Bag of words: unordered collection of words {"John":1, "likes":7, "viagra":3}
Term weightening: tf-idf → How important a word is to a document. tf = # in document
$idf(t,D) = \log[|D| / |\{d \in D : t \in d\}|]$ → D = all docs ⇒ $tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$ $\boxed{\text{Stop-Words}}$
Similarity of vectors: $sim(q_k, d_J) = \sum(t_{ik} \cdot t_{iJ})$. **Multinomial:** events are independ... DOC is multinom outcome of words. $P(W_1=n_1,\dots,W_k=n_k \mid N, \theta_1,\dots,\theta_k) = \frac{N!}{n_1! \cdots n_k!} \cdot \theta_1^{n_1} \cdots \theta_k^{n_k}$
Classification: KNN: needs a distance metric. Determine k. → Majority vote... between the k nearest. Possible weighting: $1/dist^2$. All computation deffered until classification... A popular lable may dominate due to it's quantity → try to weight closest items to overcome.

Association Rule Mining: {onions, potatos} ⇒ {burger}. $I = \{i_1,\dots,i_n\}$ items (boolean) $D = \{t_1,\dots,t_m\}$ transactions. Rule: $X \Rightarrow Y$ where $X, Y \in I$ and $X \cap Y = \emptyset$. MinSup, MinConf.
Support $Supp(X)$ = Proportion of transactions containing itemset X. **Confidence** $Conf(X \Rightarrow Y) = Supp(X \cup Y)/Supp(X) = P(Y|X)$. **Lift** $= lift(X \Rightarrow Y) = Supp(X \cup Y)/(Supp(X) \cdot Supp(Y))$
Lift>1 = מתאם חיובי, Lift<1 = שלילי, Lift=1 = אין מתאם. **Apriori Principle:** If an itemset is frequent, then all of its subsets must also be frequent. {A,B,C,D} $Conf(ABC \to D) \geq Conf(AB \to CD) \geq \dots (A \to BCD)$
$Lift(X \to Y) = P(Y|X)/P(X)$

