- Boosting → trees are grown sequentially: each tree is grown using information from previously grown trees. Boosting does not involve bootstrap: instead each tree fit on a modified version of original dataset $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$, $r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$ (B times), $\hat{f}(x) = \sum \lambda \hat{f}^b(x)$
  - learning rate: small λ requires large B.
  - B: also controls the complexity

- Bayesian Additive regression tree (BART): we grow trees successively, but each one is perturbed to avoid local optima and achieve more thorough exploration. We try to improve the fit to the current partial residuals by modifying the tree → guard against overfitting

③

SVM
- in a p-dimensional space, a hyperplane is a flat affine subspace of dimension p−1

Maximize M
$B_0 \dots B_p, \epsilon_1 \dots \epsilon_n, M$
subject to $\sum_{j=1}^{p} B_j^2 = 1$, $y_i(B_0 + B_1 x_{i1} + \dots + B_p x_{ip}) \geq M(1-\epsilon_i)$, $\sum \epsilon_i \leq C$
- if C > 0 no more than C observation on wrong side of hyperplane
- C is small: narrow margin: rarely violated, low bias high variance.
- $\epsilon_i = 0$ right side, $\epsilon_i > 0$ wrong side margin, $\epsilon_i > 1$ wrong side hyperplane
- non linear class → linear kernel, polynomial kernel, radial kernel
- one versus one, one versus all → more than two classes

③

- carries observations on the features / on the basis of observation (transpose matrix).
- K means → reduce variance intra clusters → local optima → should be repeated.
  - → random mean → obt to closest mean → new means → repeat
- hierarchical: Hierarchical structure (the best division into 3 groups may not come from taking the best division into 2 groups and splitting up one of them)
- complete (Max intercluster), single (Min intercluster), Average (Mean intercluster), centroid

④

	Truth: H0	H0
Reject H0	I	Power (OK)
Do not reject	OK	II

- Decision
- → I: uncorrectly rejecting H0
- → II: prob of not making II → correctly rejecting H0
- Family-wise error rate (FWER) → prob making at least of type I: $= 1 - (1-\alpha)^m$
  - → Control FWER with Bonferoni → $FWER(\alpha/m) \leq m \frac{\alpha}{m} = \alpha$ (conservative)
  - → Holm: less conservative, reject more null H0 resulting in fewer II and greater power. The threshold to reject each H0 P(k) depend on the value of all p-values
- trade off between FWER and power. In practice when m is large, we may be willing to tolerate a few false positive, in the interest of making more discoveries: i.e rejections of the null hypotheses, this is the motivation behind false discovery rate
  - → we look at the ratio of false positive (V) to total positive (V+S=R) is sufficiently low → FDR = E(FDP) = E(V/R)
  - → Benjamini-Hochberg procedure gives us a very easy way to determine, given a set of m p-values, which null hypotheses to reject in order to control FDR
- why use resampling: no theoretical distribution or its assumption don't hold.

- $Y = f(x) + \epsilon$ (systematic info of x on Y; random error)
  - $Y = f(x)$, $\hat{Y} = \hat{f}(x)$
  - $\hat{Y} = \hat{f}(x)$ (our model)
- Accuracy of $\hat{Y}$ → $E[(Y-\hat{Y})^2]$ → $E[f(x) + \epsilon - \hat{f}(x)^2]$ → $(f(x) - \hat{f}(x))^2 + Var(\epsilon)$
  - reducible: use better model
  - irreducible: lower bound.
- $E(Y-\hat{Y})^2$ → $Var(\hat{f}(x)) + Bias(\hat{f}(x))^2 + Var(\epsilon)$ → $Bias^2 + Variance$
- $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$

①

- Subset selection → Lasso → Least squares → generale additive models → trees
- Bagging, Boosting, Support vector machine → deep learning.
- Variance = how model change if we use other data
- Bias = error introduced by the model → approximate something hard by simpler.
- Classification → error rate → $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$ → most likely class
  - $\max_j Pr(Y=j \mid X=x)$ → $1 - \max_j Pr(Y=j \mid X=x)$ → more to Bayes classifier
  - → $1 - E(\max_j Pr(Y=j \mid X=x))$
  - → But suppose to have $Pr(Y=j \mid X=x)$ → so use KNN → $Pr(Y=j \mid X=x_0) = \frac{1}{K}\sum_{i \in N_0} I(y_i=j)$

②

- least square minimize (LS) the residual sum of squares (RSS) $= e_1^2 + \dots e_n^2$
- Unknown → residual standard error (RSE) $= \sqrt{RSS/(n-2)}$ with $\sigma^2 = Var(\epsilon)$ = average amount the response deviate from true regression line → lack of fit of model.
- F-statistic $= f = \frac{\hat{\beta}_1 - b_0}{SE(\beta_1)}$ (0 if b_0 is $\beta_1 = 0$) → number of SD that $\hat{\beta}_1$ deviate from b_0).
  - → P value = prob to observe $|t|$ given H_0 is true
- accuracy of the model (once rejecting null) → $R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$ with $TSS = \sum(y_i - \bar{y})^2$
  - % variance explained by the model
  - inherent variance of data
- multiple linear regression → F-statistic $= \frac{(TSS-RSS)/p}{(RSS/(n-p-1))}$ → no relationship
  - look at F if many indicator and not t-stat of each
  - F close to 1
- In multiple regression we get t-stat for each individual, it is equivalent of the F-stat that omit that single variable keeping the others in.
- R^2 always increase
- get F-stat ↗ P value of F is big → keep H_0 no relationship
  - → P value of F is small → if P is small we can select individually but not best way
  - → P is big (many variable) = variable selection

