- Boxplot:
  - $Q_1 = N \times 0.25$
  - $Q_2 = N \times 0.5$
  - $Q_3 = N \times 0.75$
  - $IQR = Q_3 - Q_1$
  - $Bounds = [Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$
- Pearson $= \dfrac{\Sigma(y_{1i} - \bar{y}_1)(y_{2i} - \bar{y}_2)}{\sqrt{\Sigma(y_{1i} - \bar{y}_1)^2 \times (y_{2i} - \bar{y}_2)^2}}$
- Spearman: Assign ranks and apply Pearson formula. Example: [20, 10, 20, 30, 20] $\to$ [3, 1, 3, 5, 3]
- Normalization:
  - MinMax: $\dfrac{y_i - min}{max - min}$
  - Standardization: $\dfrac{y_i - \mu}{\sigma}$
- Binarization:
  - Range (equal width): Depends on variable range
    Example: $y \in [-1, 1] : [0.2, -0.1, 0.6] \to [1, 0, 1]$
  - Frequency (equal depth): Depends on variable mean
    Example: $\bar{y} = 25 : [10, 40, 30, 20] \to [0, 1, 1, 0]$
- Confusion Matrix:

		True A	True B	True C
Pred	A	TA	FA	FA
Pred	B	FB	TB	FB
Pred	C	FC	FC	TC

B
TP
TN
FP
FN

- Metrics:
  - Accuracy $= \dfrac{TP + TN}{total}$
  - Error rate $= 1 - Accuracy = \dfrac{FP + FN}{total}$
  - Recall $= \dfrac{TP}{TP + FN}$ (Sensitivity)
  - Fallout $= \dfrac{TN}{TN + FP}$ (Specificity)
  - Precision $= \dfrac{TP}{TP + FP}$
  - $\mathrm{F}_1 = \dfrac{TP}{TP + \frac{1}{2}(FP + FN)}$
- Error:
  - Sum of Squares Error: $\sum (z - \hat{z})^2$
  - Root Maen Squared Error: $\sqrt{\frac{1}{n} SSE}$
  - Mean Absolute (Percentage) Error: $\dfrac{1}{n} \sum \left| \dfrac{z - \hat{z}}{(z)} \right|$
- Information Gain: $IG(class|y_i) = E(class) - E(class|y_i)$
- Entropy: $E(y) = \sum_{i=1}^{k} \dfrac{|y_i|}{|y|} \sum_{j=1}^{n} -P(y_{ij}) \log(P(y_{ij}))$
- Decision trees:
  1. Choose feature with highest IG.
  2. Split dataset by that feature, create leaves if necessary.
  3. Repeat until unable to proceed.

  Prune: (Given a twig)
  1. Count it's leaves labels. Example: $\#A = 5,\ \#B = 6$
  2. Remove it's leaves.
  3. Relabel twig as a leaf. Example: $B(6/11),\ \#B > \#A$
- Vector Norm:
  $\|x\|_p = \sqrt[p]{\sum |x_i|^p} \qquad \|x\|_\infty = max\,|x_i|$

