

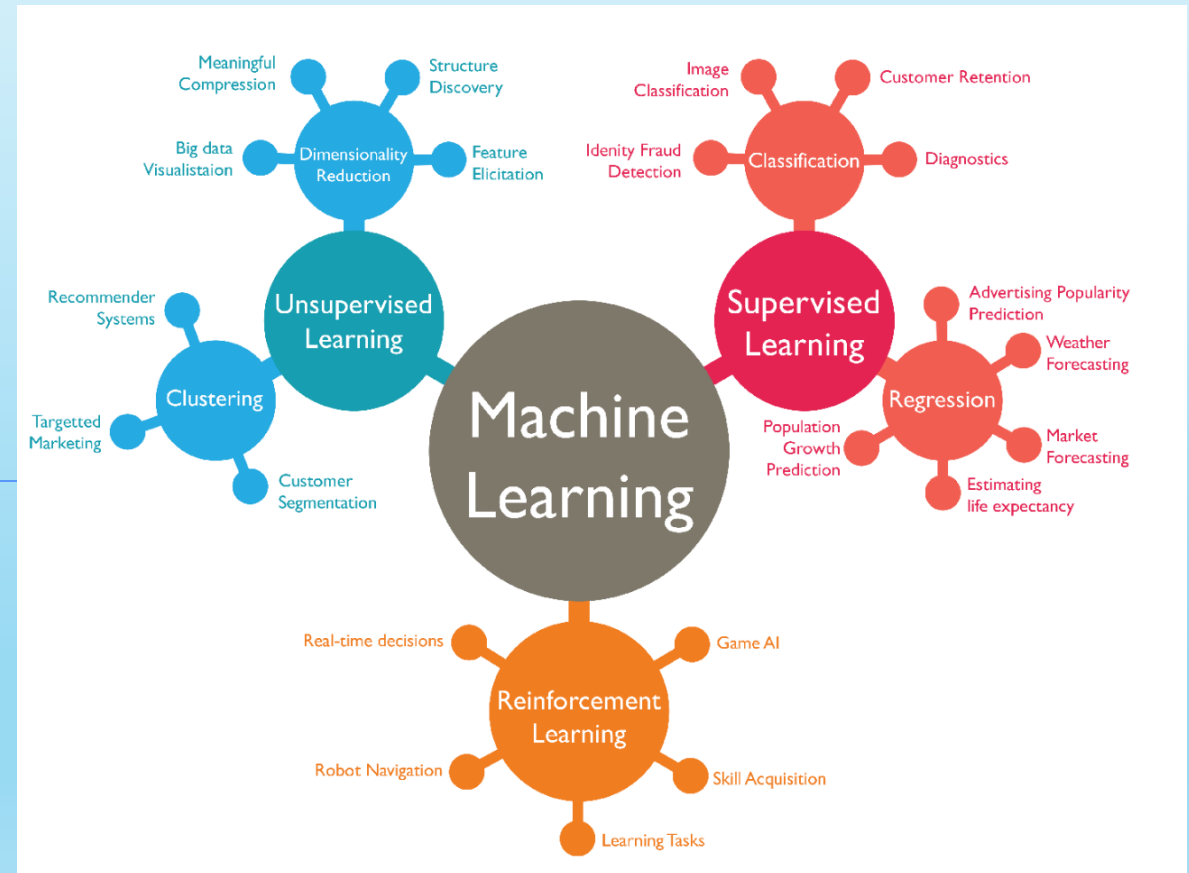
Introduction to Machine Learning

Annamaria Porreca

University of «G.d'Annunzio»,

Chieti-Pescara, Italy.

annamaria.porreca@unich.it



What is Machine Learning?

“Learning is any process by which a system improves performance from experience.”

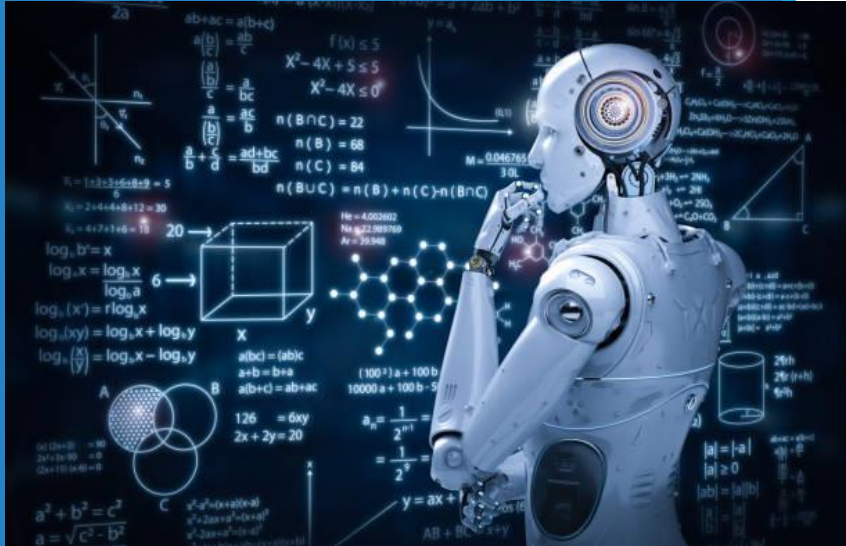
- Herbert Simon

Definition by Tom Mitchell (1998):

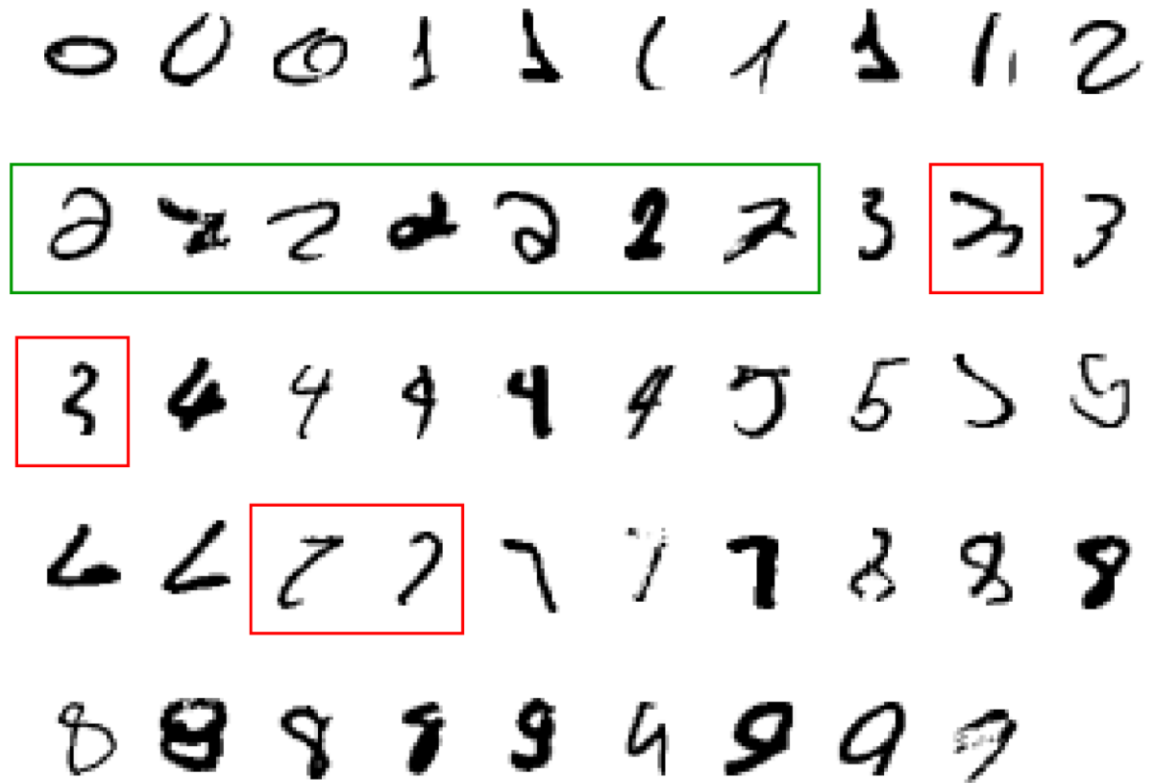
Machine Learning is the study of algorithms that

- improve their performance P
- at some task T
- with experience E .