→ $D(A,B) \leq D(A,C) + D(C,B)$. $Manhatan(a,b) = \sum(a_i - b_i)$
Clustering: **Distance Properties:** $D(A,B) = D(B,A)$; $D(A,A) = 0$; $D(A,B) = 0 \Rightarrow A = B$;
Euclidean: $d(X,Y) = \sqrt{\sum_{n=1}^{N}(x_n - y_n)^2}$; **Cosine Similarity:** $a \cdot b = |a| \cdot |b| \cdot \cos\theta$. Given $\vec{A}, \vec{B}$ similarity $= \cos(\theta) = (A \cdot B)/(|A| \cdot |B|) = \sum_1^n A_i \cdot B_i / \left(\sqrt{\sum_1^n A_i^2} \cdot \sqrt{\sum_1^n B_i^2}\right)$ range: (-1,...,1).
Jaccard Similarity: $J(A,B) = |A \cap B| / |A \cup B|$. dist = 1-sim... → A,B are groups.
In Dendograms: sim = height of lowest shared internal node. **Hierarchical Clustering:** results in a dendogram. **K-means clustering:** aims to partition n observations into k clusters. Given n observations $(x_1,\dots,x_n)$ each x is d dimentional vector, partition to k sets $S = \{S_1,\dots,S_k\}$ minimize the "within cluster sum of squares" $\arg\min_S \sum_{i=1}^{k} \sum_{x_J \in S_i} |x_J - \mu_i|^2$. μ_i = mean of points in S_i
The algorithm alternates betweet two steps until convergence: (1) Assignment step: $S_i^t = \{x_p : |x_p - m_i^t|^2 \leq |x_p - m_j^t|^2 \ \forall\ 1 \leq J \leq K\}$ (2) Update step: $m_i^{t+1} = \frac{1}{|S_i^t|} \cdot \sum_{x_J \in S_i^t} x_J$. since the arithmetic mean is a least-squares estimator, this also minimizes the → WCSS objective.
assumes clusters are of similar sizes. for different sizes the EM is better (genzation of K-means)
K-medoids: PAM ↔ Paratitioning Around Medoids: **Silhouette** - tool for determining K.
Algorithm: (1) Initialize: randomly select K of the n data-points as the medoids (2) Associate each data-point to the closest medoid. (3) for each medoid m: (3.1) For each non-medoid data point o (3.1.1) Swap m and o and comput the total cost of the configuration. (4) Select the configuration with the lowest cost. (5) Repeat steps 2-4 until there is no change in the medoid. ⊛ cost = סכום מרחקי כל נק' מהמרכזים שלהן
שלב **Silhouette (Clustering):** How well each object lies within its cluster. $a(i)$ is average distance of object i with all other objects within its own cluster. $b(i)$ is lowest average distance of i to all other clusters. the cluster with distance $b(i)$ is the neighboring cluster. (best fit after current cluster) ⇝ $S(i) = [b(i) - a(i)] / \max\{a(i), b(i)\}$
$-1 \leq S(i) \leq 1$. 1 is great, (-1) is worst, 0 point is on the kiss-border between the neighboring clusters
Agglomerative hierarchical Clustering: in begining each element is a cluster of its own. ($x \in X$, $y \in Y$)
Single-linkage: $D(X,Y) = \min d(x,y)$. Complete linkage: $D(X,Y) = \max d(x,y)$. Average: $\frac{1}{|X| \cdot |Y|} \cdot \sum\sum d(x,y)$
$-P(yes)\log_2 P(yes) - P(no)\log_2 P(no)$ Y -בשורש, האנטרופיה נקבעת לפי מופעי ה $-P(yes|r)\log_2 \dots$ classify
Decision Tree Learning: Data comes in records of form $(X,Y) = (X_1, X_2, X_3, \dots, X_k, Y)$
Work top-down by choosing a variable at each step that best splits the set of items. **Gini Impurity:** is a measure of how often a randomly chosen element from the set would be incorrectly labled if it were randomly labled according to the distribution of lables in the subset. Suppose i takes on values {1,2,...,m} and let f_i be the fraction of items labled with value i in the set. $I_G(f) = \sum_1^m f_i \cdot (1 - f_i) = \sum_1^m (f_i - f_i^2) = \sum_1^m f_i - \sum_1^m f_i^2 = 1 - \sum_1^m f_i^2$. **Information Gain:** $I_E(f) = -\sum_1^m f_i \cdot \log_2(f_i)$ האופציה הטובה ביותר לפיצול
Entropy: E(S) is the measure of the amount of uncertainty in the set S. מרחק קרוב/רחוק
$E(S) = -P(yes|big)\log_2 P(yes|big) - P(no|big) \cdot \log_2 P(no|big)$ → big: אחוז המשתנים שמתאימים ל yes/no: הגדולים
$IG(r, size) = E(r) - \frac{|S_{big}|}{|r|} \cdot E(S_{big}) - \frac{|S_{small}|}{|r|} \cdot E(S_{small})$ **Information Gain:** How much uncertainty in S was reduced
כלומר, מורידים מהאנטרופי שהתקבלה את סכום האנטרופי של המשקל בהתאם לכמות כל האופציות שמרכיבות את המאפיין שלפיו בודקים האם הפיצול כדאי. לשים לב לפיצול המביא גבוה
K-Medoids: את ה-Medoid -ים הראשונים קובעים באופן רנדומלי. הראשון הוא הקרוב לכולם ואחריו

