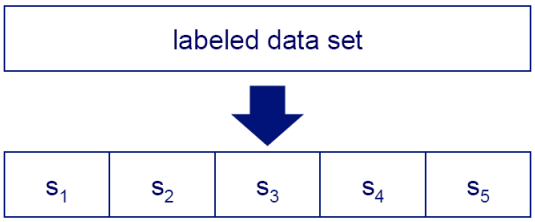
- If we are in a classification case, we can use stratified sampling in order to have the same proportion of + and - in each set




15

# Cross validation

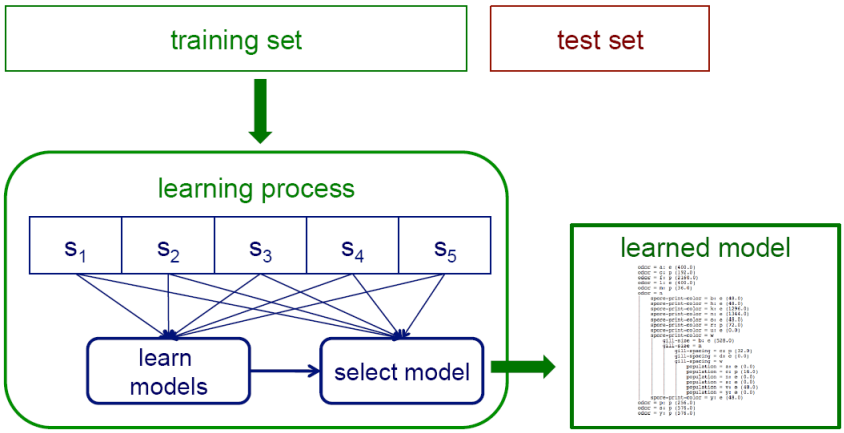
- Iteratively leave one sample out of the training set and use it for testing purposes
- The k-fold cross validation is used to validate your model and is more robust as classical train/test
- k should be chosen depending on different variables:
  - k will be larger when N is large
  - k will be smaller if training time is long
  - Values are usually between 5 and 10




| iteration | train on | test on |
|---|---|---|
| 1 | $s_2$ $s_3$ $s_4$ $s_5$ | $s_1$ |
| 2 | $s_1$ $s_3$ $s_4$ $s_5$ | $s_2$ |
| 3 | $s_1$ $s_2$ $s_4$ $s_5$ | $s_3$ |
| 4 | $s_1$ $s_2$ $s_3$ $s_5$ | $s_4$ |
| 5 | $s_1$ $s_2$ $s_3$ $s_4$ | $s_5$ |

# Cross validation

- We can use internal cross validation to set the model and still keep a test set for global testing




17

# Hyper parameters tuning

- Many machine learning algorithm have some hyperparameters and it is somehow difficult to fix their value
  - Number of neighbor in k-NN
  - Depth of a tree
  - Kernel in SVM
  - …
- The main method is grid search which consist of testing all possible combinations on a grid

18

# Hyper parameters tuning

- A search consists of:
  - an estimator (regressor or classifier such as sklearn.svm.SVC());
  - a parameter space;
  - a method for searching or sampling candidates;
  - a cross-validation scheme; and
  - a score function.

19

# Model validation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

20

# Some Metrics

- *Error Metrics for Regression Problems*
  - Mean Absolute Error, Weighted Mean Absolute Error, Root Mean Squared Error, Root Mean Squared Logarithmic Error
- *Error Metrics for Classification Problems*
  - Logarithmic Loss, Mean F Score, Multi Class Log Loss, Hamming Loss, Mean Utility
- *Error Metrics for probability distribution function*
  - Continuous Ranked Probability Score
- *Metrics*
  - Area Under Curve (AUC), Gini, Average Among Top P, Average Precision, Mean Average Precision
- *Other and rarely used:*
  - Average Precision, Absolute Error