A well-defined learning task is given by $\langle P, T, E \rangle$.



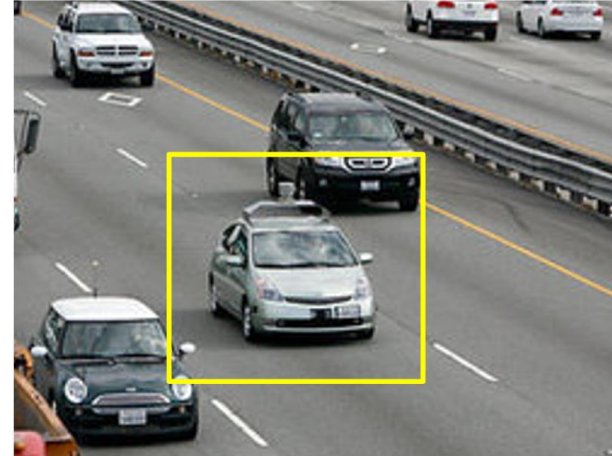
A classic example of a task that requires machine learning:
It is very hard to say what makes a 2



Slide credit: Geoffrey Hinton

6

Autonomous Cars



- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

Penn's Autonomous Car →
(Ben Franklin Racing Team)



Types of Learning

- **Supervised (inductive) learning**
 - Given: training data + desired outputs (labels)
- **Unsupervised learning**
 - Given: training data (without desired outputs)
- **Semi-supervised learning**
 - Given: training data + a few desired outputs
- **Reinforcement learning**
 - Rewards from sequence of actions

Supervised classification

When we know the number of groups because they are an objective fact and we have the availability of a training set with predictors and group variable(s).

The modalities of a grouping variable are called “Labels”.

SICK PERSON



HEALTHY PERSON



TRAINING SET

	GROUP	Predictor X1	Predictor X2
Little man	SICK
Arnold	HEALTHY
....	HEALTHY
....	SICK

Supervised classification

TRAINING SET

	GROUP	Variable X1	Variable X2
Little man	SICK
Arnold	HEALTHY
....	HEALTHY
....	SICK



With the use of a classifier, we find a classification rule



The final aim is to use this rule to classify future statistical units of which we do not know the group (but we know predictors' values)



	GROUP	Variable X1	Variable X2
Antonio	?
Pasquale	?
Erika	?
Chiara	?

Test set

	REAL GROUP	PREDICTED LABEL	Variable X1	Variable X2
Fabrizio	HEALTHY	HEALTHY
Carla	HEALTHY	HEALTHY
Erika	SICK	SICK
Chiara	HEALTHY	SICK



Error in predicting the label

Regardless of the main objective being to classify new statistical units, when we build a classifier, we are interested in whether this rule classifies well or not.

The concept of test set fits into this context.

The latter is a set of statistical units of which we know the value of the predictors and labels.

Applying the classification rule created on the training set we try to see if this classifies well the statistical units present in the test set.

Of course, when we apply the classification rule, we do not use the information on the labels of the test set. This information is used only at the end to understand if the classification was accurate.

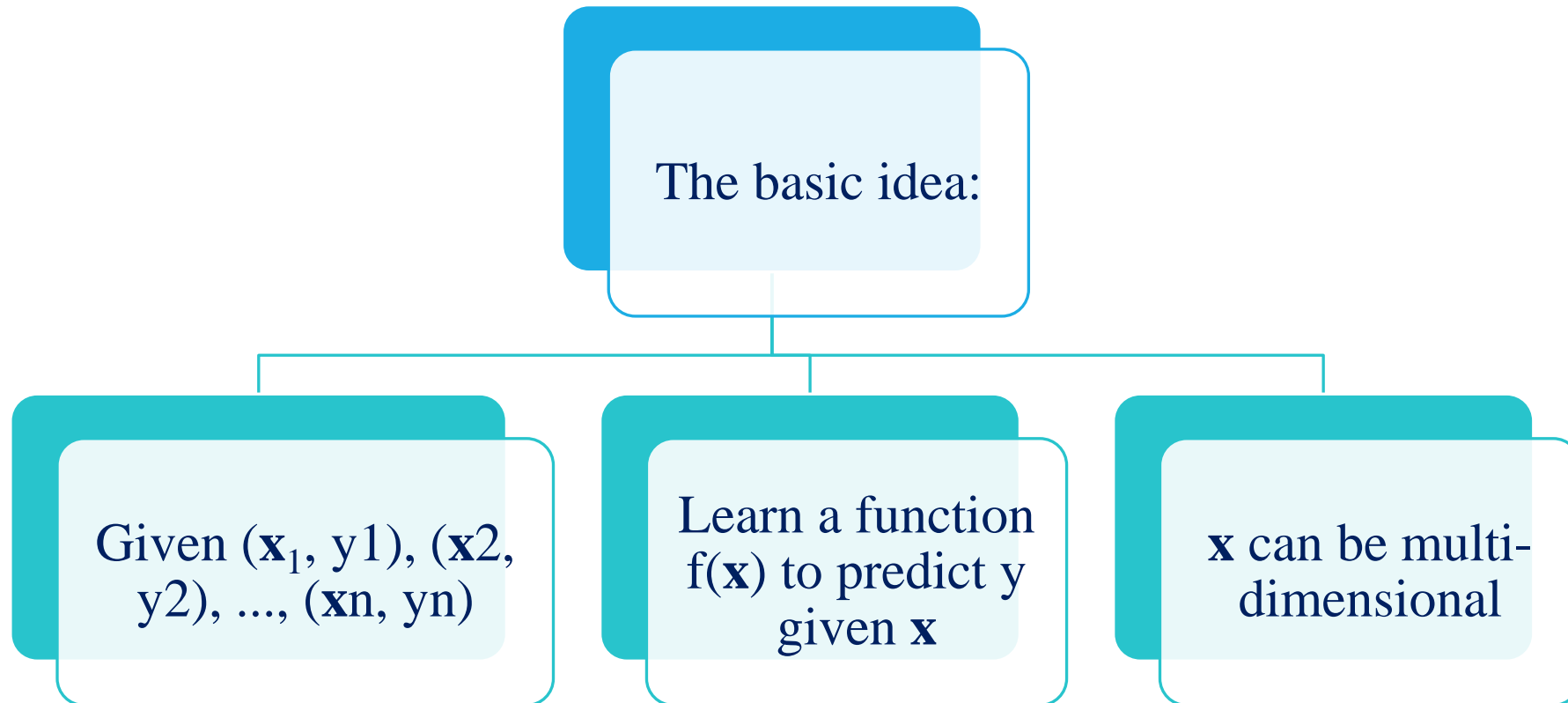
In this small introduction we do not consider the concept of validation set in order not to create confusion. When students will be confident with the concept of supervised classification, they will be able to understand the concept of validation set without making confusion.