Bayesian inference: Bayes rule: $P(H|E)=\frac{P(E|H)\cdot P(H)}{P(E)}$ Hypothesis, Evidence
There are competing hypotheses from which one chooses the most probable.
Naive Bayes Classifier: assumes that the presence or absence of a particular feature is unrelated to any other feature, given the class variable. The probability model for a classifier is conditional: $P(C|F_1,\dots,F_n)$ ← Class; Feature. Posterior $=\frac{\text{Prior}\cdot\text{likelihood}}{\text{evidence}}$
$P(C|F_1,\dots,F_n)=P(C)\cdot P(F_1,\dots,F_n|C)/P(F_1,\dots,F_n)$ ← Denom is Const.
$P(C|F_1,\dots,F_n)=\frac{1}{\text{const}}\cdot P(C)\cdot\prod_1^n P(F_i|C)$ ← Naive assumption. All model parameters: (class Priors and feature probability distributions) Can be approximated with relative frequencies from the training set. Class prior → ① $\frac{1}{\#C}$ equally for all or ② $\#C_i/\#C$ ↔ Relative Frequency of C_i
For discrete features → Multinomial and Bernoulli distributions are popular to assume.
Bernoulli ↔ $X\sim Ber(p)$ ↔ $P(X=1)=p$. Multinomial ↔ $P(x_1,\dots,x_k;n,p_1,\dots,p_n)=\frac{n!}{x_1!\dots x_k!}\cdot p_1^{x_1}\cdot\ldots\cdot p_k^{x_k}$.
For continuos features ↔ Gausian dist assumed. $P(X=v|C)=\frac{1}{\sqrt{2\pi\sigma_c^2}}\cdot\exp(-[v-\mu_c]^2/[2\sigma_c^2])$ MAP
Sample-Correction is needed when one of the freqs is 0 in training set. Maximum Aposteriori Decision Rule
Classify $(f_1,\dots,f_n)=\underset{c}{\operatorname{argmax}}\,P(C=c)\cdot\prod_1^n P(F_i=f_i|C=c)$. $\mu=\frac{1}{N}\sum X_i$. $\sigma^2=\frac{1}{N}\sum(X_i-\mu)^2$
Document Classification: $P(D|C)=\prod_i P(W_i|C)$. $P(D|C)=P(D\cap C)/P(C)$. $P(C|D)=P(D\cap C)/P(C)\Rightarrow P(C|D)=\frac{P(C)}{P(D)}\cdot P(D|C)$
Spam or ¬Spam: $P(S|D)=\frac{P(S)}{P(D)}\cdot\prod P(W_i|S)$. Same for ¬S. $P(S|D)/P(\neg S|D)=\frac{P(S)}{P(\neg S)}\cdot\prod\frac{P(W_i|S)}{P(W_i|\neg S)}$.
$\ln[P(S|D)/P(\neg S|D)]=\longrightarrow\ln\frac{P(S)}{P(\neg S)}+\sum_i\ln[P(W_i|S)/P(W_i|\neg S)]$. Spam if $P(S|D)>P(\neg S|D)\Rightarrow$
$\Rightarrow$ when $\ln[P(S|D)/P(\neg S|D)]>0$. **LDA-Linear discriminant analysis:** find a linear combination of features which characterizes or separates two or more classes of objects.
Fundemental assumption: the independent variables are normally distributed.
For 2 classes: Assumes $P(\vec{x}|y=0)$ and $P(\vec{x}|y=1)$ are both normally distributed with mean and covariance $(\vec{\mu}_0,\Sigma_{y=0})$ and $(\vec{\mu}_1,\Sigma_{y=1})$. Under this assumption, the Bayes optimal solution is to predict $y=1$ if the log of the likelihood ratios is below some treshold T: $(\vec{x}-\vec{\mu}_0)^T\Sigma_{y=0}^{-1}(\vec{x}-\vec{\mu}_0)+\ln(\Sigma_{y=0})-(\vec{x}-\vec{\mu}_1)^T\Sigma_{y=1}^{-1}(\vec{x}-\vec{\mu}_1)-\ln|\Sigma_{y=1}|<T$
Without further assumptions, the above is QDA. LDA assumes $\Sigma_{y=0}=\Sigma_{y=1}=\Sigma$ in this case several terms cancel and the above becomes $\vec{w}\cdot\vec{x}>c$ for some threshold constant c, where $\vec{w}\propto\Sigma^{-1}(\vec{\mu}_1-\vec{\mu}_0)\Rightarrow$ Criterion of an input $\vec{x}$ being of class y is purely a function of this linear combination of the known observations. In geometrical terms we project a multidimention space point x onto vector $\vec{w}$ (thus we only consider its direction) in other words, the observation belongs to y if corespoonding $\vec{x}$ is located on a certain side of a hyperplane perpendicular to $\vec{w}$. the location of the plane is defined by the threshold c. **Fisher's linear discriminant:** Suppose 2 classes of observations have means $\vec{\mu}_{y=0},\vec{\mu}_{y=1}$ and covariances $\Sigma_{y=0},\Sigma_{y=1}$ then the linear combination of features $\vec{w}\cdot\vec{x}$ will have means $\vec{w}\cdot\vec{\mu}_{y=i}$ and variances $\vec{w}^T\Sigma_{y=i}$ for $i=0,1$
Fisher defines the separation as the ratio of the variance between the classes to the variance within the classes: $S=\sigma^2_{\text{Between}}/\sigma^2_{\text{within}}=[\vec{w}\cdot\vec{\mu}_{y=1}-w\cdot\mu_{y=0}]^2/[\vec{w}^T\Sigma_{y=1}\vec{w}+\vec{w}^T\Sigma_{y=0}\vec{w}]$
$=[\vec{w}(\vec{\mu}_{y=1}-\vec{\mu}_{y=0})]^2/[\vec{w}^T(\Sigma_{y=0}+\Sigma_{y=1})\vec{w}]$ optimal when $\vec{w}\propto(\Sigma_{y=0}+\Sigma_{y=1})^{-1}(\vec{\mu}_{y=1}-\mu_{y=0})$
$\vec{w}$ is <u>normal</u> to the discriminant hyperplane. If projections of points from both classes have approx same distr, a good choice of c will be a hyperplane between the two means:
$c=\vec{w}\cdot\frac{1}{2}(\vec{\mu}_{y=0}+\vec{\mu}_{y=1})=\frac{1}{2}\vec{\mu}_{y=1}^t\Sigma^{-1}\vec{\mu}_{y=1}-\frac{1}{2}\vec{\mu}_{y=0}^t\Sigma^{-1}\vec{\mu}_{y=0}$.
Logistic Regression: measures the relationship between a categorical dependant variable and one or more independant variables (usualy continuos), by using Probability score as the predicted values of the dependent variable. Logistic regression is used to predict the <u>odds</u> of result success based on the values of the independant variables (predictors). The odds are defined as the probabil that a particular outcome is success devided by the probability that it's failure. **Logistic Function:**
$F(t)=e^t/(e^t+1)=1/(1+e^{-t})\in[0..1]$ where t is a linear function of an explanatory variable x.
It can be written as: $\pi(x)=\exp(\beta_0+\beta_1x)/[\exp(\beta_0+\beta_1x)+1]$ ⇐ Probability for success.
The inverse of the logisic function: $g(x)=\ln[\pi(x)/(1-\pi(x))]=\beta_0+\beta_1x$
משמעויות של פרמטרים: עבור Soft Margin SVM הפרמטר C מעלה את "מחיר הטעות" הפרדה כנאות תביא להקטנה בהטיה והגדלה בשונות. עבור KNN הפרמטר K תגדיל את מס' המשתתפים בהצבעה מה שמקטין את ההטיה ומוריד את השונות. ברגרסיה פולינומית הפרמטר הדרגה d תגדיל שונות ותוריד את ההטיה.
סיכון Bayes: סכימה רציפה על ההסת' לטעות: $P_e=\int_{R_{no}}P(x|yes)\cdot P(yes)dx+\int_{R_{yes}}P(x|no)\cdot P(no)dx$
ANN-Artificial Neural Networks: Perceptrons: outpute $\{-1,1\}$
$O(x_1,\dots,x_n)=1$ if $w_0+w_1x_1+w_2x_2+\dots+w_nx_n>0$ and (-1) otherwise. $(-w_0\cdot1)$ is the threshhold.
We imagine $x_0=1\Rightarrow O(x_1\dots x_n)=1$ if $\sum w_ix_i>0\Rightarrow O(\vec{x})=sgn(\vec{w}\cdot\vec{x})$. Perceptron training rule:
$w_i\leftarrow w_i+\Delta w_i$ where $\Delta w_i=\eta(t-o)x_i$ t=target; o=output; η=learning rate-must be small! (step)

