Lecture 10: Bootstrapping & Predictive Modeling Workflow

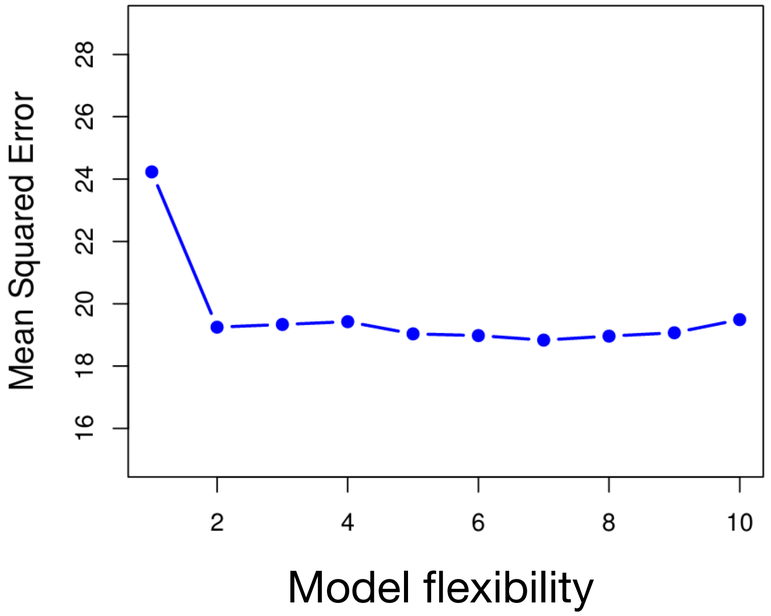
Wednesday, Oct

Contribute to the class playlist! 🎶

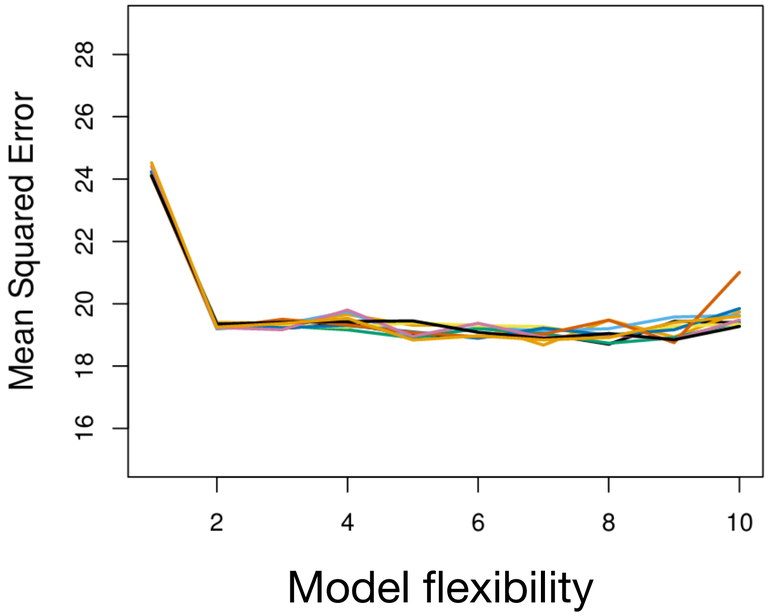
(preferably chill vibes…it is 9am after all 😴)

Comparing cross-validation methods

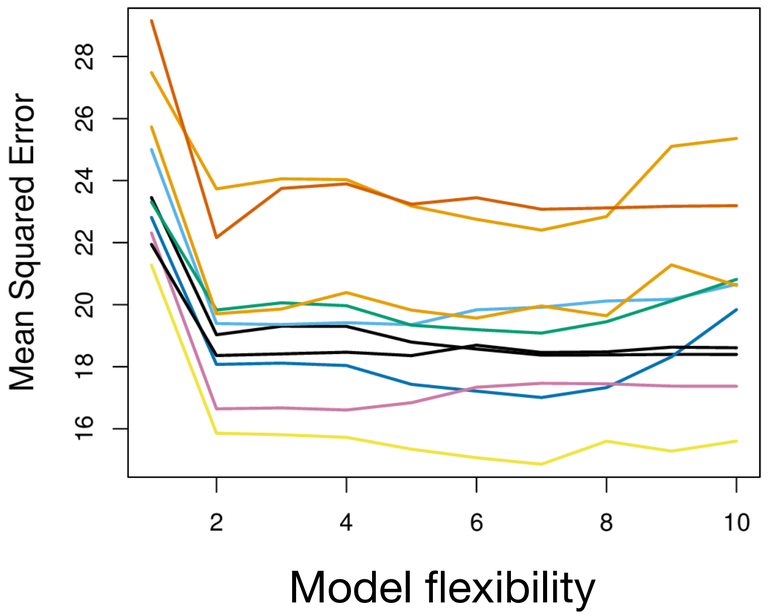
LOOCV



10-fold CV



Validation set approach



Bias and variance in test error estimates:

