Dimensions
Dims typically: $X \in \mathbb{R}^{n\times d}$, $w \in \mathbb{R}^d$, $y \in \mathbb{R}^n$, $\hat{y} = Xw$, Gram $X^TX \in \mathbb{R}^{d\times d}$, Cov $XX^T \in \mathbb{R}^{n\times n}$

Solutions:
OLS: $w^* = (X^TX)^{-1}X^Ty$ for $\text{argmin}_w ||Xw-y||_2^2$
Ridge: $w^* = (X^TX + \lambda I_d)^{-1}X^Ty$, kern $w^* = \Phi^T(\Phi\Phi^T + \lambda I_n)^{-1}y$, for $\text{argmin}_w ||Xw-y||_2^2 + \lambda||w||_2^2$
GLS: $w^* = (X^T\Sigma_Z^{-1}X)^{-1}X^T\Sigma_Z^{-1}y$, if indep noise GLS is WLS with $\Sigma_Z^{-1} = \Omega^{-1}$, for $\text{argmin}_w ||\Sigma^{-\frac{1}{2}}Xw-y||_2^2$
GLS w/prior: $w^* = \mu_W + (X^T\Sigma_Z^{-1}X + \Sigma_W^{-1})^{-1}X^T\Sigma_Z^{-1}(y - X\mu_W)$
TLS: $w^* = (X^TX - \sigma_{d+1}^2 I_d)^{-1}X^Ty$, for $\text{argmin}\,||[\epsilon_x\epsilon_y]||_F^2$, $(X+\epsilon_x)w = y + \epsilon_y$
MLE $= \text{argmax}_w\, P(Y|X,w)$, MAP $= \text{argmax}_w\, P(Y|X,w)P(w) \implies \lambda = \frac{\sigma^2}{\sigma_h^2}$
Use for ridge proofs: $\nabla_w(||Xw-y||_2^2 + \lambda||w||_2^2) = (Xw-y)^TX + \lambda w^T = X^TXw - X^Ty + \lambda w$

Bias Variance Decomp:
Bias-Var decomp: $E[(h(x;D)-Y)^2] = E[(h(x;D)-f(x))^2] + V[h(x;D)] + V[N]$
$((\text{bias})^2 + \text{variance of method} + \text{irreduce error})$
For $\hat{X}$, X', $E[(\hat{X}-X')^2] = (E[\hat{X}]-\mu)^2 + \text{Var}(\hat{X}) + \sigma^2$
$E[(\hat{X}-\mu)^2] = (E[\hat{X}]-\mu)^2 + \text{Var}(\hat{X})$
best num of 0's to inject $n_0 = \alpha n$ for $\alpha = \frac{\sigma^2}{n\mu^2}$
$|A\|_{\text{F}} = \sqrt{\sum_{i=1}^m\sum_{j=1}^n |a_{ij}|^2} = \sqrt{\text{trace}\,(A^*A)} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2(A)}$
For these problems, try adding and subtracting μ to then get in the form of vars and then the desired form.

Linear alg review
$Ax = \sum_i^n x_i a_i$ for cols a_i
eig: $|A-\lambda I| = 0$, $\det(A) = \prod_i \lambda_i$, $\det(AB) = \det(A)\det(B)$
for orth Q: Q^TQ^{-1}, $(Qx)^T(Qy) = x^Ty$
PSD matrices A: all $\lambda \geq 0$, $\forall x : x^TAx \geq 0$, $\exists U \in \mathbb{R}^{d\times d} : A = UU^T$
isocont for PD A: $f(x) = x^TAx$ are ellipse w/ axes A eigenvecs v_i and lens $\sqrt{\lambda_i}$
SVD: $A \in \mathbb{R}^{m\times n}$ has SVD $A = U\Sigma V^T = \sum_i^r \sigma_i u_i v_i^T$ with unitary $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ which form orthonorm basis.
First rank$(A) = r$ $\sigma_i \geq 0$, cols of V are eigenvec of A^TA and cols of U are eigenvec of AA^T
Fund theorem of lin alg for SVD of A with rank$(A) = r$:
Subspace Columns
range(A) The first r columns of U
range(A^T) The first r columns of V
null(A^T) The last m r columns of U
null(A) The last n r columns of V
Orthog proj: For any $v \in \mathbb{R}^n$, $S \subset \mathbb{R}^n$, $v = v_S + v_\perp$ where $v_s \in S$ and $v_\perp \in S^\perp$
col space: range(A), row space: range(A^T).
tria ineq; $|x+y| \leq |x| + |y|$, CS ineq: $|\langle x,y\rangle| \leq ||x||||y||$, $(\text{E}[XY])^2 \leq \text{E}[X^2]\cdot\text{E}[Y^2]$
best rank k approx for $A \in R^{m\times n}$ is $A_k = \sum_i^k \sigma_i u_i v_i^T$