$W \leftarrow \vec{W} + \Delta W$. where $\Delta W = -\eta \nabla E(\vec{W})$. $\Delta w_i = -\eta \cdot \frac{\partial E}{\partial w_i}$. $\boxed{\frac{\partial E}{\partial w_i} = \sum_{d\in D} (t_d - O_d)\cdot(-x_{id})}$
When we need to ensure a differentiable threshold unit: $o = \sigma(\vec{w}\cdot\vec{x})$ where $\sigma(y) = 1/(1+\exp(-y))$
Back-Propagation: Multilayer-Network. We redefine E to sum the errors over all of the network output units: $E(\vec{W}) \equiv \frac{1}{2}\sum_{d\in D}\sum_{k\in outputs}(t_{kd} - O_{kd})^2$. input/weight from i to J is denoted as x_{Ji}, w_{Ji}. ⇒ for each $\langle \vec{x}, \vec{t}\rangle$ in training examples Do: ① input the instance $\vec{x}$ to the network and compute the output O_u for every unit u in the network. ② Propagate the errors back through the network by: For each network output unit K, calculate its error term δ_K: $\delta_K \leftarrow O_K(1-O_K)(t_K - O_K)$ ③ For each hidden unit h, calculate its error term δ_h: $\delta_h \leftarrow \underline{O_h(1-O_h)}\cdot\sum_{k\in outputs} w_{kh}\delta_k$ ④ Update each network weight $W_{Ji} \leftarrow W_{Ji} + \Delta W_{Ji}$ where $\Delta W_{Ji} = \eta\delta_J x_{Ji}$. נגזרת של סיגמואיד