Supervised classification

Some of the most used approaches are:

1. Logistic Regression (LR)
2. K-Nearest Neighbors (K-NN)
3. Linear Discriminant Analysis (LDA)
4. Bayesian Approach (BA)
5. Decision Trees (es. CART)
6. Bagging and Random Forest (RF)
7. Artificial Neural Networks (ANNs)
8. Support Vector Machines (SVMs)

Class prediction



Class prediction

Class prediction is a supervised learning method where the algorithm learns from a training set and establishes a prediction rule to classify new samples.

Examples:

1. to classify cancer types using gene expression profiling
2. to predict, based on protein expression profiles, which breast cancer patients will relapse within two years of diagnosis versus which will remain disease free
3. to predict, based on either genomics or proteomics data, which patients are likely to experience severe toxicity from a new drug versus which will tolerate it well.

Class prediction algorithm

Development of a class prediction algorithm generally consists of three steps:

1. selection of predictors
2. fitting the prediction model to develop the classification rule
3. performance assessment.

The first two steps build a prediction model, and the third step assesses the performance of the model.

Some classification algorithms, such as the classification tree or stepwise logistic regression, perform the first two steps simultaneously.

Note on class prediction algorithm in Biology



With genomic and proteomic or metabolomic data, the set of predictors is often large, while the sample size is relatively small.

In such a case, a model may capture the patterns in the training data exceptionally well.

Consequently, it may lead to a problem of overfitting the training data.

k-Nearest-Neighbours classification (K-NN)

The k-NN is a non-parametric procedure to classify data points by considering their closest k neighbours (obviously, based on a distance, e.g. the Euclidean distance.)

The k-NN classifier identifies the k observations of the training data set, which are the closest to the test observation of which we would understand the class label.

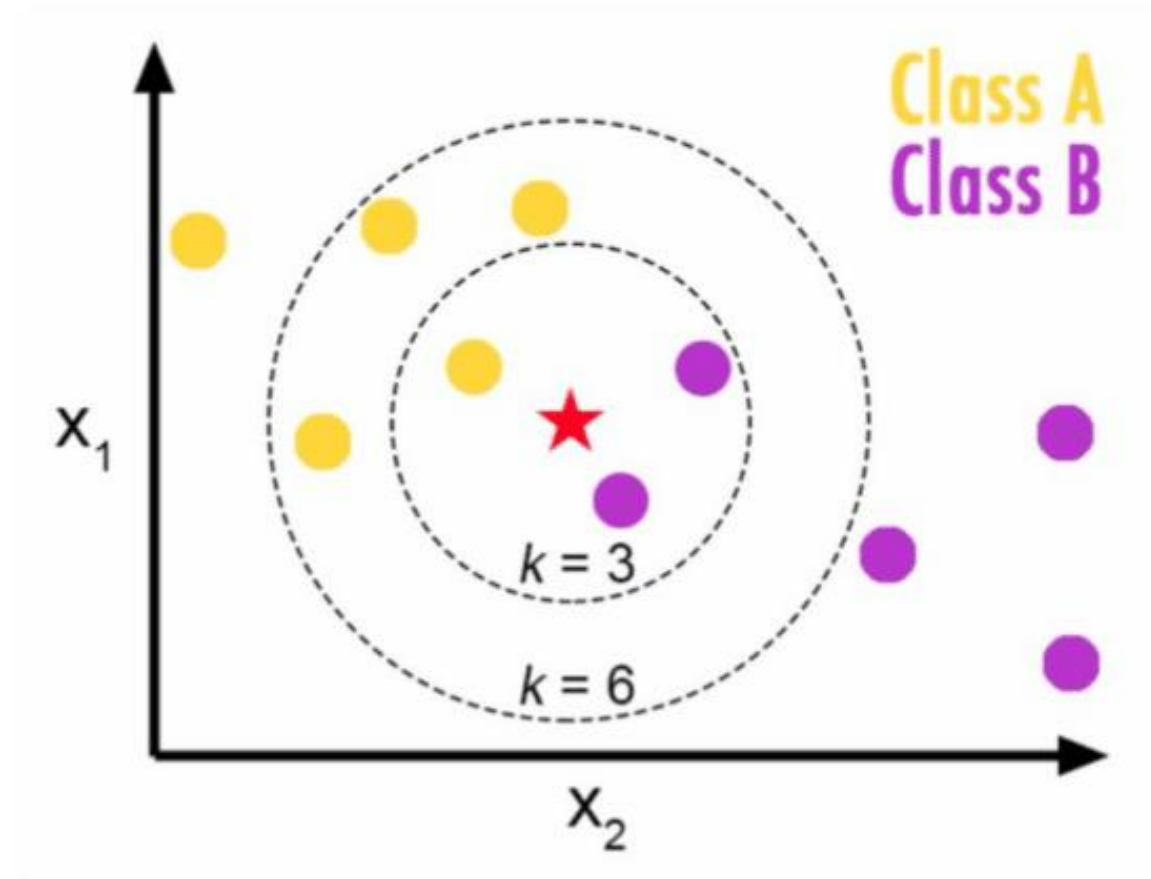
The class designation for this new observation is based on the majority vote of these k neighbours.

k-Nearest-Neighbours classifier of $P(Y|X = x_0)$

$$P(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j) .$$

As k increases, the variance of the classifiers will decrease, but the bias will be high.