→ forward selection: begin with a null model no predictor. Then fit p simple linear regression and add to the null model the variable result in lowest RSS. Repeat
→ Backward: start will all remove the one with largest P value. Repeat
→ Mixed: start nothing, add till a p-value of one ↗ then remove it. Repeat

- Select set of models with $\neq$ number of variable : don't use in high dimension → CV
- Small ($C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$ (used for fitted LS) AIC (maximum likelihood) $= \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$
  BIC (bayesian) $= \frac{1}{n}(RSS = \frac{1}{n_p}(RSS + 2d\hat{\sigma}^2)$. adjusted $R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$
- Shrinkage → $RSS + \lambda \sum_{j=1}^{p} B_j^2$ = Ridge → tend to zero = good when LS has variance (increase λ)
  → $RSS + \lambda \sum_{j=1}^{p} |B_j|$ = Lasso → sent to 0. allow to variable selection
- ROC curve : performance of a classification model at all classification threshold → maximize AOC.
- PCA → projecting p predictors on M - dimension ($M < p$) when M are linear combination of variables → then fit on M predictors
  → Centered around 0 and individually scaled $z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}$ $y_i = \theta_0 + \sum_m \theta_m z_{im} + \varepsilon$
  - PCR is done in unsupervised way, so no guarantee that the directions that best explain the predictors will also be the best directions for predicting the response → for that use least squares.
  → minimizing approximation error or maximizing variance.
- add interaction term, or polynomial regression to extend to non linear

---

Potential problems

- → non-linearity response - predictor => look residual plot => should be no pattern → else transform $\sqrt{x}, x^2, \log x$
- → correlation of error term => look plot
- → Non-constant variance of error term (heteroscedacity) => funnel shape residuals → transform y with $\log y$ or $\sqrt{y}$
- → outliers
- → high leverage points → point impact a lot
- → collinearity : Two or more predictor variables are related to one another. The proba of detecting a non zero coefficient ↓. Correlation matrix or variance inflation factor.

Beyond linearity

- → Polynomial regression (x, x^2, x^3)
- → Step function : range of variable into k distinct region $y = B_0 + B_1 C_1(x) + \dots + B_k C_k(x)$ $C_j(x) = I(c_j \leq x \leq c_{j+1})$
- → Regression spline : dividing the range x into k distinct, a polynomial function, join smoothly on knots → increase flexibility with more knots but keep the degree fixed.
- → Smoothing splines => add a roughness penality => regression splin with knots at each x_i => $\lambda \int g''(x)^2 dx$ => if $\lambda \to \infty$ = straight line
- → local regression = fit a target point x_0 using only the nearby point.
- → Generalized additive model : allow non linear functions of each the variables, while maintaining additivity $y_i = B_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i$

- logistic regression $\log\left(\frac{P(x)}{1-P(x)}\right) = B_0 + B_1 x_1 + \dots + B_p x_p$ and $P(x) = \frac{e^{B\dots}}{1+e^{B\dots}}$ estimated by likelihood
- logistic for classification / multinomial logistic regression → Softmax = $\frac{e^{B\dots}}{\sum e^{B\dots}}$ $P(y=k|X=x) = \frac{e^{B_k\dots}}{1+\sum e^{B\dots}}$
- Generative model classification → logistic we use $P_k(y=k|X=x)$ but is unstable if substantial separation between classes. instead π_k = prob to be in class k
  $f_k(x) = P_k(X \neq y = k)$ and then $P_k(y=k|X=x) = \frac{\pi_k f_k(x)}{\sum \pi_k f_k(x)}$. to get $f_k(x)$
  → linear discriminant : normal + same variance (better than logistic if normal) (more learning) (less learning)
  → quadratic discriminant ↳ gaussian, $N(\mu_k, \Sigma)$
  → naive bayes → p predictors independent $f_{jk}(x) = f_{k1}(x_1) \times \dots \times f_{kp}(x_p)$ reduce variance so good if p is large or n is small)
- kNN → completely non parametric → but doesn't tell which predictor is important
- Poisson regression $P_k(y=k) = e^{-\lambda} \frac{\lambda^k}{k!}$ and $\lambda = E(y) = e^{B\dots}$. one increase in x_j by one unit is associated with a change in $E(y) = \lambda$ by $\exp(B_j)$
- Generalized models → use $x_1, \dots x_p$ to predict y

$u =$	=> linear(gaussian) : $B_0 + B_1 \dots$	$m(u) = u$	$m(u) = m(E[y|x_1, \dots x_p])$
$E(y|x_1, \dots x_p)$	logistic : $P_k(y|x_1 \dots x_p) = \frac{e^{B\dots}}{1+e^{B\dots}}$	$m(u) = \log\left(\frac{u}{1-u}\right)$	$= B_0 + B_1 \dots$
	Poisson : $\lambda(x_1, \dots x_p) = e^{B\dots}$	$m(u) = \log(u)$	=> exponential family

→ perform a regression by modeling the response y as coming from a particular model and then transforming the mean so the mean is linear function of the predictors.

---

cross validation : leave-one-out : k-fold : Bootstrap : get distinct data set by sampling with replacement from original data.

Tree

- all predictors $x_1, \dots x_p$ and all possible values of the cut point s for each predictor
  → RSS : regression : mean of observations in R_j
  → Gini Index, Error rate, entropy : classification : most in occurring class
- Prune back → grow → get best results as function of α (cost complexity) → average results for each value of α → return best subtree from best α.
- Bootstrap aggregation : bagging : bootstrap generate data : grow tree deep but no pruning : they have variance → best average them to reduce variance.
  → average or majority → OOB MSE / classification error.
  → total amount RSS or Gini decreased due split after predictor → average it → important
- Random forest : decorrelate tree : random sample of $m = \sqrt{p}$ predictors.