The EM Algorithm: used when only a subset of the relevant instance features are observable. Example: finding the means μ_1, μ_2 of $K=2$ Gaussians where σ_i^2 are known... The task is to output hypothesis $h = \langle\mu_1 \dots \mu_K\rangle$. We want h that maximizes $P(D|h)$. We can think of the full description of each instance as $\langle x_i, z_{i1}, z_{i2}\rangle$ where x_i is the observed value and z_i are indicators of where x_i came from (which gaussian). z_i are hidden variables. EM algorithm searches for a maximum likelihood hypothesis by repeatedly re-estimating the expected values of the hidden variables z_{iJ} given the current hypothesis $\langle\mu_1,\dots,\mu_K\rangle$ then recalculating the ML-hypothesis using these expected values for the hidden variables. First set $h = \langle\mu_1, \mu_2\rangle$ arbitrarly. Then, iterativly re-estimate h by repeating 2-steps until convergence. Step 1: Calculate expected value $E[z_{iJ}]$ for each hidden variable z_{iJ} assuming $h = \langle\mu_1,\mu_2\rangle$. Step 2: Calculate a new ML-hypo $h' = \langle\mu_1', \mu_2'\rangle$ assuming value taken by each hidden variable z_{iJ} is $E[z_{iJ}]$ calculated in step 1, then replace h with h′ and iterate. In our example, $E[z_{iJ}]$ is just the probability that instance x_i was generated by the Jth Gaussian: $E[z_{iJ}] = \frac{P(X=x_i|\mu=\mu_J)}{\sum_{n=1}^2 P(X=x_i|\mu=\mu_n)} = \exp(-\frac{1}{2\sigma^2}\cdot(x_i-\mu_J)^2)/\sum_{n=1}^2 \exp(-\frac{1}{2\sigma^2}(x_i-\mu_n)^2)$.
Step 2: $\mu_J \leftarrow \frac{1}{m}\sum_i^m E[z_{iJ}]\cdot x_i$. Converges to a local maximum likelihood hypothesis for $\langle\mu_1,\mu_2\rangle$
In general case: $X = \{x_1,\dots,x_m\}$ observed, $Z = \{z_1,\dots,z_m\}$ unobserved, $Y = X\cup Z$. h current, h′ revised. EM searches for ML-hypo h′ by seeking h′ that maximizes $E[\ln P(Y|h')]$. We define a function $Q(h'|h) = E[\ln P(Y|h')|h, X]$. Step 1: Estimation (E) step: Calculate $Q(h'|h)$ using the current hypo h and observed data X to estimat prob-dist over Y. $Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$
Step 2: Maximization (M) step: replace h with h′ that maximizes this Q function $h \leftarrow \underset{h'}{\arg\max}\, Q(h'|h)$