Stat review:
$\mathbf{1}_A(x) := 1$ if $x \in A, 0$ if $x \notin A$, $\text{E}(\mathbf{1}_A) = \int_X \mathbf{1}_A(x)\,d\,\text{P} = \int_A d\,\text{P} = \text{P}(A)$
$\text{Var}(X) = \text{E}\left[(X-\mu)^2\right] = \text{E}\left[X^2\right] - \text{E}[X]^2$
$\text{Cov}(X,Y) = \text{E}\left[(X-\text{E}[X])(Y-\text{E}[Y])\right] = \text{E}\left[XY\right] - \text{E}\left[X\right]\text{E}\left[Y\right]$, $\rho = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\,\text{Var}(Y)}}$
$\text{Var}(aX+bY) = a^2\,\text{Var}(X) + b^2\,\text{Var}(Y) + 2ab\,\text{Cov}(X,Y)$
Markov Ineq: $\text{P}(X \geq a) \leq \frac{\text{E}(X)}{a}$
Marginal distrib for joint dist X,Y : $P(X) = \sum_y P(X,y)$

Gaussians:
If $X \sim \mathcal{N}(\mu,\sigma^2)$: PDF $= \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, MGF:$= E\left[e^{tX}\right] = e^{\mu t - \sigma^2t^2/2}$
$Z = (Z_1 \ldots Z_k)$ is JG random vec if $U = (U_1 \ldots U_\ell), U_i \sim \mathcal{N}(0,1)$ $R \in \mathbb{R}^{k\times\ell}$, $\mu \in \mathbb{R}^k$, and $Z = RU + \mu$.
Or if $\sum_i a_i Z_i$ is norm disrib for every $a \in \mathbb{R}^k$
Or, in non-degen case: $f_{\mathbf{Z}}(z) = \frac{\exp\left(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^{\text{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$, $\Sigma = E[(Z-\mu)(Z-\mu)^T]$
isocontours of mulitvariate Gaussian are ellipsoids with axes $v_i\sqrt{\lambda_i}$ for eigen of covariance matrix

Kernels:
kernel $k(x_i,x_j)$ is valid if feat map $\phi(.)$ so $\forall x_i,x_j$, $k(x_i,x_j) = \langle\phi(x_i),\phi(x_j)\rangle$ or gram mat $K(D)$ is PSD for any $D = \{x_1 \ldots x_n\}$
$K = \Phi^T\Phi$, $v^T\Phi^T\Phi v = ||\Phi^Tv||^2 \geq 0$
for PSD Σ, $k = \phi(x_i)\Sigma\phi(x_j)^T$ is valid kern, $\tilde{\phi} = \Sigma^{\frac{1}{2}}\phi(x_i)$
get $\hat{y}_{\text{ridge}}$ for $\ell \to d$ dims, $O(d^3 + d^2n)$ non-kern vs kern $O(n^3 + n^2(\ell + \log p))$ if $d \ll n$, non-kern is better. Elif $n \ll d$, kern better.

CCA:
$u = W_xD_xu_d$, $W_x = U_xS_x^{-1/2}U_x^T$, D decorrelates.

