Week02 - Summary

Classification

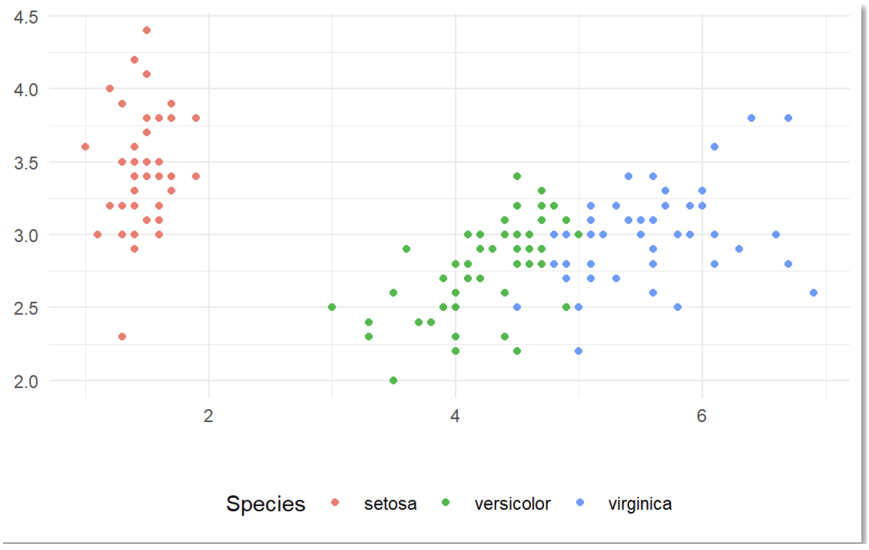
Basic Principles of Classification

- Each observation has two properties
  - A class label or response, y
  - A feature vector (vector of predictor variables), $x = (x_1, x_2, \ldots, x_p)$

Classification vs Clustering

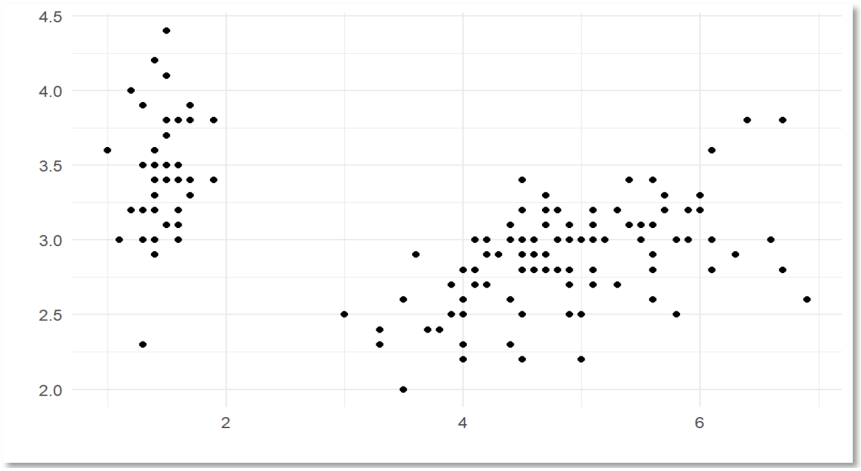
Classification:

- Classes are pre-defined, want to use trained model to form a classifier for future observations (supervised)



Clustering:

- Classes are unknown, want to discover them from the data (unsupervised)



Logistic Regression

Binary or Two Class Classification

- Binary in there are two possible values (1 or 0, TRUE or FALSE)
- Examples of binary classification:
  - Email: Spam/Not Spam
  - Tumour: Malignant/Benign
- Labels are similarly described, $y \in \{0, 1\}$
  - 0: “negative class”
  - 1: “positive class”

Why not use simple linear regression?

- The target, Y, is binary
- Linear regression is not constrained to $0 < y < 1$ for all x
  - How to interpret elsewhere? when y_hat > 1 (or 0)

Linear Regression Misspecifications

- The regression line $\beta_0 + \beta_1 x$ can span the entire real line
  - all values between -∞ to ∞
- In the tumour diagnosis problem, the target variable y only takes two values: 0 or 1
- The linear regression model is not well specified for this purpose

Logistic Regression

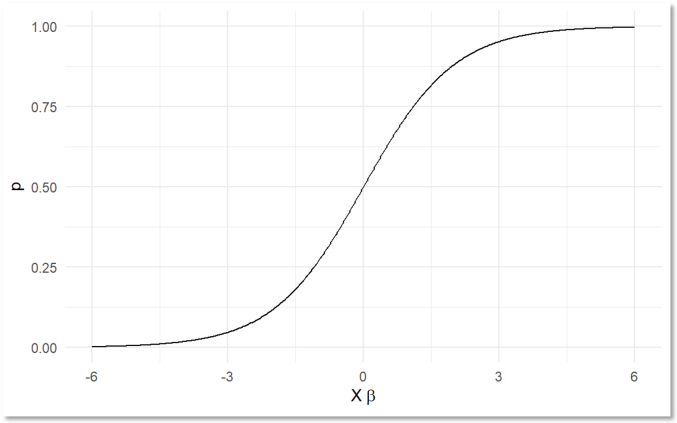
- Recall multiple regression

$$\mathbf{Y} = \mathbf{X\beta} + \boldsymbol{\varepsilon}$$

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

- Can write $\mathbb{E}Y = \mathbf{X\beta} = \mu$
- Generalizing this to

$$g(\mathbb{E}Y) = \mathbf{X\beta} = g(\mu)$$



Logistic Regression Terminology

- Logistic function $\frac{1}{1+e^{-\mathbf{X\beta}}}$
  - Responsible from mapping the features from $(-\infty, \infty) = \mathbb{R}$ to (0, 1)
- Odds ratio: $\frac{p}{1-p}$
  - Maps the probability from (0, 1) to $(0, \infty)$
- Log-odds or logit: $\log\left(\frac{p}{1-p}\right)$
- In logistic regression we want the values in the logit space to be linear in X

Linear Discriminant Analysis (LDA)

- LDA undertakes the same task as logistic regression. It classifies data based on categorical variables

- Malignant or benign
- Making profit or not
- Buy a product or not
- Satisfied customer or not

Bayes' Theorem in the Classification Context

$$p_k(x) = P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}$$

Posterior: The probability of classifying observation to group k given it has features x

Prior: The prior probability of an observation in general belonging to group k, π_k

- $f_k(x) = P(X = x|Y = k)$ is the density function for feature x given it's in group k

Logistic Regression vs LDA Formulations

- In Logistic Regression the probability of Y being from the positive class is

$$p_1(x) = P(Y = 1|X = x) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

- Bayes' Theorem states

$$p_k(x) = P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}$$

- π_k: Probability of coming from class k (prior probability)
- $f_k(x)$: Density function for X given that X is an observation from class k

LDA Estimates

- We can estimate π_k and $f_k(x)$ to compute $p_k(x)$
- The most common model for $f_k(x)$ is the Normal Density (LDA)

$$f_k(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right)$$

- Using the above density, we only need to estimate three quantities to compute p_k(x): μ_k, σ_k^2 and π_k
- For simplicity, assume common variance

Why not logistic regression?

- In the case where n is small, and the distribution of predictors X is approximately normal, then LDA is more stable than logistic regression
- LDA is more popular when we have more than two response classes; more intuitive to predict class assignments
- When the classes are well separated, the parameter estimates for logistic regression are unstable. However, LDA doesn't suffer any stability issues in this case

Logistic Regression vs LDA

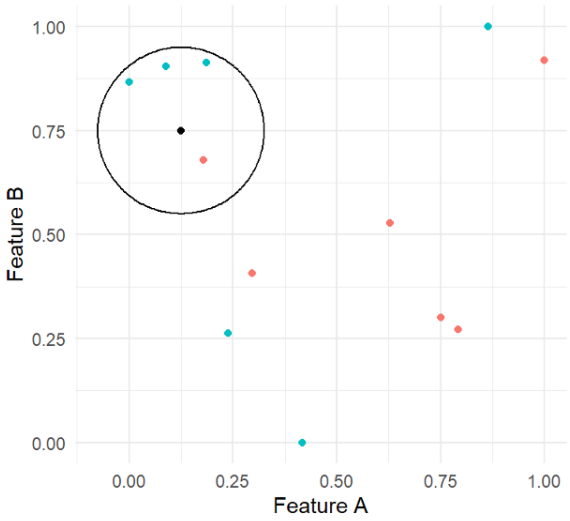
- Similarity
  - Both logistic regression and LDA produce linear boundaries
- Differences
  - LDA assumes that the observations are drawn from the normal distribution with common variance in each class, while logistic regression does not have this assumption
  - LDA would do better than logistic regression if the assumption of normality hold, otherwise logistic regression may outperform LDA

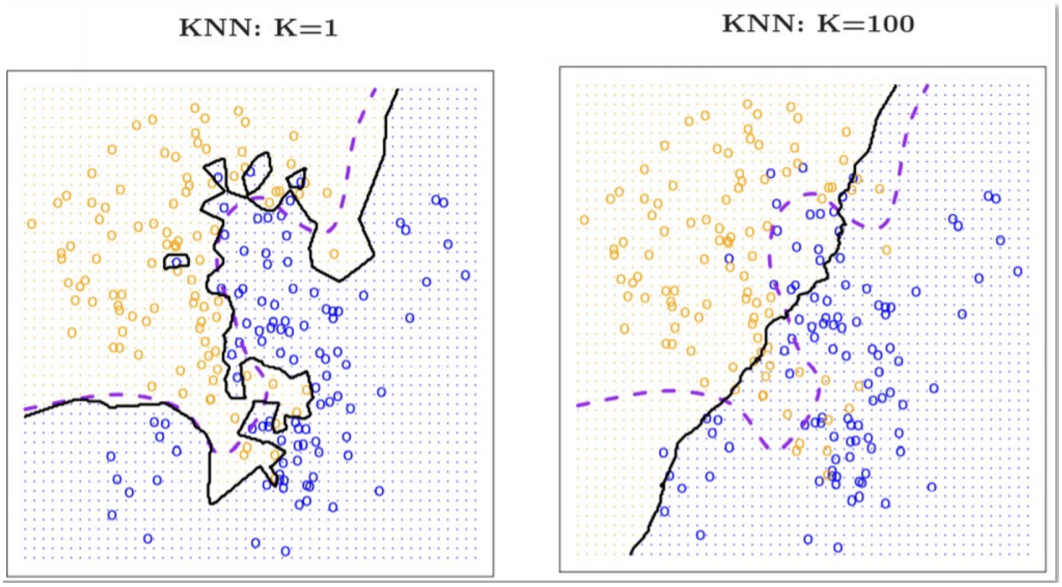
k-Nearest Neighbours (k-NN)

- k-NN model is probability of an observation with features x belonging to group l depends on the membership of the nearest points to x

$$P(Y = \ell|\mathbf{x}) = \frac{1}{k}\sum_{N_x^k} \mathbb{1}_{\{y=\ell\}}$$

= $1/k$ x Count of the closest k points that belong to group l





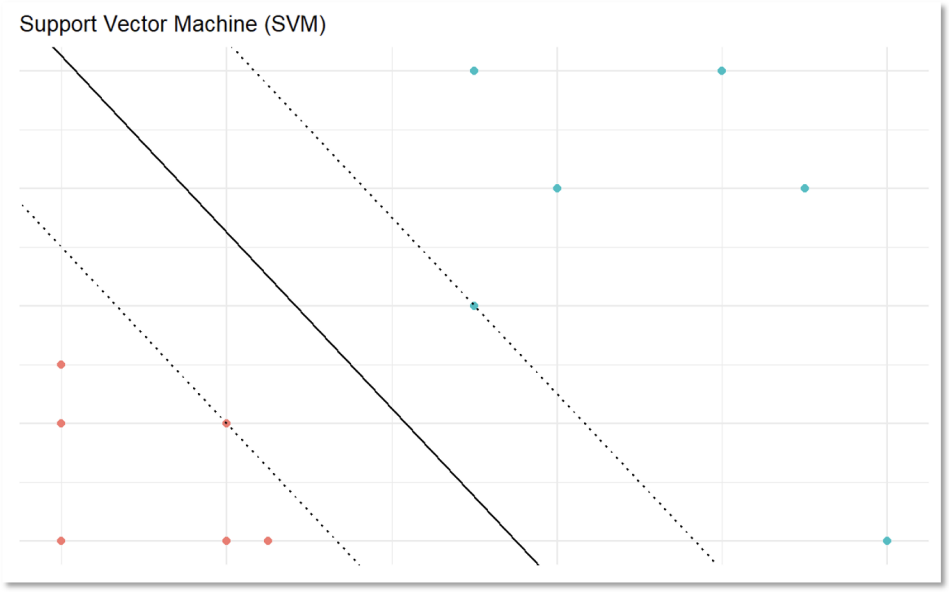
kNN vs (LDA and Logistic Regression)

- kNN takes a completely different approach
- kNN is completely non-parametric: No assumptions are made about the shape of the decision boundary
- Advantage of kNN: We can expect kNN to dominate both LDA and logistic regression when the decision boundary is highly nonlinear
- Disadvantage of kNN: kNN does not tell us what predictors are important (no table of coefficients)
- Tuning of k is needed to avoid undersmoothing or oversmoothing the boundary

Support Vector Machines (SVM)

- Basic idea behind SVM
  - Find a plane that separates the classes in the feature space
- If a basic mathematical plane is not possible due to overlap
  - Relax the idea of complete separation (allow points to violate the boundary)
  - Enrich and enlarge the feature space so that separation is possible

- Think dimension expansion



What is a Hyperplane?

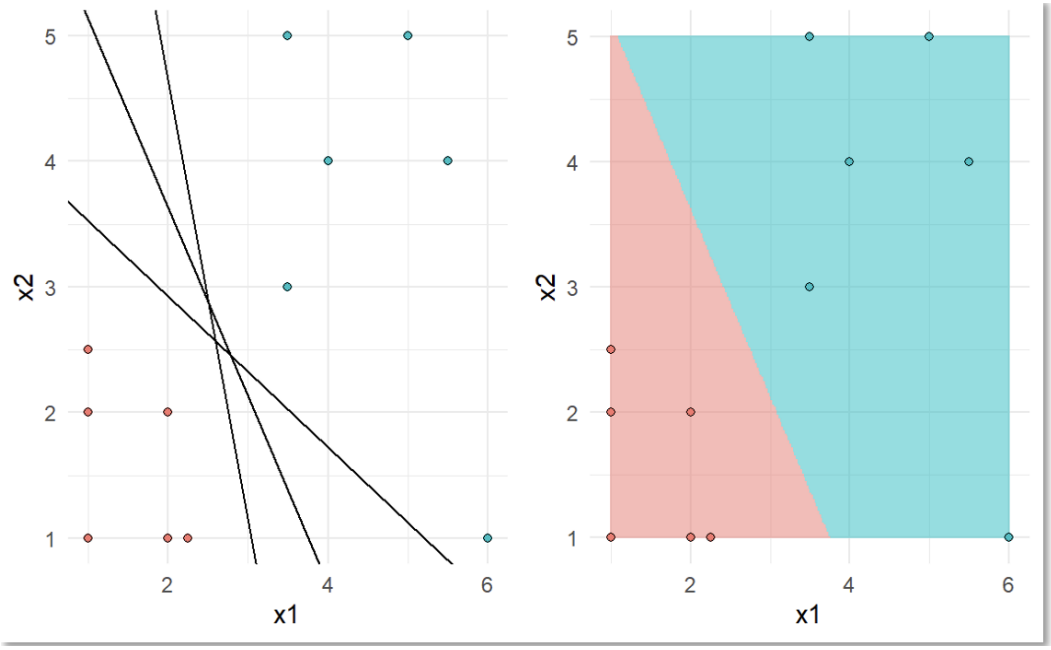
- In p dimensions it is a flat affine subspace of dimension $p - 1$
- General equation has the form

  $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$
- In $p = 2$ dimensions, the hyperplane is a line
- If $\beta_0 = 0$, the hyperplane passes through the origin, otherwise it does not
- The vector $(\beta_1, \beta_2, \ldots, \beta_p)$ is called the normal vector
  - It points in a direction orthogonal to the surface of the hyperplane

Separating Hyperplanes

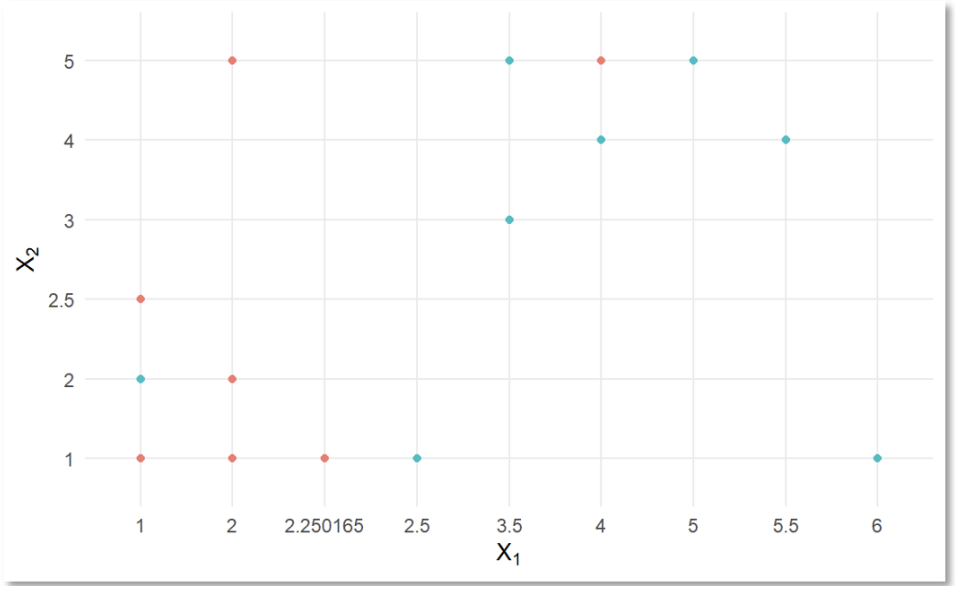
- Recode the target y_i
  - Negative class (benign) $y_i = -1$
  - Positive class (malignant) $y_i = 1$

- $f(x_i) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ defines a hyperplane
  - $f(x) > 0$ defines a region on one side of the hyperplane
  - $f(x) < 0$ defines a region on one other side of the hyperplane
- $y_i f(x_i) > 0$ for all i, $f(x_i)$ defines a separating hyperplane



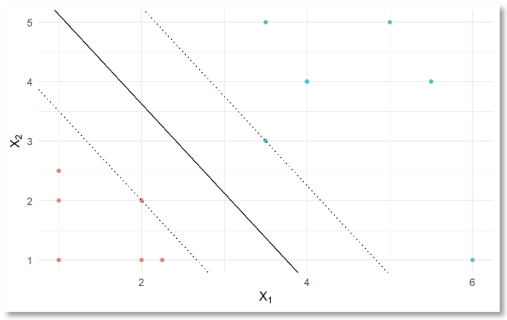
Non-Perfect Separation

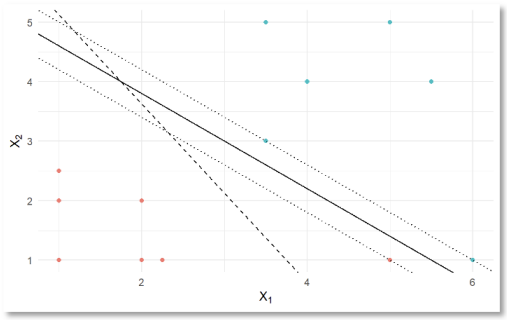
- There is no linear boundary (hyperplane) that perfectly separates the classes
- This is typically the case that observations don't have a perfect boundary of separation
  - Except in the case when $n < p$ (more features than observations)