- Matrix Multiplication:
  $\begin{bmatrix} \dots & n \\ m & \dots \end{bmatrix} \cdot \begin{bmatrix} \dots & l \\ n & \dots \end{bmatrix} = \begin{bmatrix} \dots & l \\ m & \dots \end{bmatrix}$
- Gaussian Distribution:
  - Variance: $var = \dfrac{\sum (y_i - \mu)^2}{n(-1)}$
  - Standard Deviation: $\sigma = \sqrt{var}$
  - $P(y|\mu, \sigma) = \dfrac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left(-\dfrac{1}{2}\left(\dfrac{y - \mu}{\sigma}\right)^2\right)$
- Gaussian Mixture:
  - Covariance: $cov(y_1, y_2) = \dfrac{\sum (y_{1i} - \mu_1)(y_{2i} - \mu_2)}{n(-1)}$
  - Covariance Matrix: $\Sigma(y_1, y_2) = \begin{bmatrix} var(y_1) & cov(y_1, y_2) \\ cov(y_1, y_2) & var(y_2) \end{bmatrix}$
  - $det\,|\Sigma| = det \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$
  - $P(y|\mu, \Sigma) = \dfrac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \cdot exp\left(-\dfrac{1}{2}(y - \mu)^T \Sigma^{-1} (y - \mu)\right)$
- Naive Bayes:
  - MAP: $P(C|x) = \dfrac{P(C)P(x|C)}{P(x)}$
  - ML: $P(C|x) = P(x|C)$
  1. Calculate $P(C)$ for each class.
  2. Calculate $P(y|C)$ for each variable for each class.
  3. Calculate likelihood: $P(x|C) = P(y_1|C) \times \ldots \times P(y_d|C)$
- K-Nearest Neighbors:
  1. Distances: (for n variables)
     - Manhattan: $\sum |y_{1i} - y_{2i}|$
     - Euclidean: $\sqrt{\sum (y_{1i} - y_{2i})^2}$
     - Cosine: $\dfrac{\sum y_{1i}\ y_{2i}}{\sqrt{\sum y_{1i}^2} \sqrt{\sum y_{2i}^2}}$
     - Hamming: #Differences
  2. If weighted, for each variable multiply by weight.
  3. Choose K nearest neighbors.
  4. Classify using mean if variable is numeric, or mode if it is categoric.
- Regressions: $\hat{z} = Xw = (w^T X^T)^T$
  - Linear: $y = w_0 + w_1 x$
  - Polynomial: $w = (X^T X)^{-1} X^T z$
  - Ridge: $w = (X^T X + \lambda I)^{-1} X^T z$
- Perceptron:
  $\hat{z} = a(w^T x),\ \ a \leftarrow$ activation function
  Gradient Descent:
  $w^{new} = w^{old} + \Delta w \quad \text{where} \quad \Delta w = -\eta \dfrac{\partial E}{\partial w}$
- Neural Networks (MLP):
  - Forward: $x^{[0]} \to z^{[1]} = w^{[1]} x^{[0]} + b^{[1]} \to x^{[1]} = a\left(z^{[1]}\right) \to \ldots \to z^{[i]} = w^{[i]} x^{[i-1]} + b^{[i]} \to x^{[i]} = a\left(z^{[i]}\right)$
  - Backward:
    - $\delta^{[last]} = \dfrac{\partial E}{\partial x^{[last]}} \circ \dfrac{\partial x^{[last]}}{\partial z^{[last]}}$
    - $\delta^{[i]} = \left(w^{[i+1]}\right)^T \cdot \delta^{[i+1]} \circ \dfrac{\partial x^{[i]}}{\partial z^{[i]}}$
    - $w^{[i]'} = w^{[i]} - \eta \cdot \dfrac{\partial E}{\partial w^{[i]}}$
    - $b^{[i]'} = b^{[i]} - \eta \cdot \dfrac{\partial E}{\partial b^{[i]}}$

- Multiple observations:
  1. Apply forward propagation for each observation.
  2. Calculate $\delta_i^{[l]}$ for each observation.
  3. $\dfrac{\partial E}{\partial w^{[l]}} = \sum_{i=1}^{n} \delta_i^{[l]} \cdot \left(x_i^{[l-1]}\right)^T$
  4. $\dfrac{\partial E}{\partial b^{[l]}} = \sum_{i=1}^{n} \delta_i^{[l]}$
- Derivatives:

Name	Error function	$\frac{\partial E}{\partial x^{[i]}}$
Squared Error	$\frac{1}{2}\left(x^{[i]} - t\right)^2$	$x^{[i]} - t$
Cross--entropy	$-\sum_{i=1}^{n} t_i \log\left(x_i^{[i]}\right)$	$-\frac{t}{x^{[i]}} + \frac{1-t}{1-x^{[i]}}$

Name	Activation function	$\frac{\partial x^{[i]}}{\partial z^{[i]}}$
Sigmoid $\sigma(x)$	$\frac{1}{1+e^{-x}}$	$x^{[i]}\left(1 - x^{[i]}\right)$
ArcTan $\arctan(x)$	$\arctan(x)$ or $\tan^{-1}(x)$	$\frac{1}{\left(x^{[i]}\right)^2 + 1}$
Hyperbolic tangent	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - \left(x^{[i]}\right)^2$
ReLU	0 if $x < 0$ x if $x \geq 0$	0 if $x^{[i]} < 0$ 1 if $x^{[i]} \geq 0$
Softmax	$\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$	$x^{[i]}\left(1 - x^{[i]}\right)$
Sign	1 if $x \geq 0$ 0 if $x < 0$	

NOTE:

* When cross-entropy and softmax are combined:
  $\delta^{[last]} = \dfrac{\partial E}{\partial z^{[last]}} = x^{[last]} - t$

- More Derivatives:

Function	Derivative
$x \pm y$	$x' \pm y'$
xy	$x'y + y'x$
$\frac{x}{y}$	$\frac{xy' + y'x}{y^2}$

Function	Derivative
$f(x)^a$	$af(x)^{a-1}f'(x)$
$a^{f(x)}$	$a^{f(x)}f'(x)\ln a$
$\log_a f(x)$	$\frac{f'(x)}{f(x)\ln a}$

- K-Means:
  1. Assign each point to a cluster.
  2. Update centroids: $\text{centroid}_{new} = \mu$ of cluster's points
  3. Repeat until centroids don't change.
- Sillhouette: $\in$ [-1,1] (the closer to 1 the better)
  - For an observation x_i:
    * a = average distance of x_i to the points in it's cluster
    * b = average distance of x_i to points in closest cluster
    * $s_{observation} = \dfrac{b-a}{max(a,b)}$
  - For a cluster:
    Average of the cluster's observations silhouettes
  - For the solution:
    Average of the clusters silhouettes
- EM:
  - Initializaion: Initial mixture parameters
  - Expectation (E-step):
    Calculate weights for each datapoint x_i for each cluster c_k:
    $\gamma_{ki} = \dfrac{\mathcal{N}(x_i|\mu_k, \Sigma_k) \cdot \pi_k}{\sum_{j=1}^{k} \mathcal{N}(x_i|\mu_j, \Sigma_j) \cdot \pi_j}$
  - Maximization (M-step):
    Update parameters for each cluster: (for n observations)
    * $N_k = \sum_{i=1}^{n} \gamma_{ki}$
    * $\mu_k = \dfrac{1}{N_k} \sum_{i=1}^{n} \gamma_{ki} \cdot x_i$
    * $\Sigma_k = \dfrac{1}{N_k} \sum_{i=1}^{n} \gamma_{ki} \cdot (x_i - \mu_k) \cdot (x_i - \mu_k)^T$
    * $\pi_k = \dfrac{N_k}{N}$
- Quadratic Formula: $ax^2 + bx + c = 0 \Rightarrow x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
- PCA: $\Sigma u = (\lambda I)u$
  - Σ: Covariance Matrix
  - I: Identity Matrix
  - Eigenvalue (λ): $det\,|\Sigma - \lambda I| = 0$ or $\lambda = \Sigma u \circ \dfrac{1}{u}$
    Example:
    $$\lambda = \left(\begin{bmatrix} 2.917 & 2.667 \\ 2.667 & 2.667 \end{bmatrix} \begin{bmatrix} -0.690 \\ 0.723 \end{bmatrix}\right) \circ \begin{bmatrix} -0.690^{-1} \\ 0.723^{-1} \end{bmatrix}$$
    $$= \begin{bmatrix} -0.084 \\ 0.088 \end{bmatrix} \circ \begin{bmatrix} -0.690^{-1} \\ 0.723^{-1} \end{bmatrix}$$
    $$= \begin{bmatrix} 1.122 \\ 1.122 \end{bmatrix}$$
  - Eigenvector (u): $(\Sigma - \lambda I)u = \vec{0}$
    Example:
    $$u \Leftrightarrow \left(\begin{bmatrix} 1.333 & 0.667 \\ 0.667 & 1.667 \end{bmatrix} - \begin{bmatrix} 2.187 & 0 \\ 0 & 2.187 \end{bmatrix}\right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
    $$\Leftrightarrow \left[\begin{array}{cc|c} -0.854 & 0.667 & 0 \\ 0.667 & -0.52 & 0 \end{array}\right] \Leftrightarrow \begin{array}{l} L_1 \times -0.854^{-1} \\ L_2 \times 0.667^{-1} \end{array}$$
    $$\Leftrightarrow \left[\begin{array}{cc|c} 1 & -0.781 & 0 \\ 1 & -0.781 & 0 \end{array}\right] \Leftrightarrow L_2 - L_1$$
    $$\Leftrightarrow \left[\begin{array}{cc|c} 1 & -0.781 & 0 \\ 0 & 0 & 0 \end{array}\right] \Leftrightarrow x_1 = 0.781 x_2$$
    $$= \begin{bmatrix} 0.781x \\ x \end{bmatrix}$$
  - Projecting (bivariate to univariate):
    1. Choose highest λ (higher λ means more variation)
    2. Calculate eigenvector (u)
    3. Apply formula: $\phi = u^T X^T$
- Model complexity:
  - Perceptron with d inputs: $d + 1$
  - Tree with d features que tomam n valores: n^d
  - MLP: estimated by the number of weights
  - Bayesian Classifier: estimated by the number of parameters
    * With c classes: $c - 1$ *priors*
    * With d dimension: $(d\ averages + \dfrac{d(d+1)}{2}\Sigma) \times c$