K-NN meaning



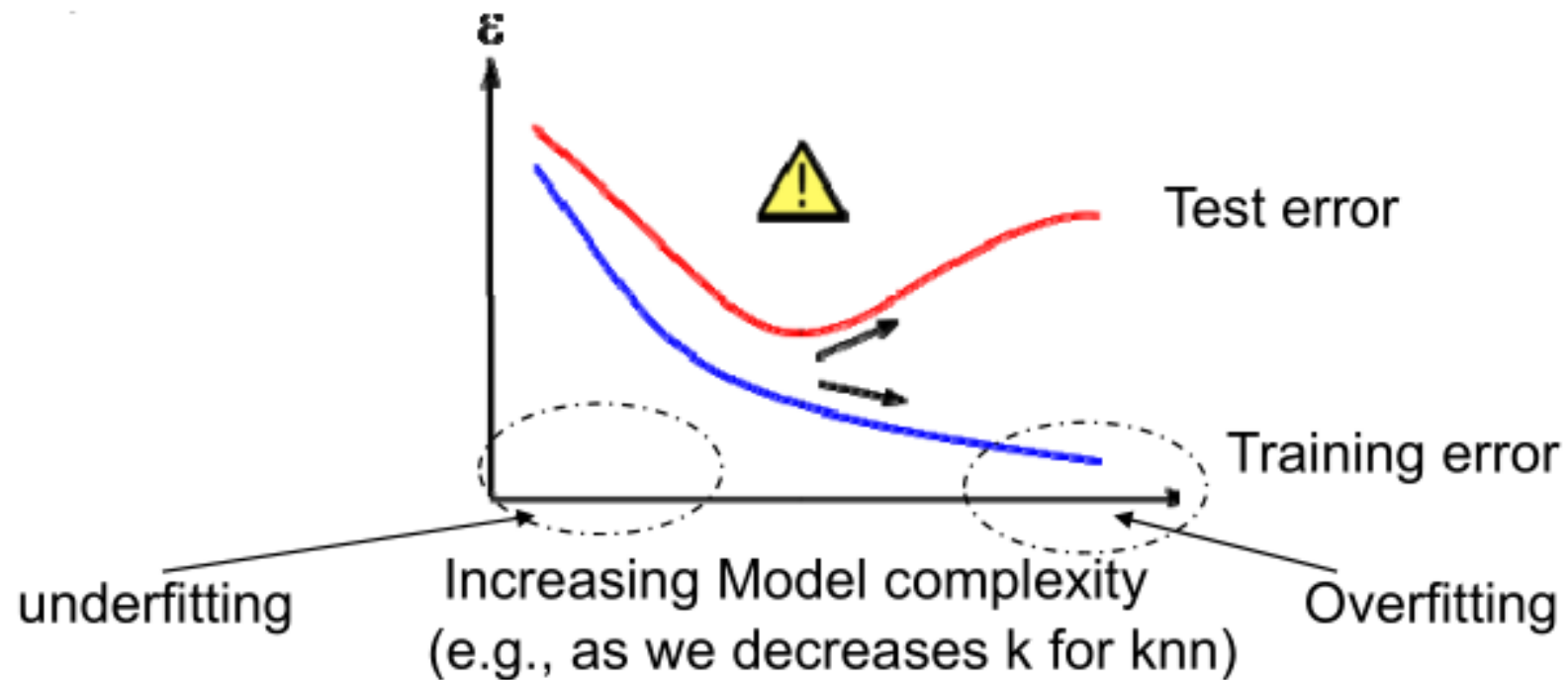
K-NN in R

knn() does predictions using a single command. The function requires four inputs:

1. A matrix containing the predictors associated with the training data, labeled train.X below.
2. A matrix containing the predictors associated with the data for which we wish to make predictions, labeled test.X below.
3. A vector containing the class labels for the training observations, labeled train.Direction below.
4. A value for K, the number of nearest neighbours to be used by the classifier.

Generally k gets decided on the square root of number of data points. But a large k value has benefits which include reducing the variance due to the noisy data; the side effect being developing a bias due to which the learner tends to ignore the smaller patterns which may have useful insights.

Misclassification Error and Curse of dimensionality



Computing the Misclassification Error:

1 - Apparent Error

One possibility is to estimate the misclassification error of the classification rule c using the training sample which was used to derive the classification rule c .

This estimate of the misclassification error is called the training error rate or apparent error:

$$\widehat{err}_{apparent}(\hat{c}) = \frac{1}{n} \sum_{i=1}^n I(\hat{c}(\mathbf{x}_i) \neq y_i),$$

where I denotes the indicator function which equals one, if the expression is true, and zero if it is false.

Computing the Misclassification Error:

2-Test sample

Sometimes we have an independent *test sample* $(\tilde{\mathbf{X}}_1, \tilde{Y}_1), \dots, (\tilde{\mathbf{X}}_{\tilde{n}}, \tilde{Y}_{\tilde{n}})$, where \tilde{n} denotes the size of the test sample and we can compute the *test error* as estimator of the misclassification error:

$$\widehat{err}_{test}(\hat{c}) = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} I(\hat{c}(\tilde{\mathbf{x}}_i) \neq \tilde{y}_i). \quad (3)$$

The test error $\widehat{err}_{test}(\hat{c})$ is an unbiased estimator of the misclassification error of a classification rule trained on a sample of size n .

The apparent error $\widehat{err}_{apparent}(\hat{c})$ tends to underestimate the misclassification error.

Computing the Misclassification Error:

3-Validation set approach

When we do not have a test set we can create it:

- 1) divide the training sample into two random partitions of equal size: a training sample of $n/2$ observations and a validation sample of $n/2$;
- 2) estimate the classification rule c on the training sample;
- 3) estimate the misclassification error on the validation sample.

The resulting estimator of the misclassification error is unbiased for the misclassification error of a classification rule trained on a sample of size $n/2$, but exploits only 50% of the observations of the sample for the estimation of the classification rule and the misclassification error:

Computing the Misclassification Error:

4-Resampling methods

When we use the training set to estimate the classification error rate, it's obvious that we should have good results because that's this set was used to build the classifier.

Instead, when we split our data into training set and test set to test for our accuracy, we waste a lot of information that could be used to build the classifier.

There are solutions to not waste this information: resampling methods.

We introduce leave-one-out, k-fold cross-validation and bootstrap estimation of the misclassification error and other regression characteristics of interest.

In classification problems we are interested to estimate the misclassification error of a given classification rule c for the class variable Y depending on a random vector of predictors X .

Our aim is to find a classifier with small misclassification error.

4.1 - Cross-validation

Cross-validation is a useful tool when the size of the data set is limited.

In a perfect world, our data sets would be large enough that we could set aside a sizable portion of the data set to validate (i.e., examine the resulting prediction error) the model we run on the majority of the data set.

Unfortunately, this type of data is not always available (<https://quantdev.ssri.psu.edu/tutorials/cross-validation-tutorial>)

The choice of the number of splits (or “folds”) impacts bias (the difference between the average/expected value and the correct value - i.e., error) and variance. Generally, the fewer the number of splits, the lower the variance and the higher the bias/error (and vice versa).

The basic idea of cross-validation



FIGURE 5.1. A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.

(Figure 5.1, James et al, 2013).

The validation method can be repeated many times, each time using a different random split of the observations into a training set and a validation set.

Variations on Cross-Validation

There are a number of variations on the k-fold cross-validation procedure.

Three commonly used variations are as follows:

- **Train/Test Split:** Taken to one extreme, k may be set to 2 (not 1) such that a single train/test split is created to evaluate the model.
- **LOOCV:** Taken to another extreme, k may be set to the total number of observations in the dataset such that each observation is given a chance to be held out of the dataset. This is called leave-one-out cross-validation, or LOOCV for short.
- **Stratified:** The splitting of data into folds may be governed by criteria such as ensuring that each fold has the same proportion of observations with a given categorical value, such as the class outcome value. This is called stratified cross-validation.
- **Repeated:** This is where the k-fold cross-validation procedure is repeated n times, where importantly, the data sample is shuffled prior to each repetition, which results in a different split of the sample.
- **Nested:** This is where k-fold cross-validation is performed within each fold of cross-validation, often to perform hyperparameter tuning during model evaluation. This is called nested cross-validation or double cross-validation.

4.1.1 Leave-one-out-cross-validation

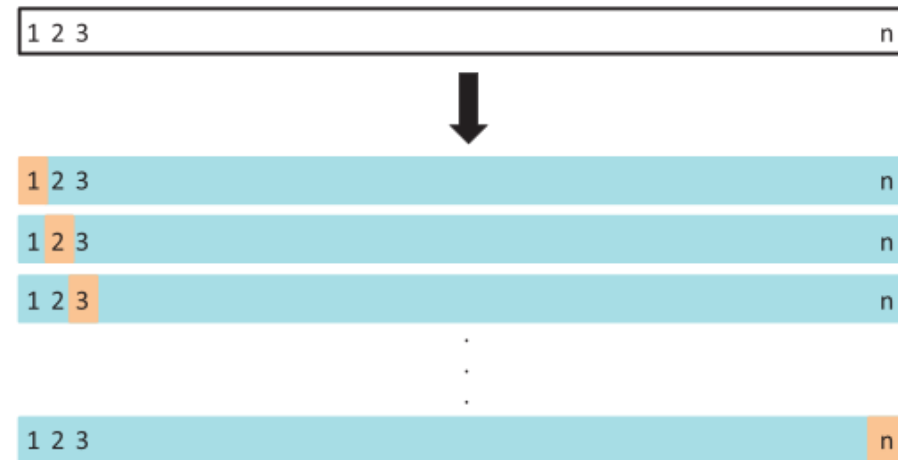


FIGURE 5.3. A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.