השלמות EM: Unsupervised. גישה מונחית מודל - מניחה שהנתונים מגיעים ממודל כלשהו והמטרה היא למצוא את הפרמטרים של המודל מתוך הנתונים. $N(X|\mu_K,\Sigma_K) = \frac{1}{(2\pi)^{d/2}\cdot|\Sigma_K|^{1/2}}\cdot\exp(-\frac{1}{2}(x-\mu_K)^T\cdot\Sigma_K^{-1}(x-\mu_K))$
נגדיר r_{ik} משתנה בינארי שמקבל ערך 1 אם התצפית x_i לקוחה מהרכיב הגאוסיאני ה-K ו-0 אחרת. $r_{ik}\in\{0,1\}, \sum_k r_{ik} = 1$. ניתן לחשוב על r_{ik} כעל משתנה נסתר שמבטא לאיזה רכיב במודל התצפית ה-i שייכת.
מכאן: $P(x_i) = \sum_K P(x_i, r_{iK}) = \sum_K P(r_{iK})\cdot P(x_i|r_{iK}) = \sum_K \pi_K N(x_i|\mu_K,\Sigma_K)$. $\gamma(r_{iK}) = P(r_{iK}=1|x_i) = \frac{P(r_{iK})\cdot P(x_i|r_{iK})}{\sum_{J=1}^K P(r_{iJ})\cdot P(x_i|r_{iJ})} = \frac{\pi_K N(x_i|\mu_K,\Sigma_K)}{\sum_{J=1}^K \pi_J N(x_i|\mu_J,\Sigma_J)}$
$\gamma(r_{iK})$ הינו ה-Posterior של המשתנה r_{iK} וניתן לחשוב עליו גם כעל ה-Responsabl שהרכיב ה-K לוקח בלהסביר את התצפית ה-i.
את התצפית x_i נשייך ל-Cluster ה-K שמקיים: $K = \arg\max\, \gamma(r_{iJ})$
Dataset log-likelihood: $\ln P(X|\pi,\mu,\Sigma) = \sum_{i=1}^N \ln\{\sum_K \pi_K N(x_i|\mu_K,\Sigma_K)\}$ ועלינו למצוא אומדי MLE עבור הפרמטרים $\{\pi_K,\mu_K,\Sigma_K\}_{K=1}^K$
אלגוריתם ה-EM לחישוב האומדנים: ① אתחל את הערכים הנ"ל באופן אקראי ② (E-step) חשב את ערכי האחריות: $\gamma(r_{ik}) = \frac{\pi_K N(x_i|\mu_K,\Sigma_K)}{\sum_{J=1}^K \pi_J N(x_i|\mu_J,\Sigma_J)}$ ③ (M-step) אמוד את הפרמטרים באמצעות האומדים המפורשים: $N_K = \sum_{i=1}^N \gamma(r_{ik})$, $\mu_K^{new} = \frac{1}{N_K}\sum_{i=1}^N \gamma(r_{iK})x_i$, $\pi_K^{new} = N_K/N$. ④ $\Sigma_K^{new} = \frac{1}{N_K}\sum_{i=1}^N \gamma(r_{iK})(x_i-\mu_K^{new})(x_i-\mu_K^{new})^T$ אם האלגוריתם התכנס עצור ואחרת חזור ל-2.