Effect of noisy data

Consider the impact of one extra observation





- Data could be separable, but noisy → unstable solution for the maximal margin classifier
- The support vector classifier maximises a soft margin
  - relaxes requirement for all observations to be on the correct side of the margin

Support Vector Classifier

Support Vector Classifier solves the following optimisation problem:

$$\max_{\beta_0,\beta_1,\beta_2,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n} M \quad \text{such that} \quad \sum_{j=1}^{p} \beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \text{ and } \epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C$$

- C is a non-negative tuning parameter
- M is the width of the margin,
- ε_i are the slack variables that allow observations to be on the wrong side of the margin,
  - if $\varepsilon_i > 1$, then observation i is on the wrong side of the hyperplane
  - if $0 < \varepsilon_i \leq 1$, then observation i is on the correct side but inside margin
  - if $\varepsilon_i = 0$, then observation i is on the correct side and past the margin

Note: The cost of changing cost parameter C will determine if the hyperplane is allowed to have points going within or beyond the boundary

Impact of Cost Parameter, C

Feature Space Expansion

- Enlarge the space of features by including transformations:
  - e.g. new features that are powers and products $X_1^2, X_1^3, X_1 X_2$
  - Hence go from p-dimensional space to $P > p$ dimensional
- Fit (linear) support vector classifier in the expanded feature space
  - Impact is a non-linear decision boundary in original feature space
- Example: Suppose we start off in 2-dimensional feature space (X_1, X_2)
  - Make new feature space $(X_1, X_2, X_1^2, X_2^2, X_1 X_2)$
  - Then the decision boundary would be of the form:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0$$

- This leads to non-linear decision boundary in the original space (quadratic conic sections)

Nonlinearity and Kernels

- Polynomials get complicated and become a burden very quick as dimension increases
- More elegant solution is to induce nonlinear structure in support vector classifier with **kernels**
- The elegance comes from the role of the **inner product** in the support vector classifier definition

Kernel Functions

- Replace the inner product with a generalised function (**kernel**, *K*) of the form $K(x_i, x_j)$
- In this context, it quantifies the similarity of two observations
- Examples:
  - Polynomial kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^d \text{ where } \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{k=1}^{p} x_{ik}\, x_{jk}$$

  - Gaussian radial kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \sum_{k=1}^{p} (x_{ik} - x_{jk})^2\right)$$

SVM with more than two classes

- Two options to expand SVM from 2 to K classes

1. One vs all:
    - Fit K different binary classifiers, $f_k(x)$ for $k = 1, 2, \ldots, K$ where each boundary attempts to separate class k vs the rest
    - Then x_i is classified to k* where $f_k*(x_i) > f_j(x_i)$ for all j ≠ k*. (i.e. the largest distance from the boundary)

2. One vs one:
    - Fit all kC2 pairwise classifiers
    - Fit x_i to the class that wins the most pairwise comparisons

- Which to use?
    - If K is small, do one vs one. Otherwise recommended One vs all

Assessing Classification Models

Classification accuracy

- Overall classification accuracy:

$$\frac{Number\ of\ correct\ predictions\ from\ classifier}{Total\ number\ of\ observations\ in\ data}$$

- Advantages:
    - Simple and easy to understand
- Disadvantages
    - Makes no distinction about the type of errors being made
        - In spam filtering, the cost of erroneous deleting an important email is likely to be higher than incorrectly allowing a spam email past a filter
    - Does not consider the natural frequencies of each class

Confusion Matrix

		Actual	
		True	False
Predicted	True	True positive	False positive
	False	False negative	True negative

- Accuracy $= \frac{(TP+TN)}{(TP+FP+FN+TN)}$
- Sensitivity $= \frac{TP}{(TP+FN)} = \frac{TP}{P}$
- Specificity $= \frac{TN}{(TN+FP)} = \frac{TN}{N}$

- Precision $= \frac{TP}{(TP+FP)}$
- Recall $= \frac{TP}{(TP+FN)} = \frac{TP}{P}$
- $F_1 = \frac{2\ \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ (Harmonic mean)
- GM $= \sqrt{\text{Precision} \times \text{Recall}}$ (Geometric mean)