Class 1: Multiple Regression, y^ as a linear combination of the Yi, hij and hii, leverage cut-off 2p/n, Partial F-test, VIF Formula

We are modeling the **EXPECTATION** → conditional mean of Y in a linear form of estimates

Marginal Slopes (confound direct & indirect effects) vs. **Partial Slopes** (MR→isolates/controls for direct effects)

Interaction X1X2: The impact of X1 on Y depends on X2; synergy not collinearity

$$VIF(X_1) = \frac{1}{1 - R^2_{X_1|X_2 \cdots X_p}}.$$

Collinearity: No correlation, VIF = 1**;** shouldn't be over 10.

$$\hat{y}_i = \sum_{j=1}^{n} h_{ij} y_j,$$

$$h_{ij} = \frac{1}{n} + \frac{(x_j - \bar{x})(x_j - \bar{x})}{\sum_{k=1}^{n}(x_k - \bar{x})^2}$$

$$\frac{d\hat{y}_i}{dy_i} = \frac{d}{dy_i}\sum_{j=1}^{n} h_{ij} y_j = h_i$$

Alternative interpretation of y^i:

hii = predicted y / actual y; shows how the model borrows strength from other observations to improve the prediction of a specific observation; ; quadratic nature

Leverage cut-off: if hii> 2p/n, then it's a large leverage value

Sum of hii = p (the number of parameters); **Average hii** = p/n

Weight (hij) depends **how far observation in x-direction away from mean** (further away put more weight) if hii small, won't impact much

Class 2: Adjusted R^2, Cp, AIC, BIC, Model Selection, Stepwise, Adjusted R^2 as a func of RMSE, KISS and parsimony, P-value cutoffs

$$\text{Adjusted}R^2 = \left(1 - \frac{RMSE^2}{s_y^2}\right),$$

Adjusted R^2 | always pick simplest one = KISS

Same as RMSE: one-to-one function of RMSE. Choose the same # parameters. Adjusted R^2 doesn't have to increase with additional variables (unlike regular R^2). It looks like a better choice but **maximizing Adjusted R^2 = to minimizing RMSE**

$$MSE = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2/n$$

MSE: calculated for out-of-sample data Mean of standard errors (leave one out, k-fold)s

$$AIC(k) \propto \frac{SSE_k}{\hat{\sigma}^2} + 2k.$$

$$: BIC(k) \propto \frac{SSE_k}{\hat{\sigma}^2} + \log(n)k.$$

One can show that a new variable is added to an existing model if:

Criterion	Approx $\lvert t\rvert$ cut-off	Equiv p-value	Goal
Adjusted R^2	$\lvert t\rvert > 1$	0.33	Minimize RMSE
C_p / AIC	$\lvert t\rvert > \sqrt{2}$	0.16	Achieve an unbiased estimate of prediction accuracy
BIC	$\lvert t\rvert > \sqrt{\log(n)}$	Depends on n	Something Bayesian!

AIC/BIC better than R^2: choose a model such that RMSE of the model is a legitimate estimate of how it performs when it sees new data → eliminate the overfitting problems | **always choose lowest AIC**

Mallows (C_p) = AIC when normally distributed; AIC will only be reliable if n>>k (if not, chooses are large number of predictors claiming better performance in terms of everything); Use AIC when trying to find a model, BIC when trying to identify the true model within sets. BIC penalizes complexity more than AIC (when log(n) > 2)

Stepwise model: iterative model selection. *if RMSE = 0.012573. The initial raw standard deviation of the returns was 0.0178048. So model/RMSE accounts for 0.012573/0.0178048*= 70.616% of the initial unexplained variation. → to judge a stepwise model; high is good

Categorical Interpretation E.g. if add Transmission (AV - M) to the model: For AV, forecast changes by ___ *estimate amt; if M, forecast changes by (negative) __ *estimate amt. All other levels change by 0. **Problem:** Sparsely populated levels may be completely full of 1's or zeroes, leading to estimated probabilities of 1 or 0. So the logits drift off to +/- infinity

Trade-off between Bias and Variance

The more parameters in the model, the better the approximation to the true underlying function (less bias). AIC and Cp are designed to trade bias and variance off against each other | ****Use AIC instead of comparing G^2 (diff in loglikelihood) for non-nested hypothesis testing**

Class 3: Bootstrap, crossvalidation, training & test errors, three types of cross-validation

Bootstrap: purpose is to obtain a measure of uncertainty → se(RMSE) measures the bootstrap standard error; Uses the sample as a proxy for the population and takes repeated samples from this pseudo-population | significance from zero (e.g. kurtosis find confidence interval, is 0 in it?)