השלמות Decision Trees: N - Number of observations, K - Number of classes, N_t - Number of observations at node t, N_t^K - Number of observations from class K at node t, $\hat{P}(K|t)$ - proportion of observations from class K at node t. $\hat{P}(K,t) = N_t^K/N_t$. $Y(t)$ - class assigned to the terminal node t. Entropy$(t) = -\sum_{K=1}^K \hat{P}(K|t)\log_2(\hat{P}(K|t))$. Gini Index$(t) = 1 - \sum_{K=1}^K \hat{P}^2(K|t)$
Misclassification error$(t) = 1 - \max_{K=1\dots K}\{\hat{P}(K|t)\}$, $IG(S,A) = \text{Entropy}(S) - \sum_{v\in Val(A)}\frac{|S_v|}{|S|}\text{Entropy}(S_v)$.

השלמות הטייה/שונות: מורכבות עולה - שונות עולה. Overfitting עולה - שונות עולה.
שונות = כמה המודל ישתנה אם אוסיף תצפית חדשה ואבנה שוב את המודל. הטייה: כמה אני טועה על מדגם האימון.
מקרה פרטי של רגרסיה לוגיסטית

השלמות רגרסיה לוגיסטית: ממדלים ישירות את $P(Y|X)$ לפי: $P(Y=1|X=x)=h(w^Tx)=\frac{1}{1+e^{-w^Tx}}$

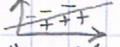
נסווג את x למחלקה 1 אם $P(Y=1|X=x)\geq P(Y=0..)$ כלומר $\log[P(Y=1|x)/(1-P(Y=1|x))]\geq 0$ $P(Y=1|X)/P(Y=0|X)\geq 1 \Rightarrow$

We determine $\hat{w}$ With MLE: $L(w)=\log\prod_{i=1}^{m}P(Y=1|X_i,w)^{y_i}\cdot P(Y=0|X_i,w)^{1-y_i}$

שיטת ניוטון-רפסון

שיטה למציאת שורשים של פונקציה ← $X_{n+1}\leftarrow X_n-\frac{f(x)}{f'(x)}$. לאחר בניית המודל יש לקבוע τ אופטימאלי עבור $P(Y=1|X)=h(w^Tx)\geq\tau$ if $g(x)=1$ נבחן את כל ה-τ האפשריים ונבחר בזה שיש לו Sensitivity & Specificity מקסימלי"?

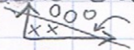
Specificity = True negative/(True neg+false pos). Sensitivity = True positive/(true positive+false negative) for the τ

הערכות והבהרות: K-medoids ← לעומת מול מטריצת המרחקים לפי טורים, בכל שלב נבדוק לכל תצפית לאיזה מדואיד היא הכי קרובה. לרשום לכל קונ' את העלות שלה. MBA ← יחס הקשר "חזק" איננו סימטרי לדוגמה כי ה-Support איננו סימטרי. KNN ← כאשר יש המון נתוני אימון ישנה בעיית אחסון ובעיית חישוב כי יש להתחשב בכל התצפיות ולא רק חלקן. Gausian Mixture והאופן כלל' Soft limit Clust ← ההשמה היא הסתברותית ולכן גם אם המרכזים שלו מתלכדים עם אלגו' לא כל התצפיות הגבולות עלולות לקבל סיווג שונה. Hierarch-Clustering שינוי פונ' מרחק יכול להביא לתוצאות שונות לחלוטין ולא רק בצורת איחוד העצים. עצי החלטה ← האנטרופיה לפני שמתרגלים לפצל כה פחות סכום האנטרופיות של ערכי הפיצול Y. בכל צומת החלטה $E(yes)+E(No)$ המכנה של האנטרופיה הוא כמה ערכי Y_i יש עבור הערך הספציפי של Attr המפצל לדוגמה כמה Y_i ק"מ (בין אם yes או no) עבור Size=big... והמונה הוא כמה ערכי Y ספציפיים יש לאותה הגבלה לדוגמה כמה תוצאות Y=yes קיימות עבור big = size. עבור Attr בעל ערכים רציפים ← ממיינים את נתוני האימון לפי הערך הרציף ובוחרים את נק' ההפרדה (ע"י ממוצע) בין 2 ערכים שעבורם משתנה הערך של Y. לפעמים מפריד ליניארי עובד טוב יותר בסיווג מעל, לדוגמה: 