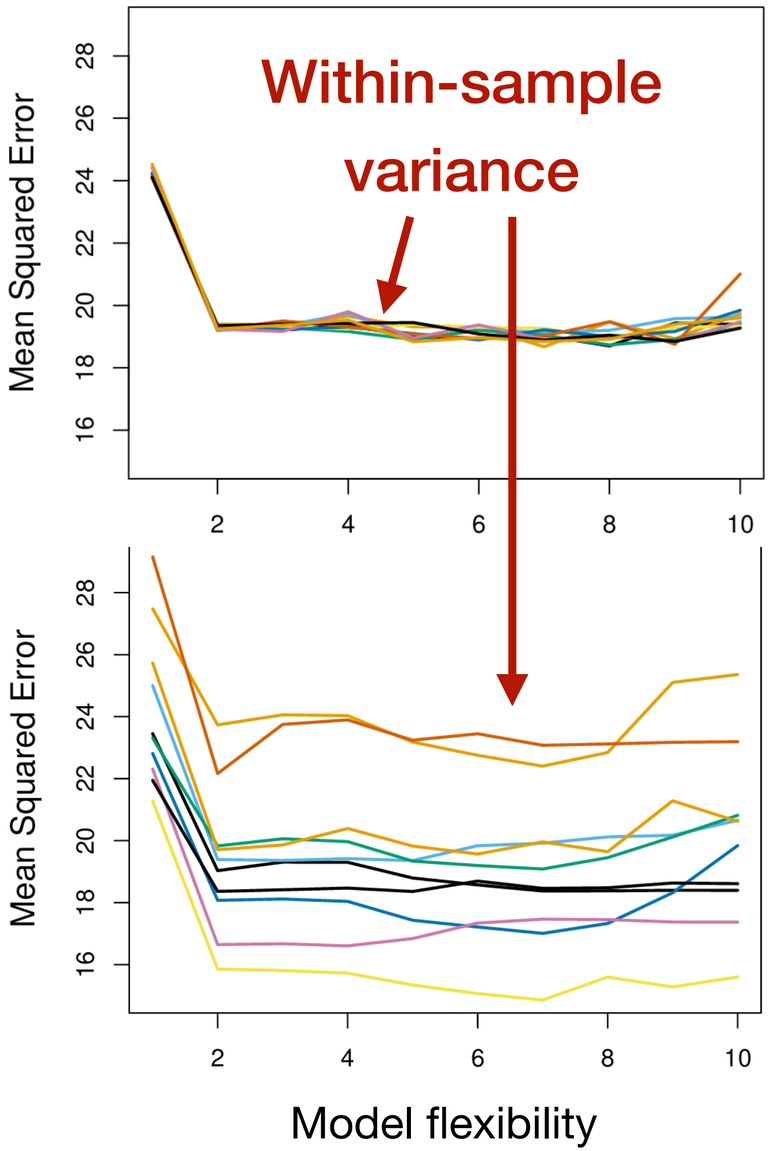
- In terms of *bias* (accuracy) in test error estimate: LOOCV < *k*-fold CV < validation set
- In terms of *variance* (noise) in test error estimate: *k*-fold CV < LOOCV < validation set

A quick note about variance

Two potential sources of variance:

- **Within-sample variance:** variability of CV error (estimated test error) **across folds** within the **same dataset**.
- **Across-sample variance:** variability of the estimated test error across **different random samples of the population.**
  - If you repeated CV with a **new dataset** drawn from the population, how much would the CV error fluctuate?