PCA:
PCA first component $\mathbf{w}_{(1)} = \arg\max\left\{\frac{\mathbf{w}^T\mathbf{X}^\mathbf{T}\mathbf{X}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}\right\}$ which is achieved when $\mathbf{w}_{(1)}$ is a unit eigenvec of X^TX with largest eigenval.

kth component found by subtracting $k-1$ comps: $\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^{\mathrm{T}}$ and then $\mathbf{w}_{(k)} = \arg\max \left\{ \frac{\mathbf{w}^T \hat{\mathbf{X}}_k^T \hat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$
PCA proj is $Z_k = XV_k$ where cols of V_k are k loading vecs of X and can approx reconstruct $\tilde{X}_k = Z_k V_k^T = XV_kV_k^T$
Data is uncorrelated in proj space

Optimization:
Convex if H is PSD, concave if NSD, saddle at crit pt if mixed eigen. $H_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$
GD: $w^{(t+1)} \leftarrow w^{(t)} - \alpha_t \nabla f(w^{(t)})$, line search: dir $u^{(t)}$ and step size α_t that maxes f: $w^{(t+1)} \leftarrow w^{(t)} - \alpha_t u^{(t)}$,
Newt: $w^{(t+1)} \leftarrow w^{(t)} - \nabla^2 f((w^{(t)})^{-1} \nabla f(w^{(t)})$
Gauss Newt (NLLS): $w^{(k+1)} \leftarrow w^{(k)} + (J^TJ)^{-1}J^T\Delta y$, $\Delta y = y - F(w^k)$ for F first order approx of $f(w^k)$, $J_{ij} = \frac{\partial f_i(w^k)}{\partial x_j}$, $J^TJ \approx H$
$F = F(w^{(k)}) + \frac{\partial}{\partial w} F(w^{(k)})(w - w^{(k)}) = F(w^{(k)}) + J(w^{(k)})\Delta w$

Neural Nets:
sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$, $\sigma'(z) = \sigma(z)(1-\sigma(z))$, softmax $\sigma(z)_j = \frac{e^{z_j}}{\sum_{i=1}^d e^{z_i}}$ $\tanh(z) = \frac{\sinh}{\cosh} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $\tanh' = \mathrm{sech}^2 = 1 - \tanh^2 = 4\sigma'(2x)$
$a_{i+1} = \sigma(W_i a_i + b_i)$, $\frac{\partial \ell}{\partial a_i} = \frac{\partial \ell}{\partial a_{i+1}} \frac{\partial}{\partial a_i}(\sigma(W_i a_i + b_i)) = \frac{\partial \ell}{\partial a_{i+1}} \sigma'(W_i a_i + b_i) W_i$, $\frac{\partial \ell}{\partial w_{ii}} = \frac{\partial \ell}{\partial z_j} a_i$

Generative: (LDA/QDA), Discriminative: (log reg)

Log reg
$\hat{y} = \max_k P(\hat{Y} = k|x, w) = 1$ if $s(w^Tx) \geq 0.5, 0$ else , equiv 1 if $w^Tx \geq 0$
for $p_i = s(w^Tx_i)$, cross ent : $L(w) = -\sum_{i=1}^n y_i \ln p_i + (1-y_i)\nabla(1-p_i)$, obtained via MLE on $P(\hat{Y}_i = y_i) = p_i^{y_i}(1-p_i)^{(1-y_i)}$
$\nabla_w L(w) = -\sum_{i=1}^n (y_i - p_i)x_i$
$D_{KL}(P||Q) = \sum_x P(x) \ln(\frac{P(x)}{Q(x)})$, $D_{KL}(P(Y_i)||P(\hat{Y}_i)) = H(P(Y_i), P(\hat{Y}_i)) - H(P(Y_i))$, $H(P(Y_i)) = -y_i \ln y_i - (1-y_i)\ln(1-y_i)$
multiclass: $L(W) = -\sum_{i=1}^n \sum_{j=1}^K \delta_{j,y_i} \cdot \ln P(\hat{Y}_i = j|x_i, W)$, $\nabla_w L(w) = -\sum_{i=1}^n \delta_{\ell,y_i} - P(\hat{Y}_i = \ell)x_i$