Cross-validation: 1) How do I think my model will perform when I see new data (Precision of predictions) 2) Which model should I prefer? Select the one w/ lowest test error. MSE -> measure of error:

Cross-validation vs. AIC: Can use AIC to estimate test error, but does so formulaically, rather than dividing the dataset; You'd do this when you have very expensive data or data you don't wanna throw out. Cross-validation uses the available dataset to get an estimate of TEST error. Asymptotically → they become the same methodology as the sample size gets increasingly large

Drawbacks of validation: lost data, less precise, ambiguity results if we draw another sample

Three types of cross-validation: 1) Testing vs. Training Dataset. training error < test error | **PRESS (for linear models) =**

$$: \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{1 - h_{ii}}.$$

2) Leave one out (PRESS*: PRESS RMSE = Sqrt(PRESS/n).* Problems: Collinearity & Computational/speed → have to fit n different models Statistical → a lot of correlation between the leave-on-out forecasts leading to a HIGH VARIANCE of the test error estimate itself **3) K-Fold Cross Validation:** *Sum up ALL MSE and divide by K to get the overall MSE.* Good compromise between the other two approaches: not computationally expensive **(*SPEED*)** & uses more of the data for model fitting

Class 4 & 5: Classification, K-nearest neighbor, LDA, Logistics…. Outcome Y takes categorical, X continuous

K-nearest neighbor method: non-parametric (no assumptions); looks locally.

Linear Discriminant Analysis (LDA) Probability of a Y being yes = (height Yes)/(Height yes + height no). LDA allows for specifying prior probability using empirical data, assumes 1) normality but like X might be categorical 2) variances are the same in each level (could fix by doing quadratic DA) Method: uses Bayes theorem to convert X given Y to Y given X (all others directly find y given x)

Logistics: Modeling Logit of Prob = Log of Odds; **Coefficient:** Holding one variable (X1) constant, for every one unit increase in the other variable (X2), the ***logit of P(Y=1)*** increases by B2, or the odds of Y=1 increases by multiplicative factors exp(B2). B1 is called log odds ratio; **Maximum Likelihood Estimate = Least Squares for normally distributed data.** The -2*log difference of two models is approx. a chi-square distribution **(small – big model); Problem of Logistics:** if any proportion is 0,1 we'd get +/- infinity; Algorithm (Iteratively Reweighted LS)

$$logit\{p(x)\} = \beta_0 + \beta_1 X.$$

Compare Models; If models are not nested you cannot use the difference in log-likelihoods for hypothesis testing (use AIC).

$$logit(p(x)) = \log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 X. \qquad -2\{ll(M_0) - ll(M_1)\} \sim \chi^2_k,$$

$$p(x) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

Class 6 : Predictive Power/Fit (positive, negative, confusion matrix, etc)

Specificity = 1 - false positive rate → False positive/ total actual negative
Sensitivity = 1 - false negative rate → False negative / total actual positive
False positive rate (Type1) = % of total actual negative that was predicted as a positive
False negative rate(Type2) = % of total predicted negative that was actually positive
Gini or Entropy are common – measures of homogeneity (similar to SSE in regression);
Whole Model: overall F test (deviance/G^2 = difference in -loglikelihood) nested to compare
Purity measures include Entropy, the Gini Index and misclassification rates

For any set of observations (in-sample or out-of-sample) we can comp
the predicted values and the actual values in a table:

	Predicted negative	Predicted positive	Total
Actual negative	a	b	a + b
Actual positive	c	d	c + d
Total	a + c	b + d	a + b + c + d = n

- False positive rate = b/(a + b). False negative rate: c/(c + d).
- Overall misclassification rate = (b + c)/(a + b + c + d).
- This table is sometimes called the *confusion matrix*.

Misclassification rate = Sum of both falses/total number | **ROC Curve:** One curve that examines the predictive quality of the models by looking at all possible cut-offs; Plots **SENSITIVITY** vs. **1 – SPECIFICITY;** Equivalent to plotting ***True Positive vs. False Positive;*** A lousy classifier if **AUC** is ½ so a 45 degrees line because it's 50/50 chance; ideal if sensitivity & specificity = 1 → shoots up and right
Lift Curve: Compare the target to whole population → what proportion is actually positive

$$\text{Lift}(5\%) = \frac{\%\text{ in target list}}{\%\text{ in population}} = \frac{40\%}{10\%} = 4.$$

Interpretation of the curve: if I were to find the top X% in terms of my target, then in that top X%, I observe ___(lift value) **more actual no shows** than in the general population.