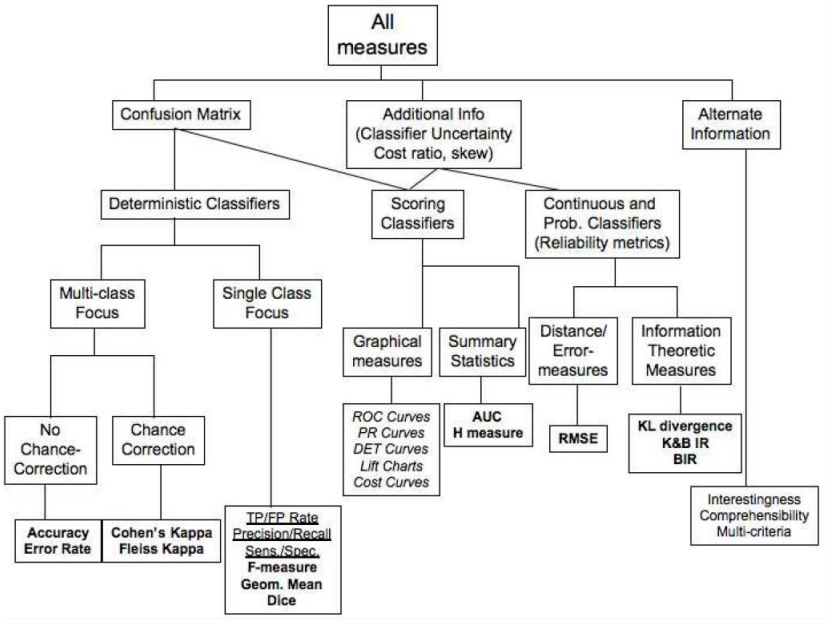
21

# Metrics available

- Here ae the results of metrics on the UCI Breast Cancer dataset

http://mlr.cs.umass.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

| Algo | Acc | RMSE | TPR | FPR | Prec | Rec | F | AUC | Info S |
|---|---|---|---|---|---|---|---|---|---|
| NB | 71.7 | .4534 | .44 | .16 | .53 | .44 | .48 | .7 | 48.11 |
| C4.5 | 75.5 | .4324 | .27 | .04 | .74 | .27 | .4 | .59 | 34.28 |
| 3NN | 72.4 | .5101 | .32 | .1 | .56 | .32 | .41 | .63 | 43.37 |
| Ripp | 71 | .4494 | .37 | .14 | .52 | .37 | .43 | .6 | 22.34 |
| SVM | 69.6 | .5515 | .33 | .15 | .48 | .33 | .39 | .59 | 54.89 |
| Bagg | 67.8 | .4518 | .17 | .1 | .4 | .17 | .23 | .63 | 11.30 |
| Boost | 70.3 | .4329 | .42 | .18 | .5 | .42 | .46 | .7 | 34.48 |
| RanF | 69.23 | .47 | .33 | .15 | .48 | .33 | .39 | .63 | 20.78 |




23

# 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:

| Confusion Matrix | | Target: Positive | Target: Negative | | |
|---|---|---|---|---|---|
| Model | Positive | a | b | *Positive Predictive Value* | a/(a+b) |
| | Negative | c | d | *Negative Predictive Value* | d/(c+d) |
| | | *Sensitivity* | *Specificity* | **Accuracy** = (a+d)/(a+b+c+d) | |
| | | a/(a+c) | d/(b+d) | | |

**a: TP (true positive)**
**b: FN (false negative)**
**c: FP (false positive)**
**d: TN (true negative)**

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

24

# Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

25

# Cost Matrix

| | | PREDICTED CLASS | |
|---|---|---|---|
| | C(i\|j) | Class=Yes | Class=No |
| ACTUAL CLASS | Class=Yes | C(Yes\|Yes) | C(No\|Yes) |
| | Class=No | C(Yes\|No) | C(No\|No) |

C(i|j): Cost of misclassifying class j example as class i

26

# Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

| Confusion Matrix | | Target: Positive | Target: Negative | | |
|---|---|---|---|---|---|
| Model | Positive | a | b | *Positive Predictive Value* | a/(a+b) |
| | Negative | c | d | *Negative Predictive Value* | d/(c+d) |
| | | *Sensitivity* | *Specificity* | **Accuracy** = (a+d)/(a+b+c+d) | |
| | | a/(a+c) | d/(b+d) | | |

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

27

# Other measures

- RMSE
  - The square root of the mean/average of the square of all of the error.
  - The use of RMSE is very common and it makes an excellent general purpose error metric for numerical predictions.
  - Compared to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors.

