**Theme 1: Model Performance & Assessment (Lecture 1)**

* **Core Concept**: All models have errors. We can break down the **test error** into **bias** (systematic error from a model being too simple) and **variance** (error from a model being too sensitive to the training data). The goal is to find the **optimal complexity** that balances this **trade-off**.
* **Key Methods**:
  + **Cross-Validation (CV)**: A method to estimate test error by splitting data into training and testing sets.
    - **K-Fold CV (e.g., 10-fold)**: Splits data into K folds, trains on K-1, tests on the held-out fold, and repeats. It has slightly more bias but lower variance than LOOCV.
    - **Leave-One-Out CV (LOOCV)**: A version of K-fold where K equals the number of observations. It's nearly unbiased but has high variance because the training sets are almost identical.
  + **Bootstrap**: A resampling method where you create new datasets by sampling *with replacement* from the original data. It's primarily used to estimate the **uncertainty** (standard error, confidence intervals) of model parameters.

**Relevant Questions:**

* *Discuss the Bias-Variance trade-off. How does it differ between LOOCV and 10-fold?*
* *What effects do you see when you cross-validate? Discuss in terms of bias-variance tradeoff.*
* *What is bootstrapping, how does it work, and when would you use it instead of CV?*

**Theme 2: High-Dimensionality & Regularization (Lectures 2 & 3)**

* **Core Concept**: When the number of predictors (p) is large, especially p > n, models suffer from the **curse of dimensionality**, high variance, and collinearity. Standard methods fail.
* **Key Methods**:
  + **Subset Selection**: Choosing a smaller set of predictors (e.g., forward or backward stepwise). Can be computationally intensive and miss the best model.
  + **Ridge Regression**: A **shrinkage** method that adds a penalty (λ∑βj2​) to the loss function. It shrinks coefficients toward zero but **never to exactly zero**. Great for highly correlated variables. Requires **standardization** of predictors.
  + **Lasso Regression**: Another shrinkage method with a different penalty (λ∑∣βj​∣). This penalty **can shrink coefficients to exactly zero**, performing automatic variable selection. Good when you believe many predictors are irrelevant. Also requires **standardization**.
  + **Principal Component Regression (PCR)**: A **dimension reduction** technique. It creates a smaller number of new, uncorrelated predictors called **principal components** that capture the most variance in the original data. Then, it runs a regression on these components.

**Relevant Questions:**

* *Discuss Lasso, Ridge, and PCR. When to use each, and what are the differences?*
* *What are three considerations when analyzing a dataset with many predictors?*

**Theme 3: Non-Linear Models (Lecture 4)**

* **Core Concept**: When the relationship between a predictor and the response isn't a straight line, we need more flexible models.
* **Key Methods**:
  + **Polynomials**: A classic approach, but high-degree polynomials can behave erratically, especially in the tails (high variance).
  + **Splines**: A better approach that fits **piecewise** low-degree polynomials in different regions of the data, connected smoothly at points called **knots**.
    - **Cubic Spline**: Uses piecewise cubic functions. Flexibility is controlled by the **number of knots**.
    - **Natural Spline**: A cubic spline with an extra constraint: it must be linear in the tails, which reduces variance.
    - **Smoothing Spline**: A very flexible natural spline with knots at every data point, where flexibility is controlled by a **smoothing parameter (λ)** that penalizes "roughness".
  + **Generalized Additive Models (GAMs)**: Extends GLMs by allowing non-linear functions for each predictor, while keeping the model **additive**. The formulation is g(E(y))=β0​+f1​(x1​)+⋯+fp​(xp​). They are more flexible than GLMs but don't inherently model interactions.

**Relevant Questions:**

* *Define smoothing splines and cubic splines. How do you control their degrees of freedom?*
* *When would you use a GAM instead of a GLM? What are the formulations?*
* *Why do we need to manually add interaction terms to a GAM?*

**Theme 4: Tree-Based Ensemble Methods (Lecture 5)**

* **Core Concept**: Decision trees are simple, interpretable models that naturally handle interactions, but they are often "weak learners" with high variance. **Ensemble methods** combine many weak learners to create one strong learner.
* **Key Methods**:
  + **Bagging (Bootstrap Aggregating)**:
    1. Create many bootstrap samples (sampling with replacement) from the data.
    2. Train a full decision tree on each sample.
    3. Average the predictions of all trees.
    4. **Effect**: Reduces variance.
  + **Random Forests**: An improvement on bagging. During tree construction, at each split, it only considers a **random subset of predictors**.
    1. **Effect**: This **decorrelates the trees**, leading to a greater reduction in variance than bagging. This is why it often outperforms bagging.
  + **Boosting**: A **sequential** method. It trains a tree, then trains the next tree on the **residuals** (errors) of the first one, and so on. It learns "slowly" by focusing on mistakes. It can overfit if too many trees are used.

**Relevant Questions:**

* *What is bagging and its algorithm?*
* *What is boosting?*
* *Why does Random Forest often outperform bagging?*