When I say a CV method has high variance, we are talking about the **across-sample variance!**



Summarizing bias and variance in CV error

Method	Training Set Size	Bias (related to training set size)	Within-sample variance	Across-sample variance
Validation set approach	*n/2*	🔺 **High:** Each model is trained on only half the data → parameter estimates are less accurate	🔺 **High:** CV error depends heavily on which random split was chosen → unstable across splits	🔺 **High:** CV error also depends heavily on which random sample from the population was chosen → unstable across samples
LOOCV	*n–1*	🟢 **Low:** Each model is trained on almost the full dataset → CV error is accurate estimate of test error	🟢 **None (deterministic):** No randomness in splitting data within a sample (always results in same CV error estimate)	🔺 **High:** Very sensitive to tiny changes across samples because models are highly correlated.
k-fold CV	*(k–1)/k·n*	🟡 **Moderate/Low:** Training set is still relatively large + averaging over multiple folds → CV error is still relatively accurate estimate	🟡 **Moderate/Low:** Averaging across folds reduces within-sample noise (similar CV error across folds)	🟡 **Moderate:** Less sensitive to tiny changes across samples because training set is < full dataset

Bootstrap

In Texas, we wear cowboy boots 🤠

“Pull yourself up by the bootstraps”

⇒ get yourself into or out of a situation using existing resources

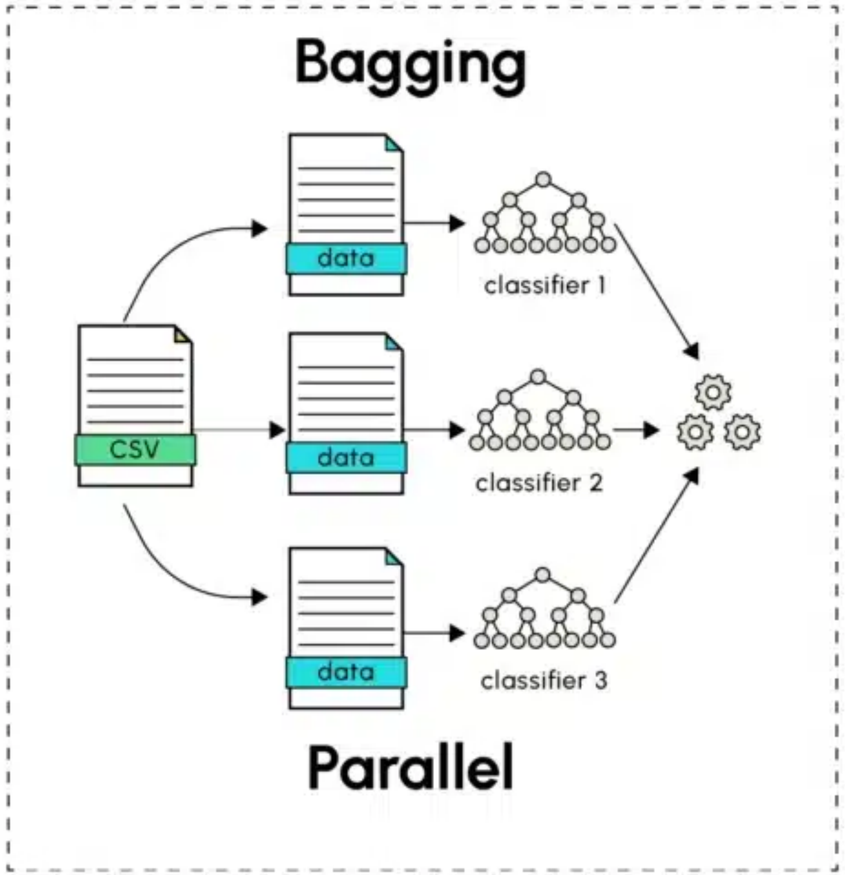
Bootstrapping in statistics / modeling

⇒ using the data you already have to make an estimate about the population data

Bootstrap demo

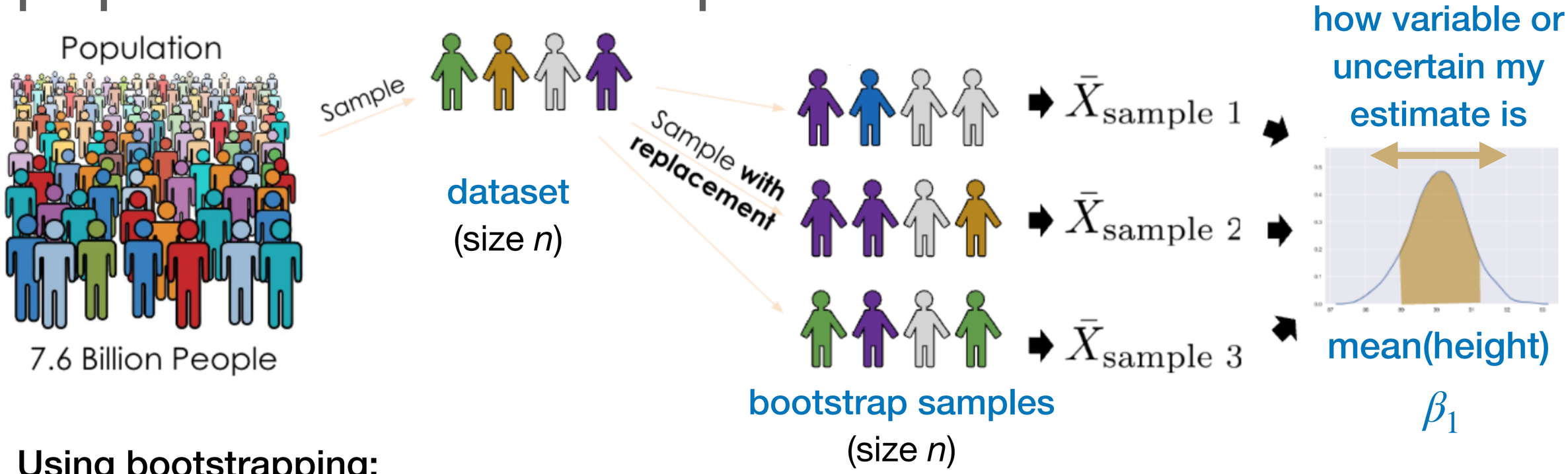
One volunteer please! 🤠

Applications of bootstrap in advanced methods



Bagging: creating random subsets of data to train independent models in parallel

Bootstrapping to estimate uncertainty in a population statistic or parameter