- RMSLE
  - RMSLE per $\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2}$ eater than an over-predicted estimate

28

# Other measures

- Mean F score: based on recall and precision

$$F1 = 2\frac{pr}{p+r}$$

29

# Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

30

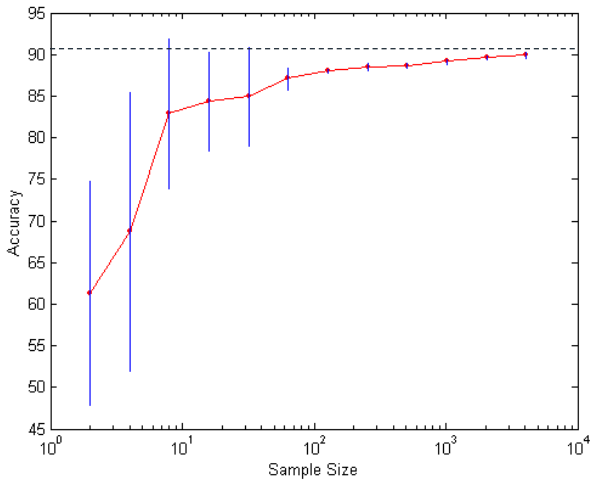
## Methods for Performance Evaluation

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

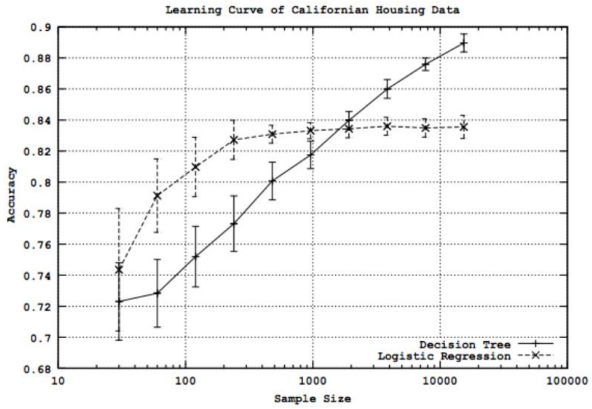
31

## Learning Curve

- Learning curve shows how accuracy changes with varying sample size
- Effect of small sample size:
  - Bias in the estimate
  - Variance of estimate



# Learning Curve




33

# Methods of Estimation

- Holdout
  - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
  - Repeated holdout
- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n
- Bootstrap
  - Sampling with replacement

34

# Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

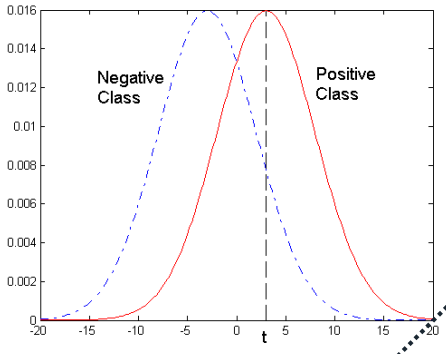
35

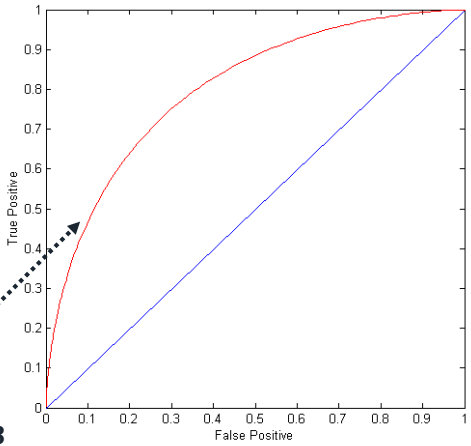
# ROC (Receiver Operating Characteristic)