GDA
prior $P(k) = \frac{n_k}{n}$, let $Q_k(X) = \ln(\sqrt{2\pi})^d P(k) p_k(x)$. $\hat{y} = Q_k(X)$
$\hat{\mu_k} = \frac{1}{n_k} \sum_{i:y_i=k} x_i$, $\hat{\Sigma_k} = \frac{1}{n_k} \sum_{i:y_i=k} (x_i - \hat{\mu_k})(x_i - \hat{\mu_k})^T$, for LDA $\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu_{y_i}})(x_i - \hat{\mu_{y_i}})^T$

Clustering and EM
softmax $= \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^d e^{z_k}}$
k-Means $\hat{c_k} = \mathrm{argmin}_{c_k} \sum_{x \in C_k} ||x - c_k||^2 = \frac{1}{|C_k|} \sum_{x \in C_k} x$
soft k-means $\hat{c_k} = \mathrm{argmin}_{c_k} \sum_{i=1}^N r_i(k)||x_i - c_k||^2 = \frac{\sum_{i=1}^N r_i(k) x_i}{r_i(k)}$, $r_i(k) = \sigma(z)_k$, $z = -\beta||x_i - c_k||^2$
EM for MOG (update for $t+1$): E: $q(z_i = k|x_i)^{t+1} = p(z_i = k|x_i; \theta^t) = \frac{\alpha_k^T p(x_i|z_i=k;\theta^t)}{\sum_{j=1}^k \alpha_j^t p(x_i|z_i=j;\theta^t)}$ M : $\mu_l^{t+1} = \frac{\sum_{i=1}^N q_{k,i}^{t+1} x_i}{\sum_{i=1}^N q_{k,i}^{t+1}}$,
$\Sigma_k^{t+1} = \frac{\sum_{i=1}^N q_{k,i}^{t+1} (x_i - \mu_k^{t+1})(x_i - \mu_k^{t+1})^T}{\sum_{i=1}^N q_{k,i}^{t+1}}$, $\alpha_k^{t+1} = \frac{1}{N} \sum_{i=1}^N q_{k,i}^{t+1}$

SVMs:
hard $\min_{w,b} \frac{1}{2}||w||_2^2$ s.t. $y_i(w^Tx_i - b) \geq 1 \forall i$
soft $\min_{w,b,\xi_i} \frac{1}{2}||w||_2^2 + C\sum_{i=1}^n \xi_i$ s.t. $y_i(w^Tx_i - b) \geq 1 - \xi_i \forall i$ and $\xi_i \geq 0 \forall i$
large C keeps ξ_i small or zeor but can overfit and is sensitive to outliers. small C will tend to max margin but may underfit, it is less sensitive to outliers. Can be formulated as empirical risk minimilazation as
$\min_{w,b} C\sum_{i=1}^n \max(1 - y_i(w^Tx_i - b), 0) + \frac{1}{2}||w||^2$ dividing by Cn, we see it is reg regress with $\lambda = \frac{1}{2Cn}$
$L_{\text{Hinge}}(y, w^Tx - b) = \max(1 - y(w^Tx - b), 0)$

k-NN:
bias $\frac{1}{k}\sum_{i=1}^k f(x_i) - f(z)$, var $= \frac{\sigma^2}{k}$
curse of dim $\frac{(r-\epsilon)^d}{r^d} = (1 - \frac{\epsilon}{r})^d \approx e^{-ed/r} \to 0$ as $d \to \infty$
Improve: obtain more training data, or reduce dimensionality. Consider other c dist functs.