(Figure 5.3, James et al, 2013).

Leave-one-out-cross-validation

Instead of creating two equal sized data sets - test and validation sample - we use a single observation as validation set and all remaining observations as training sample. Leaving all observation once out we get the LOOCV estimator of the misclassification error:

$$\widehat{err}_{LOOCV}(\hat{c}) = \frac{1}{n} \sum_{i=1}^n I(\hat{c}_{(i)}(\mathbf{x}_i) \neq y_i), \quad (4)$$

where $\hat{c}_{(i)}$ denotes the classification rule estimated by the sample excluding the i -th observation.

Leave-one-out-cross-validation

The LOOCV estimator $\widehat{err}_{LOOCV}(\hat{c})$ is an unbiased estimator of the misclassification error of a classification rule trained on a sample of size $n - 1$:

$$E\left[\widehat{err}_{LOOCV}(\hat{c})\right] = \frac{1}{n} \sum_{i=1}^n E\left[I\left(\hat{c}_{(i)}(\mathbf{x}_i) \neq y_i\right)\right] = P\left(\hat{c}_{(\cdot)}(\mathbf{X}) \neq Y\right).$$

4.1.2 k-fold cross-validation

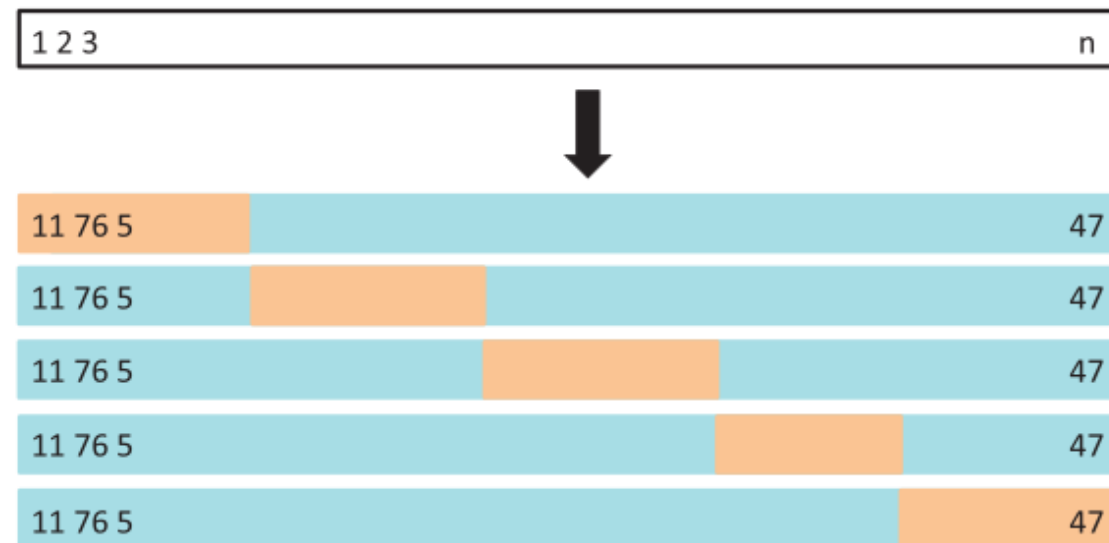


FIGURE 5.5. A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.

From: Berthold Lausen - Supervised Learning / Classification

k-fold cross-validation

A disadvantage of LOOCV is that the $(\hat{c}_{(i)})$ are highly correlated. To reduce the correlation we partition the sample in k sets of approximately equal size:

$$U = \{1, 2, \dots, n\} = \cup_{i=1}^k U_i \text{ and } \cap_{i=1}^k U_i = \emptyset.$$

The k -fold cross-validation estimator is defined as average of k estimators which use all k sets once as validation set and the data of the other $k - 1$ sets as corresponding training set:

$$\widehat{err_{CV}}(\hat{c}) = \frac{1}{k} \sum_{i=1}^k \frac{1}{|U_i|} \sum_{j \in U_i} I(\hat{c}_{(U_i)}(\mathbf{x}_j) \neq y_j), \quad (5)$$

where $\hat{c}_{(U_i)}$ denotes the classification rule estimated by the sample excluding the observations of set U_i .

k-fold cross-validation

The k -fold cross-validation estimator $\widehat{err_{CV}}(\hat{c})$ is an unbiased estimator of the misclassification error of a classification rule trained on samples of approximately size of $n(1 - 1/k)$:

$$E[\widehat{err_{CV}}(\hat{c})] = \frac{1}{k} \sum_{i=1}^k \frac{1}{|U_i|} \sum_{j \in U_i} E[I(\hat{c}_{(U_i)}(\mathbf{x}_j) \neq y_j)] = P(\hat{c}_{(kCV)}(\mathbf{X}) \neq Y).$$

4.2 The Bootstrap

The (theoretical) bootstrap estimators are defined by the observed empirical distribution. Often we approximate these estimators by simulation using the empirical distribution.

The simulation is defined by :

- 1) Generate B bootstrap samples of size n by sampling with replacement from the empirical distribution;
- 2) Compute for each bootstrap sample U^*i , $i = 1, \dots, B$, a classifier $c(U^*i)$ and a validation error using the observations of the so called out-of-bag (OOB) sample which is defined by all observations which are not in the i -th bootstrap sample.
- 3) The average of the B validation errors defines a bootstrap estimator of the misclassification error.

Default Metrics to Evaluate Machine Learning Algorithms in “*Caret*”

Accuracy is the percentage of correctly classifies instances out of all instances. It is more useful on a binary classification than multi-class classification problems because it can be less clear exactly how the accuracy breaks down across those classes (e.g. you need to go deeper with a [confusion matrix](#)).

Kappa or Cohen’s Kappa is like classification accuracy, except that it is normalized at the baseline of random chance on your dataset. It is a more useful measure to use on problems that have an imbalance in the classes (e.g. 70-30 split for classes 0 and 1 and you can achieve 70% accuracy by predicting all instances are for class 0).

From: <https://machinelearningmastery.com/machine-learning-evaluation-metrics-in-r/>

All the Metrics

Evaluation metrics with the caret package in R:

1. Accuracy and Kappa
2. RMSE and R^2
3. ROC (AUC, Sensitivity and Specificity)
4. LogLoss

Logistic Regression (LR)

Logistic regression is parametric approach used as prediction models.

It is one of the most popular classification algorithms mostly used for binary classification problems, however, some variants may deal with multiple classes as well.

Note that a logistic regression model is unstable if the number of predictors exceeds the number of observations. In this case, a variable selection needs to be conducted.

Area Under ROC Curve

The usefulness of a medical test related is linked to the evaluation of:

- how correct its results are
- how much uncertainty is present in the following:
 1. the group of sick subjects should be potentially positive to the test
 2. the group of healthy subjects should be potentially negative to the test.

Tests can give correct results, but they can also present errors:

1. a healthy subject who is positive for the test is defined as false positive
2. a sick person who is instead negative for the test is defined as false negative.

These types of errors relate to decision errors in statistical tests.