- 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

36

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




**At threshold t:**

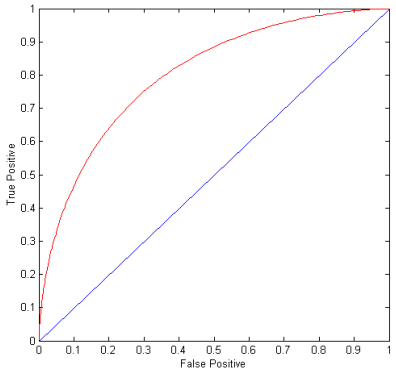
**TP=0.5, FN=0.5, FP=0.12, FN=0.88**

# ROC Curve

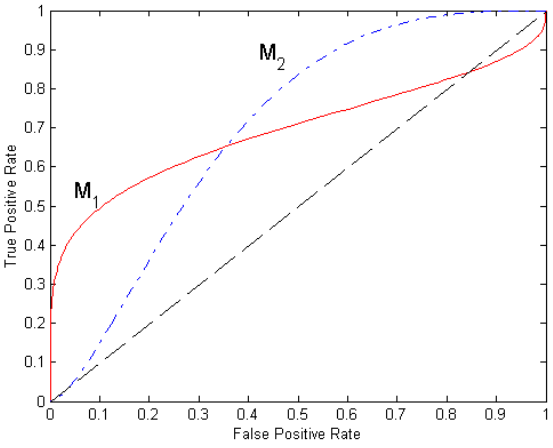
(TP,FP):

- (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



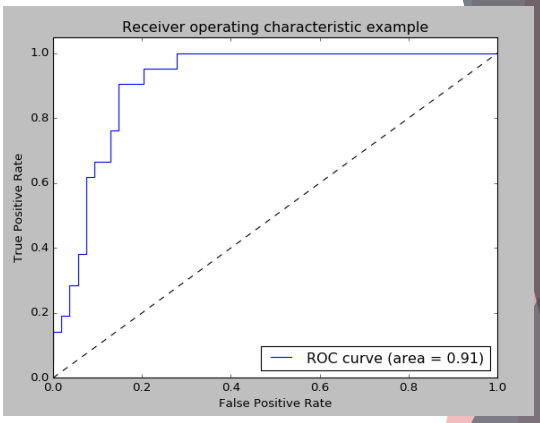

- No model consistently outperform the other
  - $M_1$ is better for small FPR
  - $M_2$ is better for large FPR
- Area Under the ROC curve
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

39

# Obtaining ROC cruve with python

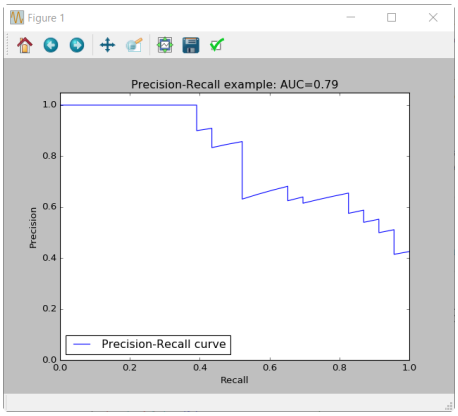
- Scikit-learn has funtions for ROC curve and Area Under Curve (AUC)
- You should use:

```
from sklearn.metrics import roc_curve, auc
```



# Precision / recall curve

- A *precision/recall curve* plots the precision vs. recall (TP-rate) as a threshold on the confidence of an instance being positive is varied
- You can use `precision_recall_curve()` and `average_precision_score()` functions from `sklearn.metrics`




# Strenghts

- both
  - allow predictive performance to be assessed at various levels of confidence
  - assume binary classification tasks
  - sometimes summarized by calculating *area under the curve*
- ROC curves
  - insensitive to changes in class distribution (ROC curve does not change if the proportion of positive and negative instances in the test set are varied)
  - can identify optimal classification thresholds for tasks with differential misclassification costs
- precision/recall curves
  - show the fraction of predictions that are false positives
  - well suited for tasks with lots of negative instances

42

## Comparing models with tests

- You can compare models using statistical testing
- Using the null hypothesis that suppose models are not different
- Using cross validation you will obtain as many score as folds for each model
- Using non parametric comparison test will lead to statistical significance of the difference

- In that case, we suppose that scores are randomly sampled from the population of all possible scores associated to the models

43

## Lets practice with python

44