Sparsity:
for SVM case, let's see how changing some arbitrary slack variable ξ_i affects the loss. A unit decrease in ξ_i results in a "reward" of C, and is captured by the partial derivative $\frac{\partial L}{\partial \xi_i}$. No matter what ξ_i is, reward for decreasing ξ_i is constant. Of course, decreasing ξ_i may change the boundary and thus the cost attributed to the size of the margin $||w||^2$. The overall reward for decreasing ξ_i is either going to be worth the effort (greater than cost incurred from w) or not worth the effort (less than cost incurred from w). ξ_i decrease until it hits a lower-bound "equilibrium" - which is often just 0. For ℓ_2 regularization, the reward is $2C\xi_i$ so we get diminishing returns and decreasing ξ_i causes increase in $||w||^2$ cost, so there will be ξ_i^* threshold where decreasing further will no longer outweigh the cost incurred by the size of the margin, and it will not reach zero. Basically same argument can be made for LASSO vs ridge (λ vs $2\lambda w$ reward).
Lasso single coord $\hat{w_i} = \frac{\lambda - \sum_{j=1}^n 2X_{j,i}r_j}{\sum_{j=1}^n 2X_{j,i}^2}$ if $w_i > 0$ and $\hat{w_i} = \frac{-\lambda - \sum_{j=1}^n 2X_{j,i}r_j}{\sum_{j=1}^n 2X_{j,i}^2}$ if $w_i > 0$ if $\lambda \geq 2|\sum_{j=1}^n X_{j,i}r_j|$, $\hat{w_i} = 0$. $r := \sum_{j \neq i} w_j X_j - y$

Decision Trees:
Gini measures how often a randomly chosen element would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. Gini impurity can be computed by summing the probability p_i of an item with label i being chosen times the probability $\sum_{k \neq i} p_k = 1 - p_i$ of a mistake in categorizing that item. $G = \sum_{i=1}^J p_i \sum_{k \neq i} p_k = 1 - \sum_{i=1}^J {p_i}^2$
Decision tree algorithm:
Create node N
Check stopping conditions

if met, return N as a leaf node labeled with majority class
Apply attribute selection method to find best splitting criterion (Gini or Info)
Label node with splitting criterion
If splitting attribute is nominal/categorical and multiway splits are allowed then
remove attribute from attribute list
for each outcome of j of splitting criterion
let D_j be the set of data tuples in D satisfying outcome j
if D_j is empty then
attach a leaf labeled with the majority class in D to node N
else attach the node returned by generate_decision_tree to node N
return N
Pruning: $\frac{\text{err}(\text{prune}(T,t),S)-\text{err}(T,S)}{|\text{leaves}(T)|-|\text{leaves}(\text{prune}(T,t))|}$
Surprise $= -\log P(Y=k)$, Entropy $H = \mathbb{E}[-\log P(Y=k)] - \sum_k P(Y=k)\log P(Y=k)$

ADABoost:
assume $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$
init point weights to $\frac{1}{n}$
for $m = 1, \dots M$:
Build classifier $G_m : \mathbb{R}^d \to -1, 1$ where in training data are weighted by w_i. compute weighted error $e_m = \frac{\sum_{\text{missclass}\,i} w_i}{\sum_i w_i}$
$w_i \leftarrow \sqrt{\frac{1-e_m}{e_m}}$ is missclass $\sqrt{\frac{e_m}{1-e_m}}$ else.

Error Metrics:
Confusion Matrix: $\begin{bmatrix} & \textbf{True} & \textbf{False} \\ \textbf{Pred} & \text{TP} & \text{FP} \\ \textbf{¬Pred} & \text{FN} & \text{TN} \end{bmatrix}$ Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$, Recall (TPR) $= \frac{TP}{TP+FN}$, Specificity (TNR) $= \frac{TP}{TP+FP}$, Precision (PPV) $= \frac{TP}{TP+FP}$, NPV $= \frac{TN}{TN+FN}$, $F_1 = 2 \cdot \frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$ For binary classification, ROC curve is the parametric plot created by changing the classification threshold of TPR plotted against FPR (1 - Specificity). It's integral AUC $= \int_{x=0}^{1} \text{TPR}(\text{FPR}^{-1}(x))\,dx$ represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. It is related to the Gini coefficient by $G = 2\text{AUC} - 1$
Precision-Recall curves may be more useful in practice if you only care about one population with known background probability and the "positive" class is much more interesting than the "negative" class or if there is class imbalance. Precision is not conditioned on the true class distribution (rather, on the estimate of it)

Ensembling:
Simple combiner: Combine predictions through simple averaging or other non-trainable combiner (mean, min, max). Bagging: Diversifying a model by bootstrapping the training set and averaging the predictions (aka bootstrapped aggregation). Stacking uses cross validation, blending uses hold out validation set.