Using bootstrapping:

1. Draw bootstrap samples of size *n* **with replacement** from your dataset of size *n*
2. Using bootstrap samples, estimate the statistic or parameter of interest (e.g., mean, β_1) and compute the standard error and 95% CI (measure of uncertainty)

Revisiting inferential statistics

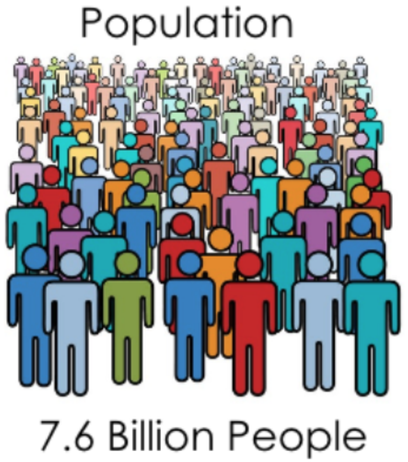
Descriptive statistics: Describe the sample (e.g., mean, median, std)

Inferential statistics: Use sample to make estimates about the *population*.

Standard error and confidence intervals, revisited:

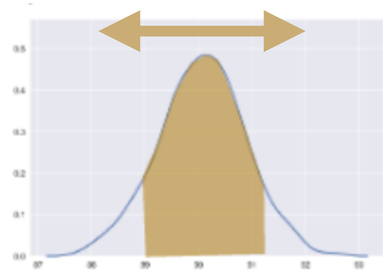
- The **standard error (SE)** of a parameter (e.g., β_1) quantifies **how much that estimate would vary** across repeated samples from the same **population.**
- A **confidence interval (CI)** uses the SE to express **a range of plausible values for true parameter** (e.g., 95% CI for $\beta_1 \in [\beta_1 \pm 1.96 \times \text{SE}(\beta_1)]$

Takeaway: Bootstrapping can be used to estimate this uncertainty for <u>any</u> population statistic or parameter estimate!



sample (dataset)

how variable or uncertain my estimate is



mean(height)

Why would we use bootstrap if we already have CIs?

OLS Regression Results — 95% CI for coefficients

	coef	std err	t	P>|t|	[0.025	0.975]
const	539.5872	750.011	0.719	0.472	-933.729	2012.903
x1	1.2932	0.104	12.379	0.000	1.088	1.498
x2	-120.6634	29.708	-4.062	0.000	-179.021	-62.305
x3	0.2246	0.025	9.030	0.000	0.176	0.273
x4	32.4225	6.807	4.763	0.000	19.051	45.794
x5	76.3294	9.716	7.856	0.000	57.243	95.416

- The SE (and therefore 95% CIs) are calculated using several **assumptions** (e.g., errors must be normally distributed).
- For **complex, non-linear models,** or data that are **non-normal**, it may be hard to estimate the SE
- **Bootstrapping** ⇒ a general way of estimating the SE that makes very few assumptions and that works for any model!