Class 7: Multiple Logit regressions

Categorical X Variables: JMP uses default [+1/-1] coding scheme → You have to **DOUBLE** the coefficient to get the exact difference. E.g.: Gender[Female] = 0.11864: on the logit scale, the difference between men and women is 2*0.11864; the ratio of the odds (***odds ratio***) for quitting between men and women is e^(2*0.11864) = e^(2*B1). ***Note**: Sparsely populated levels may be completely full of 1's or zeroes, leading to estimated probabilities of 1 or 0. So the logits drift off to +/- infinity…B1 → log odds ratio; e^B1 → odds ratio | you cannot logit 0 or 1
For **multi-level categorical** predictor, all their coefficient estimates add up to 0; **Range Odds Ratio** → Raise range (X) to the power of odds ratio

Class 8/9: Trees

Setting min. split size: big minim. split size will give you a very **PARSIMONIOUS** tree | **Split node:** Log worth, choose split with smallest p-value
Good: Easy to interpret; No prior structure/knowledge needed; Don't care about scale and outliers; Incorporate complex interactions; Correspond to how some decisions are made. **Bad**: Not efficient summaries; No neat equation to work with in the background to estimate an elasticity or a marginal cost; bad at prediction; Can be quite unstable → small changes in data can lead to big change in tree structure; Low bias, but high variance in terms of prediction; When there's a single important continuous predictor, then it requires A LOT OF splits to smooth the relationship. **TEST:** Trees can provide # of different predictions based on # of terminal nodes
Which variable to split on next? Continuous: variable that makes SSE go down the most |Discrete: variable that makes nodes "pure"

Class 10: Calibration, multiplicity, family error rate

Calibration goal: to ensure that the average of response at a predicted value is approximately equal to the predicted value (improve predictions)
Multiplicity: problem of testing multiple hypothesis; below 3 solutions address it by making it harder to declare an effect significant than p-value
Tukey: P(at least one error) = 1−(1−α)^k (problem with multiplicity since this probability is high); adjusts multiplicity by replacing 2 with Q = 2.409
Bonferroni: a* = a/k (k is the number of hypotheses. Bonferroni does not assume independence between the tests (which is helpful; but can be overly conservative - may have many false negatives

Example: I have ten hypotheses to be tested and I want the familywise error rate to be at most 0.05, then use $\frac{0.05}{10} = 0.005$ cut-off.

FDR: False Discovery Rate
False Discovery Rate = E(V/R), **expected value of (# of false positive) / (total # of declared positive) = FDR-adjusted p value (more stringent)**

Class 11: Ridge, Lasso (Multi-level categorical Y)

Ridge Regression: good for dealing w/ collinearity (in presence of extreme collinearity but still want to talk about regression slopes or variability in coefficients| **if lambda big = beta penalized/smaller** | outcome: slope of term will get closer to zero after ridge regression
Lasso: for **parameter estimation + simultaneous variable selection** | not only minimizes model also chooses model for you because can be 0 if the penalty term is changed to $\lambda \sum_{j=1}^{k} |\beta_j|$ Least absolute shrinkage + Selection operator (must tune lambda though)
Neural Networks for non-linear responses: More nodes&layers = more parameters = >> training data # on credit card example | out of sample confusion matrix shows multinomial logistic prediction is bad, random forest only some predicted and fundamentally wrong (not well tuned), neural networks w/ 2 hidden layers (high signal low noise complex) change nodes until **predictions of confusion matrix must be diagonal**

- Number of rows: gives the number of observations in the data table.
- Number of terms: gives the number of columns specified as predictors.
- Number of trees in the forest: is the number of trees to grow, and then average together.
- Number of terms sampled per split: is the number of columns to consider as splitting candidates at each split. For each split, a new random sample of columns is taken as the candidate set.
- Bootstrap sample rate: is the proportion of observations to sample (with replacement) for growing each tree. A new random sample is generated for each tree.
- Minimum Size Split: is the minimum number of observations needed on a candidate split.
- Maximum Splits Per Tree: is the maximum number of splits for each tree.
- Minimum Splits Per Tree: is the minimum number of splits for each tree.

$$min_\beta \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots, + \beta_k x_k))^2 + \lambda \sum_{j=1}^{k} \beta_j^2.$$

$$PRESS = \sum_{i=1}^{n} (y_i - \hat{y}_{i,-i})^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{1 - h_{ii}}.$$

tests:

	Declared not significant	Declared significant (discovery)	Total
Null true	U	V	k_0
Alternative true	T	S	$k - k_0$
Totals	k - R	R	k