

# Machine learning

Prepared by : Dr. Hanaa Bayomi  
Updated By: Prof Abeer ElKorany



## Lecture 12: Feature and model selection

# Model

- How many features to have in the model
- Prediction accuracy vs model interpretability

Less number of features	More number of features
Easy to interpret	Difficult to interpret
Less likely to over fit	More likely to over fit
Low prediction accuracy	High prediction accuracy

# Feature selection

**Performance of Machine Learning model depend on**

- Choice of algorithm
- Feature selection
- Feature creation
- Model selection

**Top reasons to use feature selection are:**

- It enables the machine learning algorithm to **train faster**.
- It **reduces the complexity of a model** and makes it easier to interpret.
- It **improves the accuracy of a model** if the right subset is chosen.
- It **reduces overfitting**.

# Classification of FS methods

- **Filter (single factor analysis)**

- Assess the relevance of features only by looking at the essential properties of the data.
- Usually, calculate **the feature relevance score** and remove low-scoring features.

- **Wrapper**

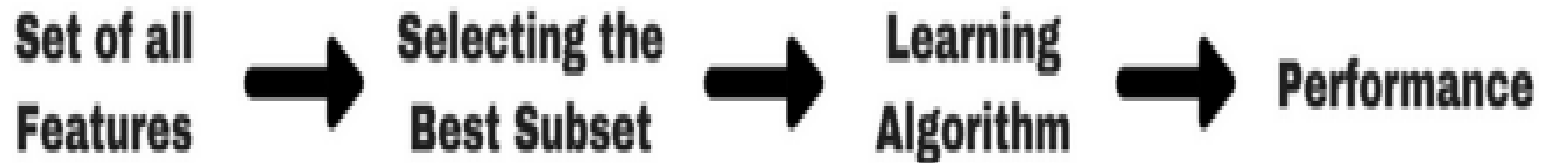
- Bundle the search for best model with the FS.
- Generate and evaluate various subsets of features. The evaluation is obtained by training and testing a specific ML model.

- **Embedded**

- Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods.

# Filter methods

- Filter methods are generally used as a *preprocessing step*. The selection of features is independent of any machine learning algorithms.
- Two steps (score-and-filter approach)
  1. assess each feature individually for its potential in discriminating among classes in the data
  2. features falling beyond threshold are eliminated



# Filter methods

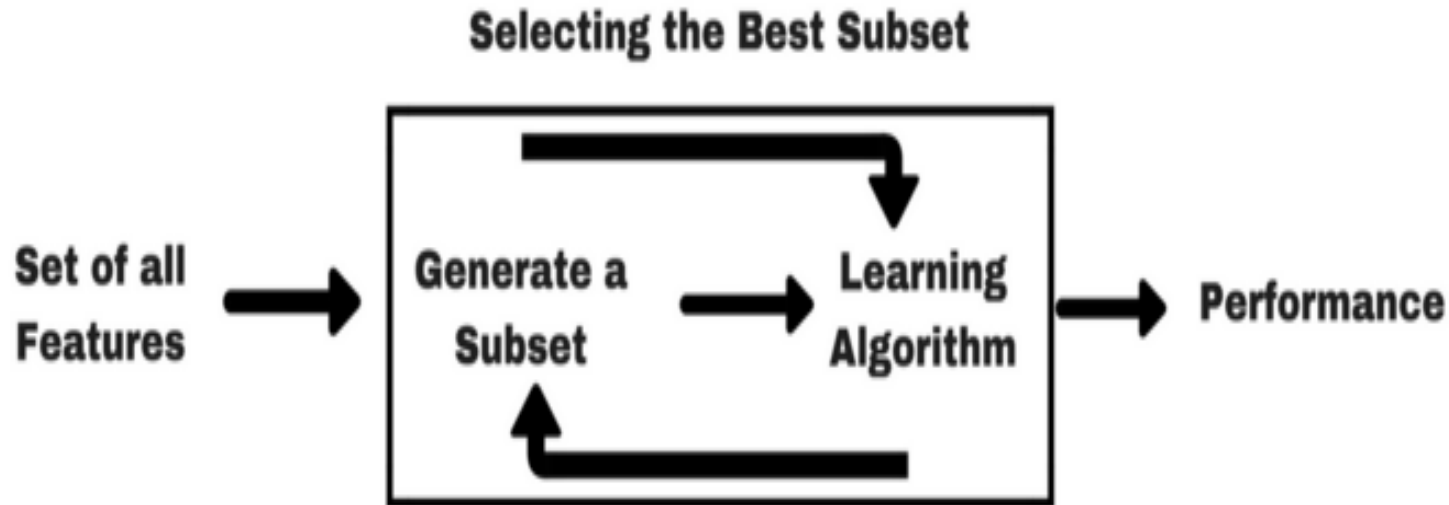
- **Advantages:**
  - easily scale to high-dimensional data
  - simple and fast
  - independent of the classification algorithm
- **Disadvantages:**
  - ignore the interaction with the classifier
  - most techniques are uni variate (each feature is considered separately)

