SET A

1. Definition of Statistical Learning, supervised vs unsupervised, and the accuracy–interpretability trade-off

Statistical learning refers to a collection of methods used to understand and model the relationship between a set of predictors X and an outcome Y, either for the purpose of prediction or inference. In supervised learning, both X and Y are observed, and the goal is to estimate the functional relationship $Y = f(X) + \varepsilon$ for classification or regression, while in unsupervised learning there is no outcome variable and the objective is to uncover structure among the predictors themselves, such as clusters or latent components. A central concept in supervised learning is the trade-off between prediction accuracy and model interpretability: more flexible models (like trees, splines, and neural networks) typically achieve lower bias and potentially better predictive accuracy but become harder to interpret, whereas simpler models (like linear regression) offer interpretability at the cost of higher bias and restricted predictive power. At the same time, greater flexibility usually increases variance and the risk of overfitting; therefore, one must balance bias and variance to achieve optimal generalization on unseen data.

2. Assessing model accuracy – the bias–variance trade-off

The accuracy of a statistical learning method is usually evaluated using its expected test mean squared error (MSE), which decomposes into a reducible part (bias^2 plus variance) and an irreducible part due to the inherent noise ε. As model flexibility increases, bias decreases because the model can better capture the underlying signal, but variance increases because the model becomes more sensitive to fluctuations in the training data. Models that are too rigid underfit the data (high bias), while overly flexible models overfit (high variance), fitting noise rather than structure. The optimal model minimizes the test error at an intermediate level of flexibility. Understanding this trade-off is fundamental when selecting between models like linear regression (low variance, high bias) and more flexible approaches such as splines, trees, or complex machine learning algorithms, which achieve lower bias but must be controlled via pruning, regularization, or cross-validation to avoid excessive variance.

3. Analysis of residuals of a linear model

Residual analysis evaluates whether the assumptions underlying linear regression are satisfied. By examining plots of residuals versus fitted values, versus individual predictors, and versus time or order, one can detect heteroscedasticity (changing variance), autocorrelation (patterned residuals over time), and model misspecification such as omitted nonlinearities or important variables. Residual histograms and QQ-plots are used to assess normality, which matters for the validity of t-tests and F-tests. Standardized and studentized residuals help identify outliers, while leverage values and Cook's distance highlight observations with high influence on the fitted model. Proper residual analysis ensures that conclusions drawn from a linear model—especially regarding inference and prediction—are valid and not distorted by violations of the core assumptions.

4. Multicollinearity and its analysis

Multicollinearity occurs when predictors are highly correlated, making the estimation of regression coefficients unstable and inflating their standard errors. While perfect collinearity prevents model fitting entirely, quasi-multicollinearity leads to coefficients that can shift dramatically in magnitude or even change sign with small changes in the data. The degree of multicollinearity is commonly assessed using the Variance Inflation Factor (VIF), where values above roughly 2.5 or 10 indicate problematic inflation, and graphical tools or correlation matrices can also reveal clusters of correlated predictors. Although multicollinearity does not necessarily harm prediction accuracy, it makes interpretation of individual coefficients unreliable. Remedies include variable selection methods, ridge regression, or replacing correlated variables with principal components to stabilize estimates.

5. Effect size measures for a linear model

Effect size measures quantify the magnitude and importance of predictors beyond mere statistical significance. Standardized coefficients allow comparison across predictors measured on different scales, whereas partial and semi-partial R^2 measure the unique portion of variance in the outcome explained by a specific predictor after accounting for others. Global effect measures such as R^2 and adjusted R^2 describe the overall explanatory power of the model, with the latter adjusting for model complexity and helping to guard against overfitting when adding predictors. Cohen's f^2 provides a ratio-based measure of effect size, capturing how much a predictor or set of predictors improves model performance relative to the variance that remains unexplained. These metrics ensure that interpretations are based not only on statistical significance but also on substantive importance.

6. The validation set approach

The validation set approach involves splitting the dataset into a training set used to fit a model and a validation (or hold-out) set used to estimate its test error. This method is simple and intuitive: a model is trained on one part of the data and its performance is evaluated on the other part using an error measure such as MSE or misclassification rate. However, it suffers from high variance because the outcome depends heavily on which observations fall into each set, and it tends to be biased upward because the training sample is smaller than the full dataset, leading to worse performance than a model trained on all available data. Nevertheless, the validation set approach provides a straightforward way to compare models or tuning parameters when computational cost is low and a rough estimate of generalization performance is sufficient.

7. Leave-One-Out Cross-Validation (LOOCV)

Leave-One-Out Cross-Validation (LOOCV) fits the model n times, each time leaving out one observation and evaluating the prediction error on that single left-out point; the average of these errors yields an almost unbiased estimate of test error. