- 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] → [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	B	C
Pred	A	TA	FA	FA
	B	FB	TB	FB
	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: $SSE = \sum (Z - \hat{Z})^2$
  - Maen Squared Error: $MSE = \frac{1}{n} SSE$
  - Root Maen Squared Error: $RMSE = \sqrt{MSE}$
  - Mean Absolute Error: $MAE = \frac{1}{n} \sum |Z - \hat{Z}|$
- Information Gain: $IG(y_{out}|y_i) = E(y_{out}) - E(y_{out}|y_i)$
- Entropy: $E(y) = -\sum P(x_i) \log(P(x_i))$
- 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:

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p} \qquad \|x\|_\infty = max\,|x_i|$$

- Matrix Multiplication:

$$\begin{bmatrix} \dots & \dots & n \\ \dots & \dots & \dots \\ m & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} \dots & \dots & l \\ \dots & \dots & \dots \\ n & \dots & \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & l \\ \dots & \dots & \dots \\ m & \dots & \dots \end{bmatrix}$$

- 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: $\Sigma(y_1, y_2) = \begin{bmatrix} var(y_1) & cov(y_1, y_2) \\ cov(y_1, y_2) & var(y_2) \end{bmatrix}$
  - $|\Sigma| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$
  - $P(y|\mu, \Sigma) = \dfrac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \cdot exp\left(-\dfrac{1}{2}(y - \mu)^T \Sigma^{-1} (y - \mu)\right)$
- 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: $P(x|C) = P(y_1|C) \times \ldots \times P(y_d|C)$
  4. Normalize or compare: $P(C)P(x|C)$
- 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:
  - Linear: $W = (X^T X)^{-1} X^T Z$
  - Ridge: $W = (X^T X + \lambda I)^{-1} X^T Z$
- Perceptron:
  $\hat{Z} = a(W^T X),\ \ a \leftarrow$ activation function
  If $Z \neq \hat{Z} \longrightarrow W' = W + \eta(Z - \hat{Z})X$
- Neural Networks (MLP):
  - Forward: $x^{[0]} \to z^{[1]} = w^{[1]} x^{[0]} + b^{[1]} \to x^{[1]} = a\left(z^{[1]}\right) \to \ldots \to z^{[i]} = w^{[i]} x^{[i-1]} + b^{[i]} \to x^{[i]} = a\left(z^{[i]}\right) \to E$
  - Backward:
    * $\delta^{[last]} = \dfrac{\partial E}{\partial x^{[last]}} \circ \dfrac{\partial x^{[last]}}{\partial z^{[last]}}$
    * $\delta^{[i]} = \left(w^{[i+1]}\right)^T \cdot \delta^{[i+1]} \circ \dfrac{\partial x^{[i]}}{\partial z^{[i]}}$
    * $w^{[i]'} = w^{[i]} - \eta \dfrac{\partial E}{\partial w^{[i]}}$ * $\dfrac{\partial E}{\partial w^{[i]}} = \delta^{[i]} \cdot \left(x^{[i-1]}\right)^T$
    * $b^{[i]'} = b^{[i]} - \eta \dfrac{\partial E}{\partial b^{[i]}}$ * $\dfrac{\partial E}{\partial b^{[i]}} = \delta^{[i]}$

- 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 $\tan^{-1}(x)$	$\frac{1}{\left(x^{[i]}\right)^2+1}$
Hyper. tan. $\tanh(x)$	$\frac{e^x-e^{-x}}{e^x+e^{-x}}$	$1-\left(x^{[i]}\right)^2$
ReLU $R(x)$	0 if $x<0$ x if $x\geq 0$	0 if $x<0$ 1 if $x\geq 0$
Softmax $S(x)$	$\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$	$x^{[i]}\left(1-x^{[i]}\right)$

NOTE:

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

- K-Means:
  1. Assign each point to a cluster.
  2. Update centroids: centroid$_{new}=\mu$ of cluster's points
  3. Repeat until centroids don't change.
- EM:
  - Initializaion: Initial mixture parameters
  - Expectation (E-step):
    Calculate weights for each datapoint x$_i$ for each cluster c$_k$:
    $\gamma_{ki}=\frac{\mathcal{N}(x_i|\mu_k,\Sigma_k)\cdot\pi_k}{\sum_{j=1}^{k}\mathcal{N}(x_i|\mu_j,\Sigma_j)\cdot\pi_j}$
  - Maximization (M-step):
    Update parameters for each cluster: (for n observations)
    - $N_k=\sum_{i=1}^{n}\gamma_{ki}$
    - $\mu_k=\frac{1}{N_k}\sum_{i=1}^{n}\gamma_{ki}\cdot x_i$
    - $\Sigma_k=\frac{1}{N_k}\sum_{i=1}^{n}\gamma_{ki}\cdot(x_i-\mu_k)\cdot(x_i-\mu_k)^T$
    - $\pi_k=\frac{N_k}{N}$
- Sillhouette: ∈ [-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
    - $s_{observation}=\frac{b-a}{max(a,b)}$
  - For a cluster:
    Average of the cluster's observations silhouettes
  - For the solution:
    Average of the clusters silhouettes
- More Derivatives:

Function	Derivative
u^n	$n\ u^{n-1}\ u'$
uv	$u'v+v'u$
e^u	$e^u u'$

- PCA:
  $Cu_i=(\lambda_i I)u_i$
  - C: Covariance Matrix
  - I: Identity Matrix
  - Eigenvector (u): $(C-\lambda I)u=\vec{0}$
  - Eigenvalue (λ): $det\,|C-\lambda I|=0$
  - Higher eigenvalue means more variation

  Example:

$$\left(\begin{bmatrix}1.333 & 0.667\\0.667 & 1.667\end{bmatrix}-\begin{bmatrix}2.187 & 0\\0 & 2.187\end{bmatrix}\right)\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}\Leftrightarrow$$

$$\begin{bmatrix}0.854 & 0.667\\0.667 & 0.52\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}\Leftrightarrow$$

- 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: $(d\ averages+\frac{d(d+1)}{2}\Sigma)\times c$