# Scores in filter methods

- **Distance measures**
  - Euclidean distance
- **Dependence measures**
  - Pearson correlation coefficient
  - Chi-Square: It is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.
- **Information measures**
  - information gain
  - mutual information

# Wrappers

- Search for the best feature subset in combination with a fixed classification method.
- The goodness of a feature subset is determined using cross-validation ( $k$ -fold, LOOCV) Leave-one-out cross-validation





# Wrappers

- Advantages:
  - interaction between feature subset and model selection
  - take into account feature dependencies
  - generally more accurate
- Disadvantages:
  - higher risk of overfitting than filter methods
  - very computationally intensive

# Wrappers Methods Examples

- **Subset Selection:** fit model with each possible combination of  $P$  features
  - Total number of models  $2^P - 1$
  - if  $P=2$  (Lets say  $X_1, X_2$ )
    - $Y = B_0 + B_1 X_1$
    - $Y = C_0 + C_1 X_2$
    - $Y = D_0 + D_1 X_1 + D_2 X_2$
- If  $P > 20$  it is almost impossible to use subset selection method

# Wrappers Methods Examples

- **Forward Selection:** Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.
- **Backward Elimination:** In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

# WRAPPERS METHODS EXAMPLES

## Forward selection

Initialize  $s = \{\}$

Do:

    Add feature to  $s$

    which improves  $K(s)$  most

While  $K(s)$  can be improved

## Backward elimination

Initialize  $s = \{1, 2, \dots, n\}$

Do:

    remove feature from  $s$

    which improves  $K(s)$  most

While  $K(s)$  can be improved

	Backward	forward
	X1 X2 X3 X4 X5	X1
	X1 X3 X4 X5	X1 X2
	X1 X3 X5	X1 X2 X4
	X1 X3 X5	X1 X2 X4 X5
	X1 X3 X5	X1 X2 X4 X5

# WRAPPERS METHODS EXAMPLES

## Forward selection

Initialize  $s = \{\}$

Do:

    Add feature to  $s$   
    which improves  $K(s)$  most

While  $K(s)$  can be improved

## Backward elimination

Initialize  $s = \{1, 2, \dots, n\}$

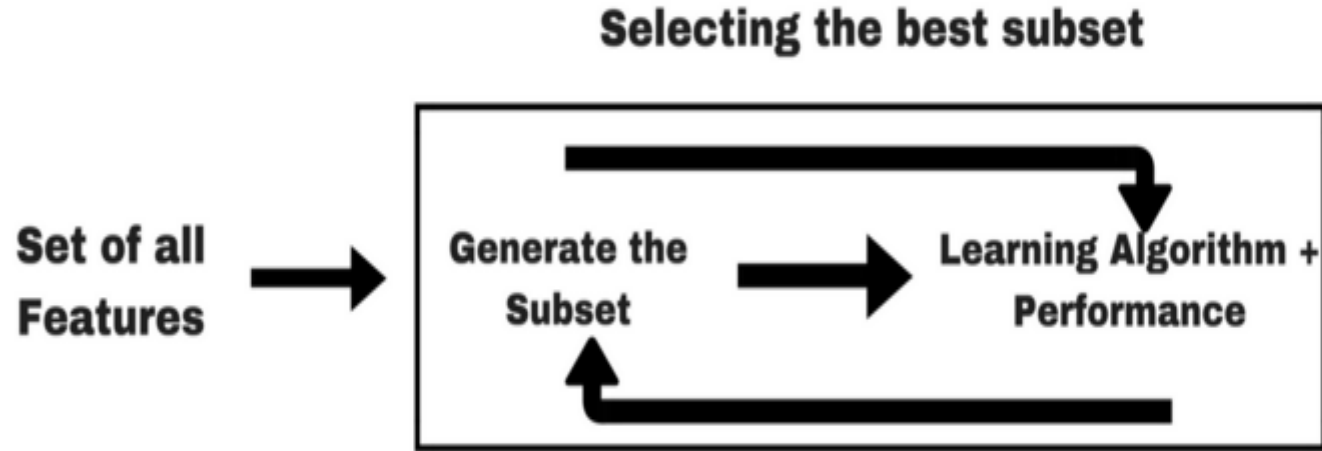
Do:

    remove feature from  $s$   
    which improves  $K(s)$  most

While  $K(s)$  can be improved

- Backward elimination tends to find better models
  - Better at finding models with interacting features
  - But it is frequently too expensive to fit the large models at the beginning of search

# EMBEDDED



Some of the most popular examples of these methods are LASSO and RIDGE regression which have inbuilt penalization functions to reduce overfitting.

**Lasso regression** performs L1 regularization which adds penalty equivalent to absolute value of the magnitude of coefficients.

**Ridge regression** performs L2 regularization which adds penalty equivalent to square of the magnitude of coefficients.

- How can you tell that your model is **overfitting** or **underfitting** the data?

- 1 **Plotting hypothesis**: is one way to decide – you may look for “curvy – waggle”
  - But does not always work
  - Often have lots of features – harder to plot the data and visualize what feature to keep and which one to throw out
- 2 **cross-validation metrics**: If a model **performs well on the training data** but **generalizes poorly** according to the **cross-validation metrics**, then your model is **overfitting**.
  - If it **performs poorly on both**, then it is **underfitting**.
- 3 **Learning curves**: Another way is to look at the ***learning curves***:
  - these are plots of the **model's performance on the training set and the validation set** as a **function of the training set size**
  - To generate the plots, simply train the model several times on different sized subsets of the training set

## ■ Addressing overfitting

- ≡ If you have lots of features and little data – overfitting can be a problem
- ≡ simple way to regularize a polynomial model is to reduce the number of polynomial degrees
- ≡ Reduce number of features:
  - Manually select which features to keep
  - Feature engineering – reduce numbers of features
  - But, we lose information
  - why do not we just stop adding features when we got an acceptable model!!!
  - You do not know which features to drop, and even worse if it turns out that every feature is fairly informative, which means that dropping some feature will likely ruin the algorithm performance – the answer is **regularization**
- ≡ **Regularization** : keep all features, but reduce magnitude of parameters  $\theta$



# Regularization

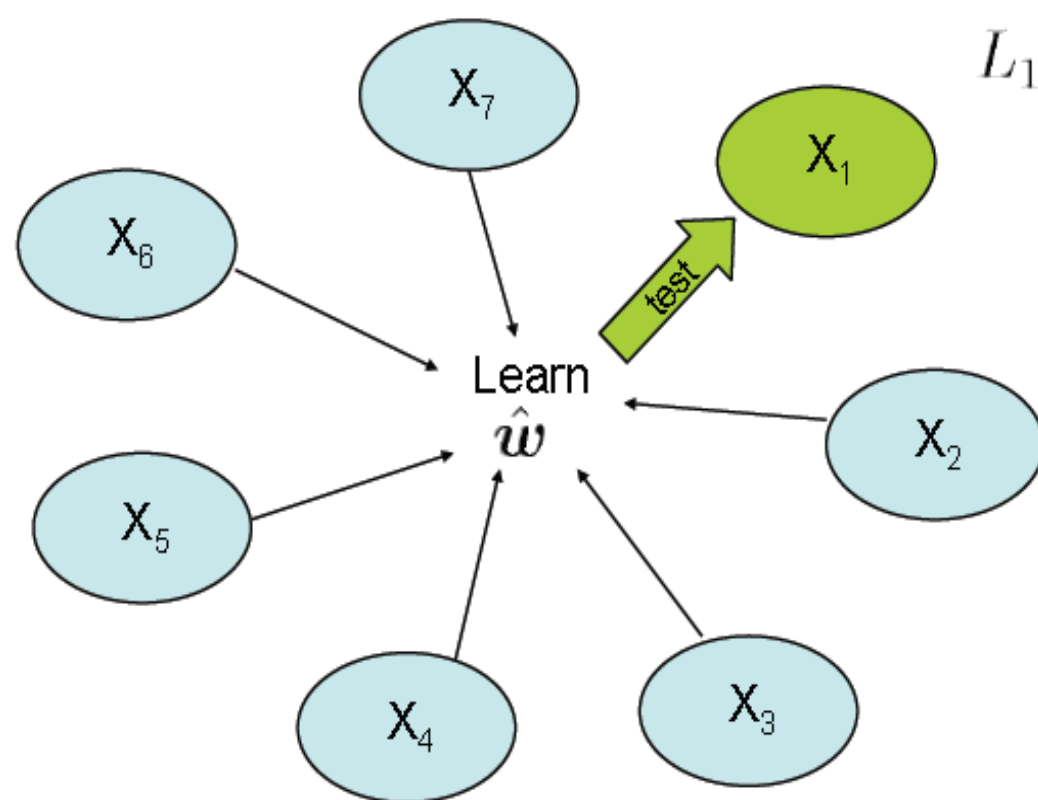
- ≡ Regularization is the process of **regularizing the parameters that constrain, regularizes, or shrinks** the coefficient estimates towards zero
- ≡ **Higher coefficient** of polynomial terms lead to **overfitting**
  - Having **large coefficients** can be seen as evidence of **memorizing data rather learning** from them
  - For example, if you have some noises in training dataset, those noises will cause our model to put more weight into the coefficient of higher degree, and this lead to overfitting
- ≡ with an **increasing number of features**, a machine learning model can **fit your data well** – **but** if you **add too many features** we will be subjected to **Overfitting**

# How to regularize?

- Smaller values for parameters corresponds to a simpler hypothesis
  - You get rid of some of the terms
- A simpler hypothesis is less prone to overfitting
- But, we do not know what are the high order terms
  - How do we choose the ones to shrink?
- It is better to shrink all parameters

# K-fold cross validation

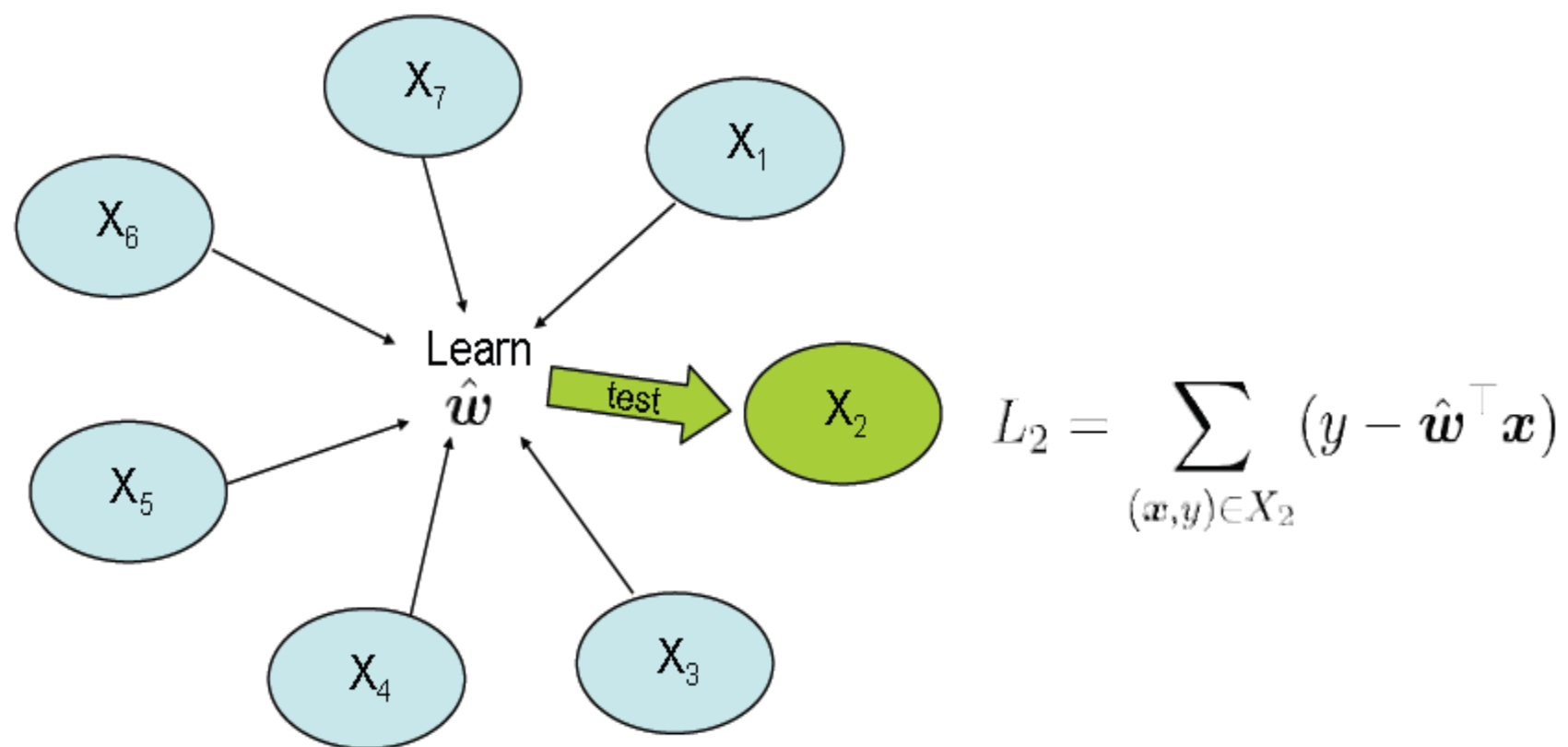
- A technique for estimating test error
- Uses *all* of the data to validate
- Divide data into K groups  $\{X_1, X_2, \dots, X_K\}$ .
- Use each group as a validation set, then average all validation errors



$$L_1 = \sum_{(\mathbf{x}, y) \in X_1} (y - \hat{\mathbf{w}}^\top \mathbf{x})$$

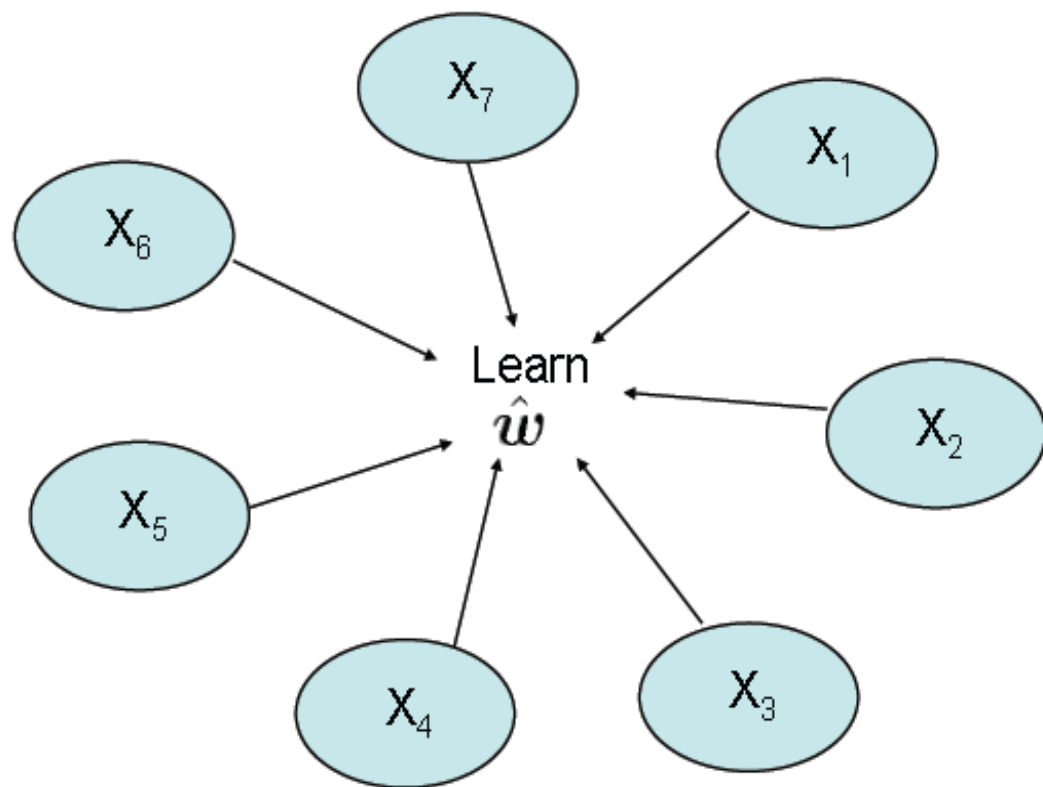
# K-fold cross validation

- A technique for estimating test error
- Uses *all* of the data to validate
- Divide data into K groups  $\{X_1, X_2, \dots, X_K\}$ .
- Use each group as a validation set, then average all validation errors



# K-fold cross validation

- A technique for estimating test error
- Uses *all* of the data to validate
- Divide data into K groups  $\{X_1, X_2, \dots, X_K\}$ .
- Use each group as a validation set, then average all validation errors



$$CV(s) = \frac{1}{K} \sum_{i=1}^K L_i$$

# MODEL EVALUATION

## Precision and Recall

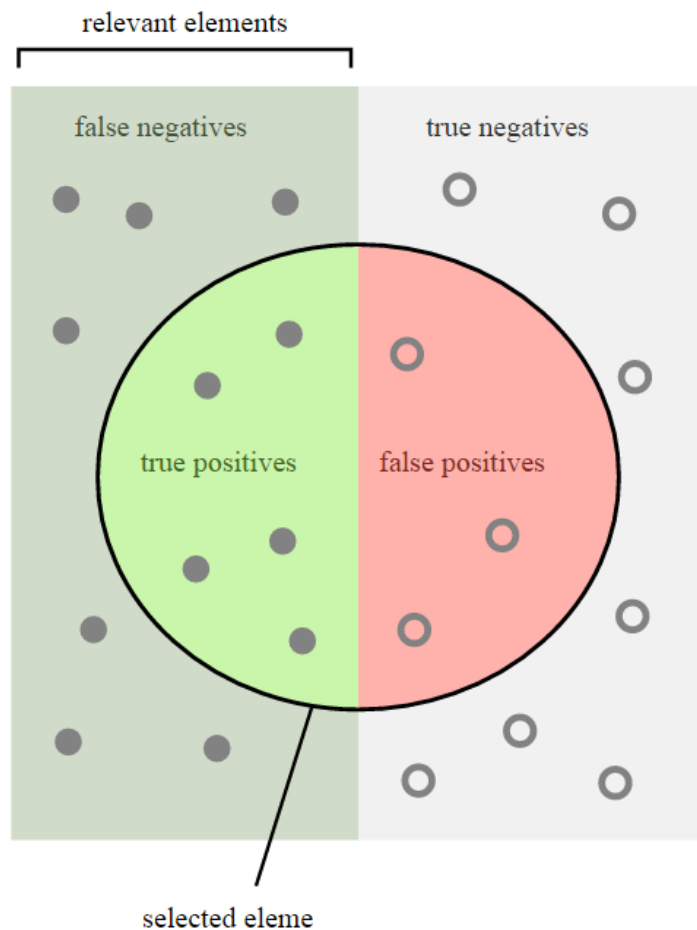
---

- **Precision:** fraction of output are relevant =  $P(\text{relevant} | \text{retrieved})$
- **Recall:** fraction of relevant output that are retrieved =  $P(\text{retrieved} | \text{relevant})$

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision  $P = tp / (tp + fp)$
- Recall  $R = tp / (tp + fn)$

# Precision and Recall Definition



How many selected  
items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant  
items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$F\_measure = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

# PROBLEM

Assume the following:

- A database contains **80 records** on a particular topic
- A search was conducted on that topic and **60 records** were retrieved.
- Of the 60 records retrieved, **45 were relevant**. Calculate the precision and recall scores for the search

## Solution:

- TP = The number of relevant records retrieved, 45
- SE = The number of retrieved records, 60
- FP = The number of irrelevant records retrieved. (60-45)

In this example

$$\text{Recall} = 45/80 * 100\% = 56\%$$

$$\text{Precision} = (45 / (45 + 15)) * 100\% \Rightarrow 45/60 * 100\% = 75\%$$

$$\text{F\_Measure} = (2(0.56 * 0.75) / (0.56 + 0.75)) * 100 = 64.12\%$$