Bootstrapping to estimate test error

Using bootstrap to obtain test error estimates:

1. Draw **bootstrap samples** of size *n* **with replacement** from your dataset of size *n*
2. Train your model on a **bootstrap samples**
3. Then test it on the data points that were **NOT** included in that sample (the **"out-of-bag (OOB)"** observations)
4. Compute the **OOB error** (estimate of test error)
5. Repeat #1-4 a couple hundred times (e.g., 100-300) and average across OOB errors

Dataset:

Sample:

Out of bag (OOB) sample:

Bootstrapping to estimate test error

Only use bootstrap > CV to obtain test error estimates for small sample sizes!

When n is small, k-fold cross-validation leaves **too few points in each validation fold** → high variance in the CV error (unstable)

- **Example:** If $n = 30$ observations…
  - 10-fold CV → folds of size 3
  - 5-fold CV → folds of size 6
  - In this case, use bootstrap resampling! (Gives more stable, though slightly biased, estimates).

Dataset:

Sample:

Out of bag (OOB) sample:

Takeaway: CV and bootstrapping

- Use **cross-validation** when your goal is to get a good estimate of how well your model does on unseen data (estimated test error, generalization error).
- Use **bootstrap** when your goal is uncertainty for a statistic (computing SEs and 95% CIs) or for estimating test error for small *n.*

Predictive modeling workflow (so far)

1. **Choose** a dataset and **define** the goal / question / problem
   - Using the **ISLP Brain Cancer dataset**, we want to predict **whether a patient is still alive or not** (classification problem) from various brain tumor characteristics.
   - Our goal is to accurately predict and to determine the factors that are most predictive of whether a patient is alive.

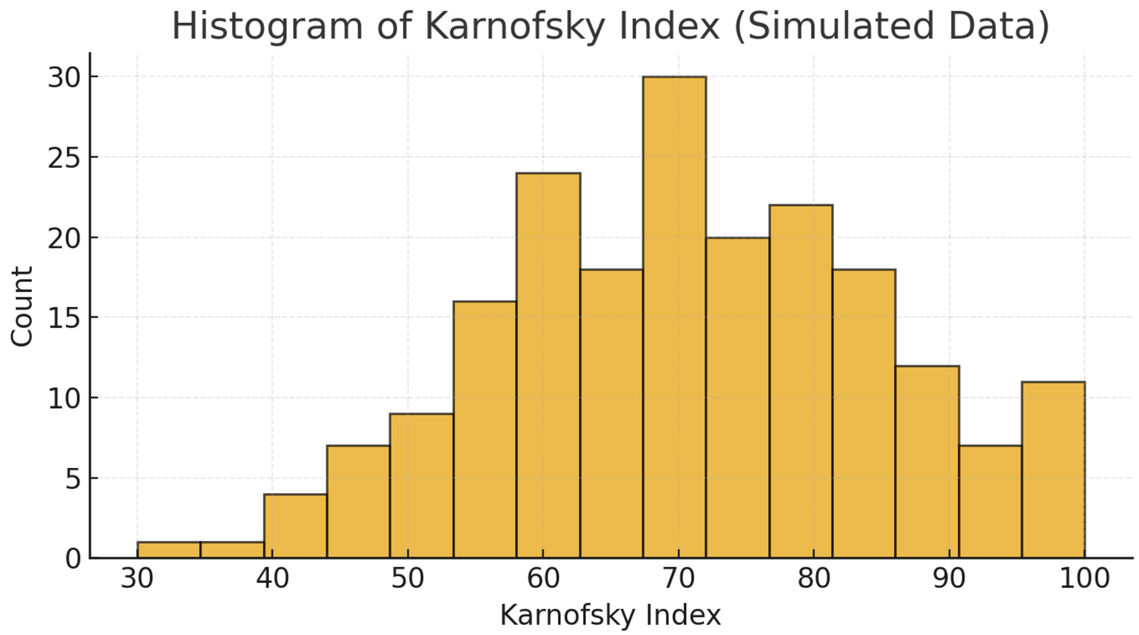
2. **Write down the model:** Multiple logistic regression with binary outcome

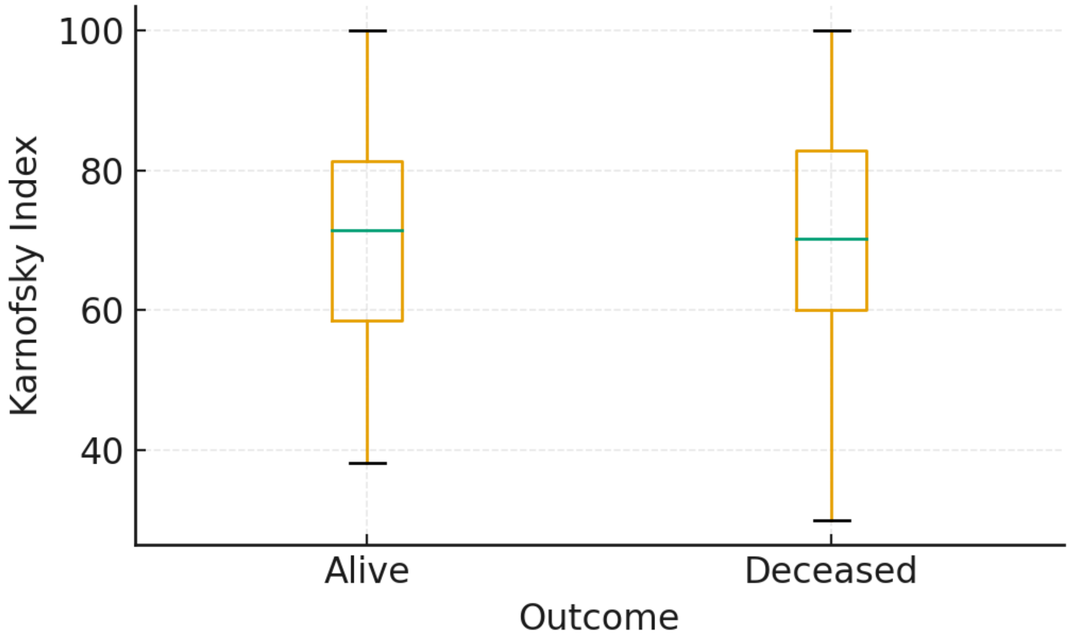
$$\text{Let } p = P(\text{status} = 1 \mid \text{sex}, \text{diagnosis}, \text{loc}, \text{ki}, \text{gtv})$$

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \text{ki} + \beta_2 \text{gtv} + \beta_3 \text{Male[Yes]} + \beta_4 \text{LGglioma[Yes]} + \ldots$$
$$\beta_5 \text{Meningioma[Yes]} + \beta_6 \text{OtherDiag[Yes]} + \beta_7 \text{Supratentorial[Yes]}$$

Predictive modeling workflow (so far)

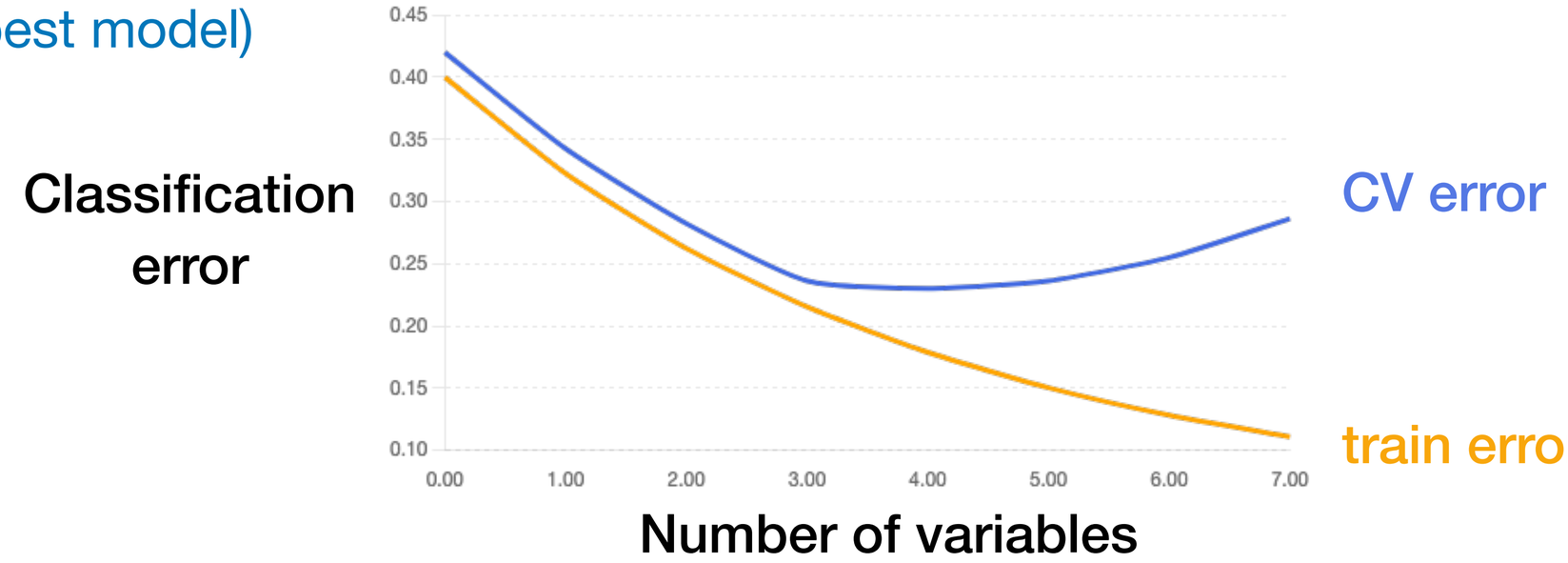
3. **Exploratory data analysis (EDA)**
   - Inspect missing values and outliers, check assumptions
   - **Plot:** histograms of predictors, scatterplot of outcome vs. predictors (regression), correlation matrices or tables (regression), box plots of predictors (classification)





Predictive modeling workflow (so far)

4. **Split the data** into train / test / validation sets, **put the test set away** until the very end!
   - Test: 20% of data, Train: 80% of data → further split into training / validation set
5. **Train the model** on training set, use validation set to **refine** / **find the best model variant**
   - Diagnose overfitting / underfitting using train and CV error
   - Try out models with different # of variables or parameters
     - **Plot:** Train and CV error vs. number of parameters / variables, examine where the CV error is lowest (best model)



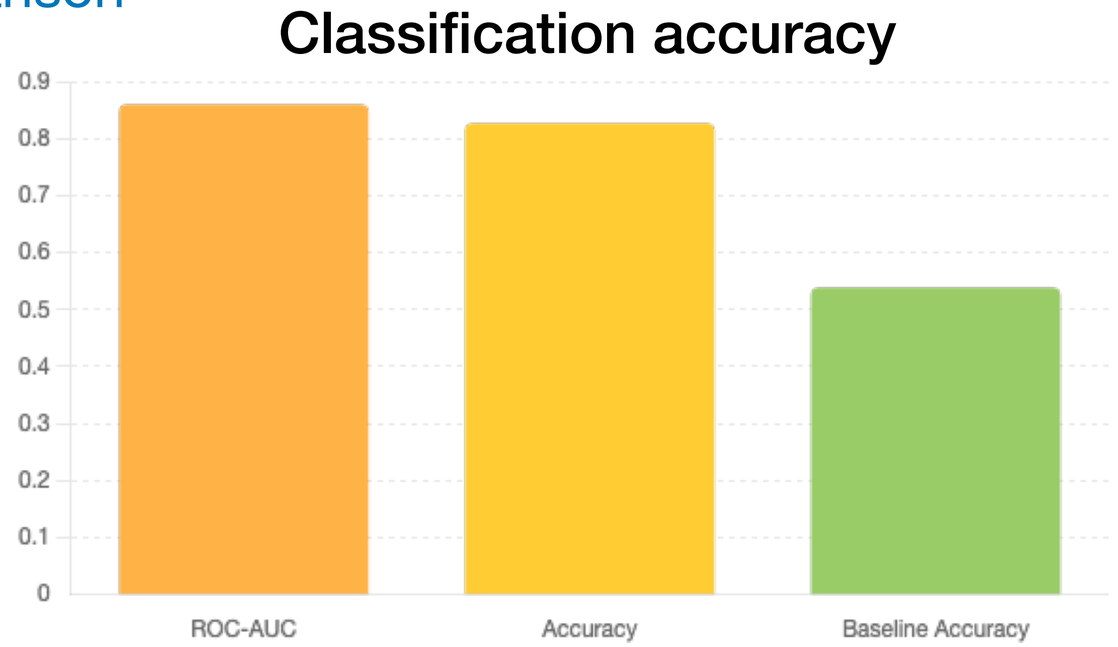
Predictive modeling workflow (so far)

6. **Refit the best model on the entire training set** and **evaluate final performance on the held-out test set** → unbiased estimate of generalization error

   - **Plot:** Scatter plot of $\hat{y}$ (predicted) vs. y (actual) for regression
   - **Plot:** Confusion matrix table (TP, TN, FP, FN counts) for classification
   - **Report final model performance metrics:** R^2, Adjusted R^2, RMSE, ROC-AUC, test error / accuracy rate, naive baseline comparison