Types of errors in medical tests related to decision errors in statistical tests

In the hypothesis tests, two hypotheses are compared, the null hypothesis H_0 and the alternative hypothesis H_1 . However, the decision made on which of the two hypotheses is the true one does not always lead to correct results

	I accept H_0 (and therefore rejection H_1)	I refuse (and therefore I accept H_1)
H_0 true		first type error (α)
H_0 false	second type error (β)	

	Positive People	Negative People
Sick people	SHOULD BE	False negatives (error β)
Healthy people	False positives (α)	SHOULD BE

These two types of errors are strictly related to the concepts of sensitivity and specificity

Sensitivity and specificity

SENSITIVITY is the ability of a test to identify the disease when it is present, therefore in sick subjects.

SPECIFICITY is the ability of a test to rule out disease when it is not present, therefore in healthy subjects.

Sensitivity and specificity

Let's consider the following contingency table on the test result and the actual state of the disease:

	Positive People +	Negative People -	
Sick people	a	b	a+b
Healthy people	c	d	c+d
	a+c	b+d	a+b+c+d=n

Sensitivity (Recall)

We define the sensitivity of the test as the probability of being positive when being sick:

	Positive People +	Negative People -	
Sick people	a	b	a+b
Healthy people			

$$\text{Sensitivity (or Recall)} = a/(a+b)$$

The sensitivity of the test is linked to the false negative error β corresponding to the probability of being negative in the test being sick:

$$\beta = b/(a+b)$$

Specificity

We define specificity of the test the probability of be negative on the test when being healthy.

	Positive People +	Negative People -	
Sick people			
Healthy people	c	d	c+d

$$\text{Specificity} = d/(c+d)$$

The specificity of the test is linked to the false positive error α corresponding to the probability of being positive in the test being healthy:

$$\alpha = c/(c+d)$$

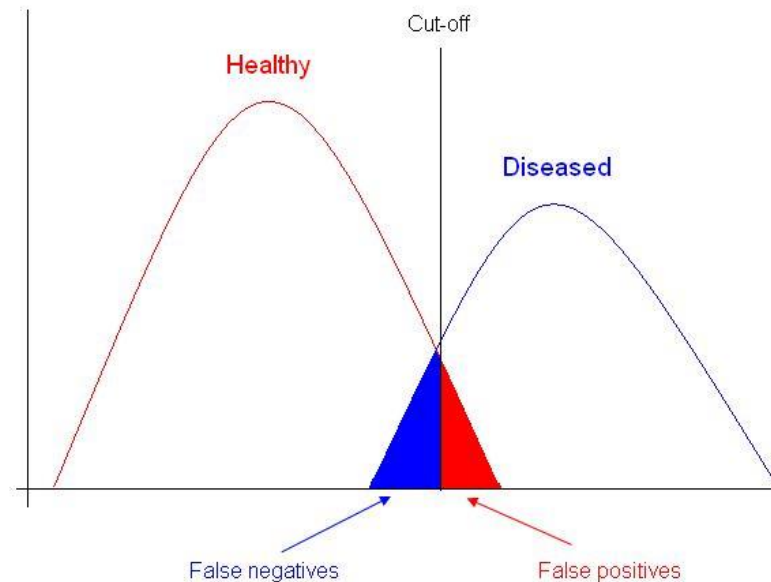
Pay attention

If a test is not sensitive, it will fail to detect the disease in some actually sick person and therefore increase the number of false negatives by increasing β .

If a test is not specific, it will incorrectly indicate the disease in healthy subjects and therefore increase the number of false positives and consequently α .

Cut off or cut point or threshold value

The cut off value or threshold value, chosen to discriminate between positive and negative tests, varies sensitivity, specificity, α and β .



A cut off more shifted to the right will allow to correctly identify most of the healthy, giving the test a high specificity and therefore few false positives, but it will underestimate the proportion of sick patients, giving the test a low sensitivity.

A cut off more shifted to the left will allow to correctly identify most of the sick patients, giving the test a high sensitivity and therefore few false negatives, but it will underestimate the proportion of the healthy ones, giving the test a low specificity.

https://en.wikivet.net/Evaluation_of_diagnostic_tests

Cut off Choice

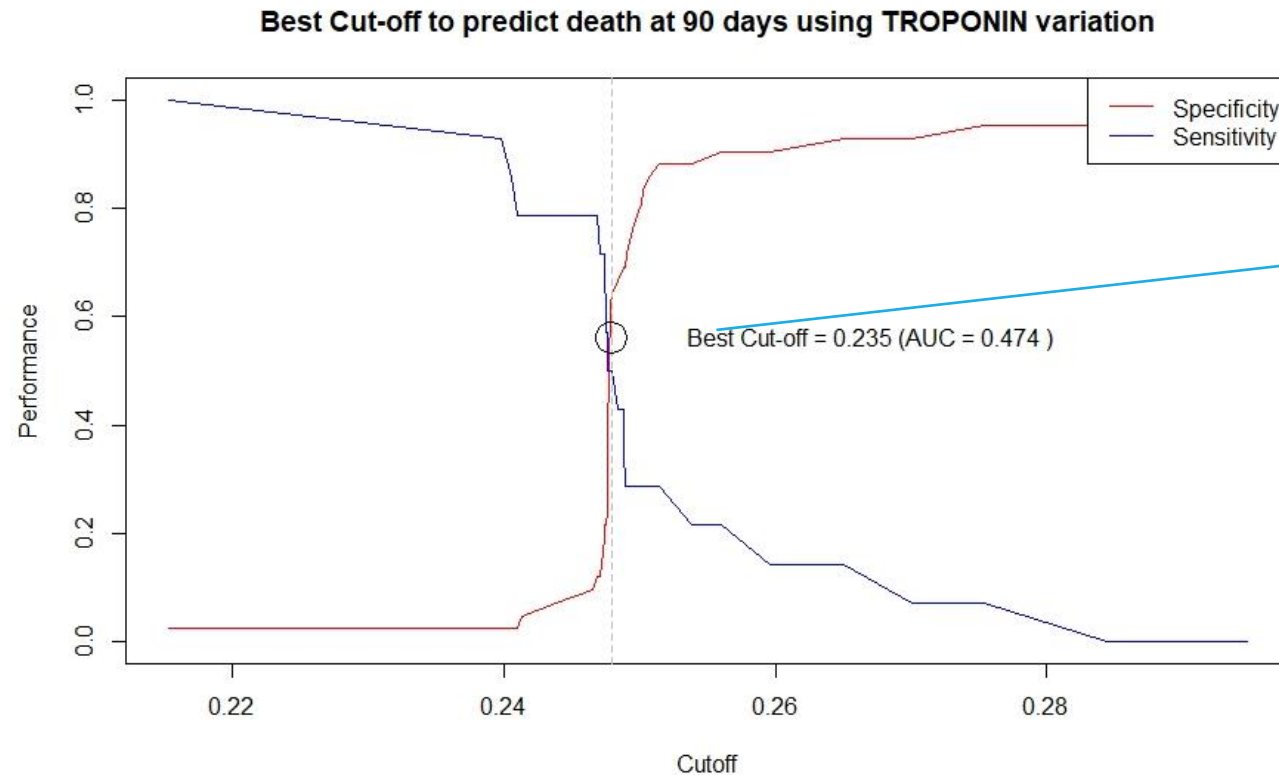
Thus, sensitivity and specificity are inversely related to each other depending on the choice of the cut off value: if we increase the specificity by moving the threshold value, the sensitivity decreases and vice versa.

