**Bayes’ Rule**:  
\[ P(A \land B) = P(A|B)P(B) = P(B|A)P(A) \quad \text{and} \quad P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]

If all \( n \) are independent the full join is:  
\[ P(X_1, \ldots, X_n) = \prod_{i=1}^n P(X_i) \]
If \( B \) is conditionally independent of \( C \) given \( A \):  
\[ P(B|C,A)=P(B|A) \quad \text{and} \quad P(C|B,A)=P(C|B|A) \]

**Naïve Bayes**:  

**Maximum a posteriori Hypothesis**  
\[ C_{MAP} = \text{argmax}_c P(D|c)P(c) \]

**Naïve Bayes**:  

**Smoothing**  
\[ \hat{P}(x_i | c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k} \]

**Normalization**  
\[ P(A = a_i | B = b) = \frac{P(B = b | A = a_i)P(A = a_i)}{P(B = b)} \]

\[ \ldots \]

\[ P(A = a_m | B = b) = \frac{P(B = b | A = a_m)P(A = a_m)}{P(B = b)} \]

Adding these up and noting that  
\[ \sum P(A = a_i | B = b) = 1 \]

**Decision Trees**

**Entropy**:  
\[ I(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^n -P(v_i) \log_2 P(v_i) \]

**p positive training, n negative training**  
\[ I(\frac{p}{p+n}, \frac{n}{p+n}) = \frac{p}{p+n} \log_2 \frac{p+n}{p} - \frac{n}{p+n} \log_2 \frac{n}{p+n} \]

**remainder (A)**  
\[ \sum_{i=1}^n \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}) \]

**Information Gain (A)**  
\[ I(\frac{p}{p+n}, \frac{n}{p+n}) - \text{remainder}(A) \]

**Perceptron**:  
\[ W_i = W_i + LR \cdot \Delta i \]

**Multi-layer NN**  

The Error Term for output unit \( k \):  

For weight \( w_{ik} \) between input unit \( i \) and hidden unit \( j \)

\[ \delta_O_k = o_k(E)(1-o_k(E))(t_k(E) - o_k(E)) \]

\[ \Delta_{ij} = \eta \delta O_j h_i(E) \]

The Error Term for hidden unit \( k \):  

For weight \( w_{ik} \) between hidden unit \( i \) and output unit \( j \)

\[ \delta_h = h_k(E)(1-h_k(E)) \sum_{i \in \text{out put s}} w_{ik}\delta_{O_i} \]

\[ \Delta_{ij} = \eta \delta O_j h_i(E) \]

**k-mean**

**Euclidean distance**  
\[ d(i, j) = \sqrt{\left(\frac{x_{i1} - x_{f1}}{T_{f1}}\right)^2 + \left(\frac{x_{i2} - x_{f2}}{T_{f2}}\right)^2 + \ldots + \left(\frac{x_{ip} - x_{fp}}{T_{fp}}\right)^2} \]

**Hierarchical Clustering**:  

**single link**: min; **complete link**: max

**Local search**

- Simulated annealing  
- acceptance function:  
\[ \frac{e^{-\Delta E}}{e^c} \]

**Markov Decision Process (MDP)**

- \(<S, A, P, R>\)
  - \( S \) State:  
  - \( A \) Action:  
  - \( P \) Transition function Table  
  - \( R \) Reward \( R(s) = \text{cost or reward being in state s} \)

- Optimal policy \( \pi^*(s) = \text{argmax}_a \sum_s T(s,a,s') U(s') \)

- Bellman Equation:  
\[ U(s) = R(s) + \gamma \text{max}_a \sum_s (T(s,a,s') U(s')) \]

\[ U(s) = \max_a \sum_s T(s,a,s') [R(s,a,s') + \gamma U(s')] \]

**Nash Equilibrium** - Happens when no player can benefit by changing his strategy while the other players keep theirs unchanged.

**Pareto Efficient** - Happens when there is no choice of action that can improve one person’s profit without decreasing another.

**Mixed Strategy** - Happens when each player selects a probability associated with each action.