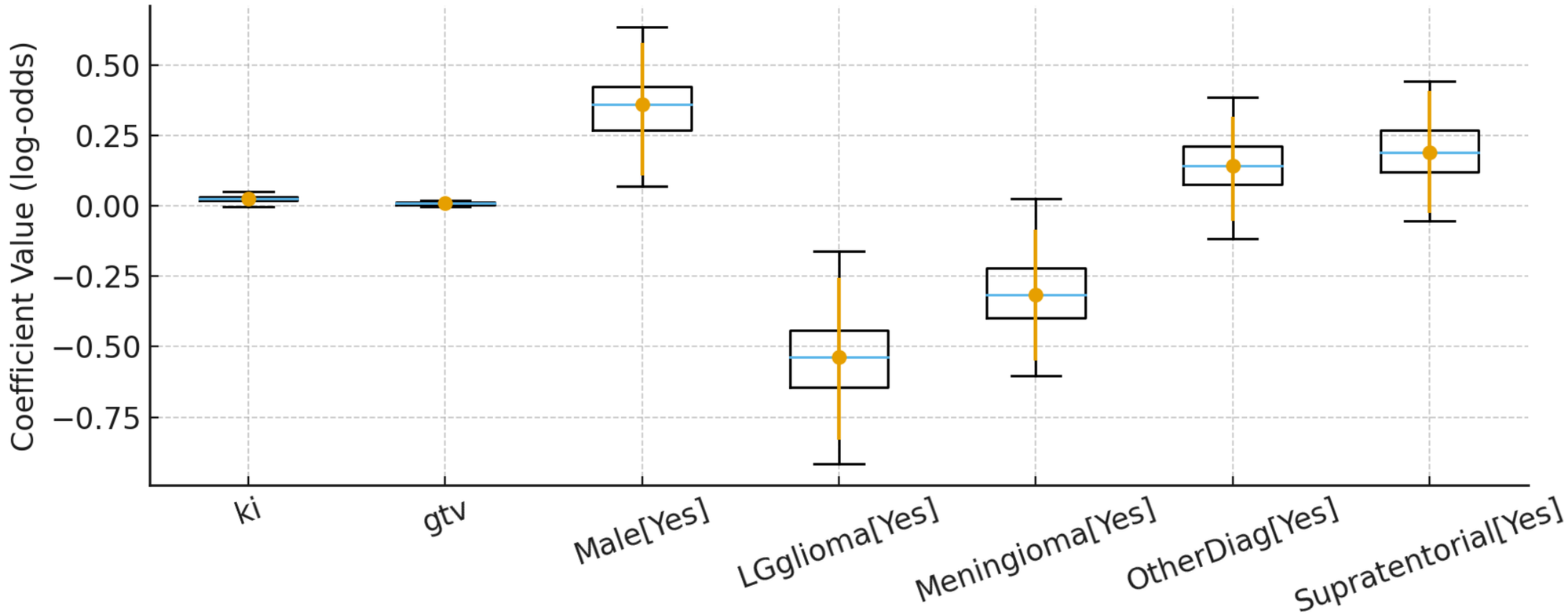
Confusion matrix

Predicted Y	Actual Y: Y = dead	Actual Y: Y = alive
Y = dead	61	12
Y = alive	27	76



Predictive modeling workflow (so far)

7. (If your model is complex) **Use bootstrap resampling** on the final model to obtain **95% confidence interval estimates for the fitted parameters**
   - **Plot:** Model parameters as box plots with 95% CI error bars



Predictive modeling workflow (so far)

8. **Interpret model** and **communicate findings**
   - **Interpret parameters:** describe key predictors and their direction and magnitude of effect, interpret odds / odds ratios (classification)
   - **Plot:** Make visual plots of main takeaways. Use AI as your advisor!
     - *“I found that sex is a significant predictor of whether a patient is alive or not at the end of the study. What kind of plot best shows this point?”*

9. **Discuss limitations** and possible **next steps** to improve the model.
   - We want to add in non-linear and interaction terms on the predictors. We also want to try out another model type (other than logistic regression).

Upcoming + Reminders

Updates:

- **HW 3 feedback** should be out by tonight!

Assignments:

- HW 4 (DUE: Friday @ 11:59pm)
- Quiz 4 (optional—DUE: Monday @ 11:59pm)

Friday's topic: *Generative Discrimination Models (cont'd) & kNN*

- **Read:** ISLP Ch. 4.4 & 2.2