If the sick and healthy populations are assumed normal, the optimal threshold value that minimizes classification errors (most powerful test) is equal to the value corresponding to the intersection point of the two sensitivity and specificity curves in function of the threshold value itself.

For the threshold value for which there is specificity = 1, all the healthy are such (there are no false positives), but we also have that all the patients will be healthy (sensitivity = 0).

For the threshold value for which there will be sensitivity = 1, all the sick are such, but also all the healthy will be sick (specificity = 0).

Real case Example




The optimal threshold simultaneously maximizes sensitivity and specificity, minimizing classification errors.

However, this choice, dictated by purely probabilistic considerations, is not necessarily the best.

It depends on the type of test we want to do and on its clinical, economic and social impact.

The correct procedure

Diagnostic tests of screening: usually, we proceed first with very sensitive and not very specific diagnostic tests that are used to identify people at risk of disease



Confirmatory diagnostic tests: then, patients who tested positive for these first tests undergo more specific and less sensitive tests to discriminate between real sick and false positives in the previous test.

Predictive values

Sensitivity and specificity are measures of a diagnostic test but do not respond to two important clinical problems:

1. if a patient is positive for the test, what is the probability that he has the disease?
2. if instead the test result is negative what is the probability that it is really healthy?

These questions can be answered through predictive values.

Positive Predictive Value (Precision)

It is the probability of being sick when positive to the diagnostic test.

	Positive People +	Negative People -	
Sick people	a		
Healthy people	c		
	a+c		

$$\text{Positive Predictive Value} = a/(a+c)$$

Negative Predictive Value

It is the probability of being healthy when resulting negative the tests.

	Positive People +	Negative People -	
Sick people		b	
Healthy people		d	
		b+d	

$$\text{Negative Predictive Value} = d/(d+b)$$

Probability ratios

To decide the quality of a diagnostic test, you can use the probability or likelihood ratios.

Positive probability ratio LR^+ is defined as

$$LR^+ = (1-\beta)/\alpha = \text{Sensitivity}/(1- \text{Specificity})$$

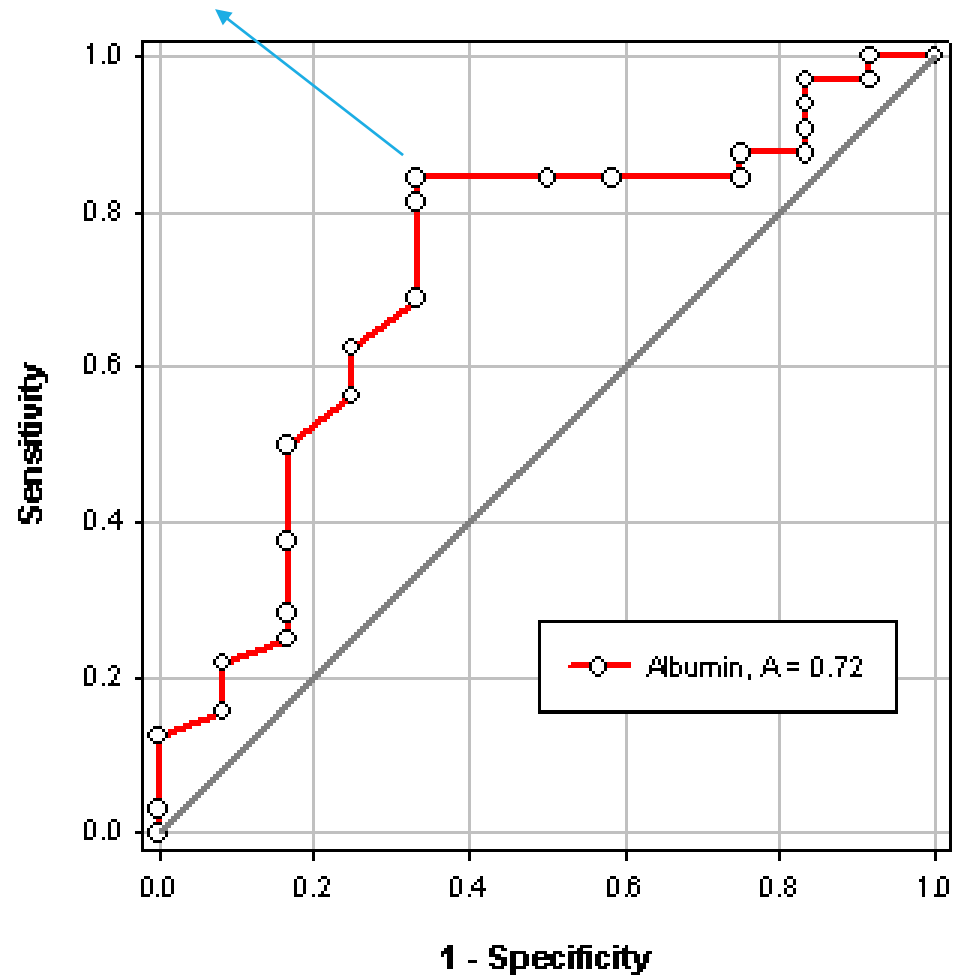
The higher the better the test, because $(1- \text{Specificity})$ is the probability of having false positives.

Negative probability ratio LR^- is defined as

$$LR^- = \beta/(1- \alpha) = (1- \text{Sensitivity})/ \text{Specificity}$$

The lower the better the test, because $(1- \text{Sensitivity})$ is the probability of having false negatives.

Generic point that is obtained for a particular test threshold value which leads to having a well-determined x value of α and y of sensitivity



<http://www.sigmaplot.co.uk/splot/products/sigmaplot/productuses/prod-uses42.php>

ROC (Receiver Operating Characteristic) curve

It is a curve used for the evaluation of a binary classifier and it consists of representing the trend of sensitivity with respect to $(1 - \text{specificity}) = \alpha$ as the value of the cut off of the diagnostic test changes.

The ROC curve therefore links the probability of obtaining a true positive among the people who are actually sick (sensitivity) to the probability α of obtaining a false positive among healthy people when the chosen test threshold varies (cut off value).

Usually, however, the available data are discrete and the ROC curve is obtained by interpolation (smoothing).