Bayes Classifier ← כלל בייס אופטימאלי במקרה של מטריצת קנסות על חיזוי $\begin{bmatrix}0&\alpha\\ \beta&0\end{bmatrix}$ הוא לסווג ל-1 אם $P(1|X)>\frac{\alpha}{\alpha+\beta}$ וזה שקול לסיווג למחלקה 1 אם $\beta\cdot P(1|X)>\alpha\cdot P(2|X)$. Naive Bayes ← $\prod_i$ בפועל ההסתברות של תצפית ? ב נתוני האימון. צריך לחשב את כל ההסתברויות $P(X_1|y)$ ו- $P(X_2|y)$ לכל הערכים האפשריים של X ו- y ולבסוף לחשב את ההסת' הסופי' ולבחור את המקסימליות: $\prod_i f(x_1,x_2)$. גם אם יש לנו אינסוף נתוני אימון להם עדיין מודל הסתברותי ולכן תמיד תהיה שגיאה בלתי נמנעת. גם לא כל ההתפלגויות ניתן למדל ע"י Bayes אפילו אם כל התנאים מתקיימים. ML/MAP ← ML ממקסם $P(Data|\text{model params})$ לעומתו MAP ממקסם $P(\text{model params}|data)$. ID3 ← למדד ה- IG יש הטייה לטובת תכונות מרובות ערכים לדוגמה האוצר' של ID הוא אפס אבל זה פיצול גרוע. (אוברפיטינג) ככל שמספר העלים קטן יותר כך מודל העץ פשוט יותר הטייה גדולה והשונות קטנה. רגרסיה לוגיסטית ← τ מתפקד לקביעת y: $Y=1$ if $\hat{y}>\tau$ יחס הסיכויים הוא: $P(Y=1|\ldots)/P(Y=0|\ldots)=(1+e^{-w^T\ldots})^{-1}/[e^{-w^T\ldots}(1+e^{-w^T\ldots})^{-1}]=e^{w^T\ldots}$ ולכן אם נגדיל את w_2 ב-2 יחידות (בתצפית כלשהי) השינוי יהיה: $e^{w^T\ldots}\cdot e^{2w_2}=e^{w^T\ldots}$ היחס יגדל פי e^{2w_2}. כדי למצוא את ערך ה-τ האופטימאלי נספר את הניבויים במטריצת הבלבול:

	True1	True0
Pred1	a	b
Pred0	c	d

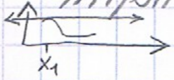
$Sensitivity=\frac{a}{a+c}$, $Specificity=\frac{d}{b+d}$

עקומת ROC הציר y: Sensitivity והציר x: 1-Specificity עבור כל ערך של τ ונבחר את ה-τ שהכי למעלה ושמאלה. **תכונות:** SVM ← מפריד ליניארי (גם עם קרנל, עדיין ליניארי) המפריד הוא $wx+b=0$ . PCA ← זהו לא מסווג אלא שיטה להתבוננות בנתונים ע"י הורדת מימד תוך שמירה מקס' על הפיזור המקורי שלהן לצירים שונים. LDA ← רוצים קו ישר שכשהתצפיות מוטלות עליו הן מופרדות באופן ברור.

מיקסום ← ב LDA ממקסמים $L(X|Y)$ ואת $L(Y)$. ברגרסיה לוגיסטית ממקסמים $L(Y|X)$.

פונקציית נראות: מכפלת ההסתברויות. כדי למצוא MLE גוזרים או לוקחים log וגוזרים או בלי log.

תכונות: ב-KNN אין תהליך בניית מודל. **שיטת ניוטון רפסון:** משתמשים בה לקירוב שורשי פונקציה.

(1) מנחשים נק' התחלה באופן אקראי $x_1=x$ ואם זה לא שורש אזי ממשיכים לעדכן את הנקודה לפי הכלל $X_{n+1}=X_n-f(X_n)/f'(X_n)$ עד שמתקרבים מספיק ל-0 ל- $y=0$. עלול לא לעבוד: 

x_1

כשהמשיק לא חותך את ציר x.