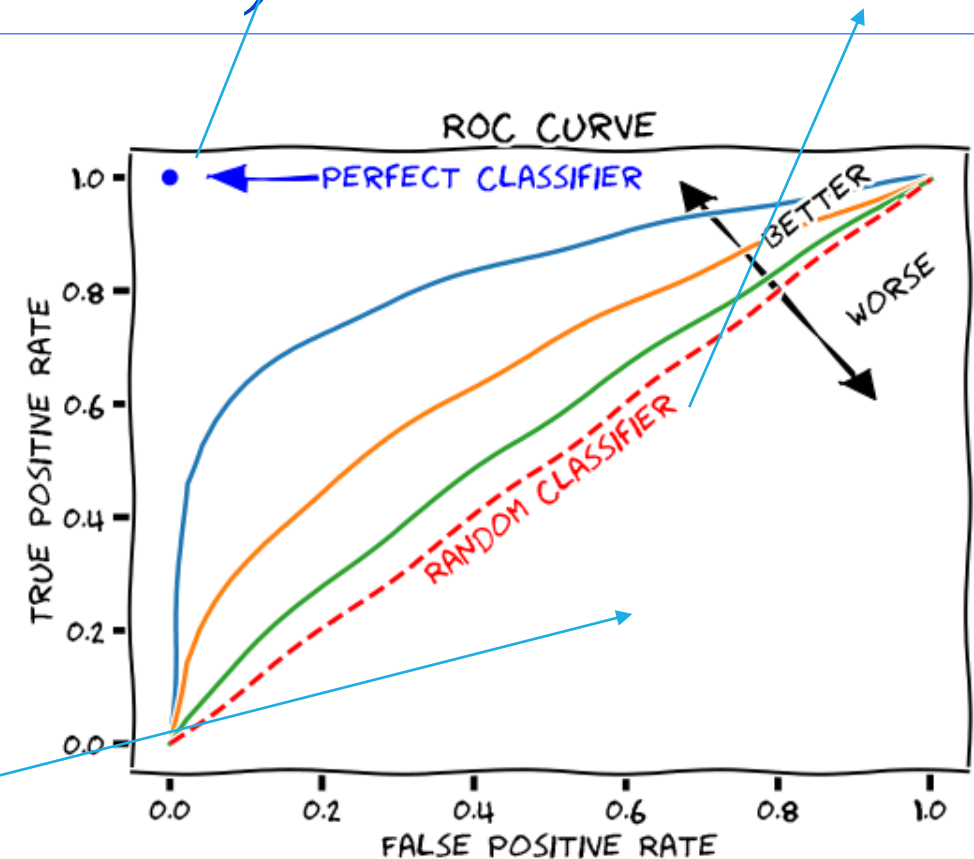
In this case the point is reduced to the coordinate point (0,1) and the area under the curve corresponds to the area of the coordinate square (0,0), (0,1), (1,0), (1,1) holds $AUC = 1$.

AUC (Area Under Curve)

Classifier with null information value

The discriminating ability of the test, i.e. how well the test performed can distinguish between healthy and sick, is proportional to the area underlying the ROC curve ($AUC = \text{Area Under Curve}$) and is equivalent to the probability that a randomly extracted subject is correctly classified as sick if positive to the test and as healthy if negative.

If the false positives are more than the true positives and therefore the test does not make sense (when $AUC < 0.5$)



<https://commons.wikimedia.org/wiki/File:Roc-draft-xkcd-style.svg>

Classification of the discriminating ability of a test proposed by Swets (1998)

- >AUC <0.5 The test does not make sense
- >AUC = 0.5 Non-informative test
- >0.5 <AUC ≤ 0.7 Poor test
- >0.7 <AUC ≤ 0.9 Moderately accurate test
- >0.9 <AUC < 1.0 Highly accurate test
- >AUC = 1.0 Perfect test

Testing Hypothesis

Under the hypothesis of proper ROC curves in which AUC can be considered approximately normally distributed, it is possible to test the significance of the discriminating ability of the tests by formulating the following statistical hypotheses:

$$H_0: E(AUC)=0.5$$

$$H_1: E(AUC)>0.5$$

It is also possible to compare two tests using a significance test with hypotheses:

$$H_0: E(AUC_a - AUC_b)=0$$

$$H_1: E(AUC_a - AUC_b) \neq 0$$

Bayes Classifier

Naive Bayes classifier is a simple classifier that has its foundation on the well known Bayes's theorem.

In the field of supervised classification, the algorithm evaluates a probability for each class, when the predictor values are given.

Intuitively, we can select the class that has highest probability.

This classifier can perform well in many complex real-world problems but has strong assumptions regarding independence.

Let us recall some concepts of probability... on the blackboard!

Take notes...

The diagram shows the formula for Bayes' theorem:
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
 Each part of the formula is labeled with an arrow:

- $P(c|x)$ is labeled "Posterior Probability" with a downward arrow.
- $P(x|c)$ is labeled "Likelihood" with an upward-left arrow.
- $P(c)$ is labeled "Class Prior Probability" with an upward-right arrow.
- $P(x)$ is labeled "Predictor Prior Probability" with a downward-right arrow.

Bayes Classifier

Suppose we want to classify an observation into one of K classes with $K \geq 2$. Let $f_k(X) = P(X = x|Y = k)$ denote the probability density function (pdf) of X for an observation that belongs to the k -th class. Let π_k be the **prior** or overall probability that Y comes from class k .

Consequently, $f_k(X)$ will be relatively large, if there is a high probability that an observation in the k -th class has $X = x$. On the other hand, $f_k(X)$ will be relatively small, if there is unlikely that an observation in the k -th class has $X = x$.

Using Bayes Theorem we get:

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}. \quad (2)$$

Bayes Classifier

Assuming a random sample of the population we estimate π_k by the relative frequencies of each class and assuming that we have a consistent estimate of the pdf $f_k(x)$ we can classify each observation by plugging the estimates in the Bayes formula (2) above.

We refer to $p_k(x) = P(Y = k|X = x)$ as the **posterior** probability of an observation belonging to class k given that $X = x$. The resulting classification rule or classifier is known as *Bayes classifier* which classifies an observation to the class with highest probability.

For the true, but in **general unknown**, π_k and $f_k(x)$ with $k = 1, \dots, K$ the *Bayes classifier has the lowest possible error rate*.

Bayes Classifier

This algorithm is termed "naive" because it starts with the assumption of independence of variables. In other words, a Naive Bayes classicist assumes that the presence of a specific feature in a class is uncorrelated with the presence of other features.

Let's take a practical example: I consider a dog to be such in that:

- It has a tail
- It has four legs
- It has two eyes
- It has a nose

Now, these features coexist and depend on each other, but the Naive Bayes algorithm considers them as if they independently contribute to the probability that a specific observation is a dog. Thus, it might mistake a person with two eyes and a nose for a dog because the variables are not considered individually in their coexistence and dependence

.

How Bayes Classifier works

I want to know whether I can play golf given the Weather conditions. Thus, I have a target consisting of two possible values, "Yes," "No." Binary classification.

I can calculate a frequency table of the Game event based on Time.

Weather conditions	Play=YES	Play=NO
Clouds	4	
Rain	2	3
Sun	3	2
Total	9	5

Bayes Classifier

I can now exploit Bayes' theorem to calculate the probability of playing given a certain event. The class that will have the highest probability will then be the one that will be the outcome of our prediction.

What is the probability of play given the sun?

$$P(\text{Game} \mid \text{Sun}) = P(\text{Sun} \mid \text{Game}) * P(\text{Game}) / P(\text{Sun})$$

$$P(\text{Sun} \mid \text{Game}) = 3/9 = 0.33 \text{ (on three occasions it was sunny and I played)}$$

$$P(\text{Game}) = 9/14 = 0.64 \text{ (I played 9 times in total out of 14 times)}$$

$$P(\text{Sun}) = 5/14 = 0.36 \text{ (it was sunny 5 out of 14 times)}$$

At this point I can calculate the formula: $0.33 * 0.64 / 0.36 = 0.60$

Bayes Classifier

Let's do the same with rain:

$$P(\text{Rain} \mid \text{Game}) = 2/9 = 0.22$$

$$P(\text{Game}) = 9/14 = 0.64$$

$$P(\text{Rain}) = 5/14 = 0.36$$

$$\text{Probability of playing with rain: } 0.22 * 0.64 / 0.36 = 0.39$$

Of the two, he is more likely to play with sunshine. The conditional probability of playing with rain is indeed higher: $0.60 > 0.39$



*Thank
you*

A gold fountain pen nib is positioned at the end of the word "you", as if it has just finished writing it.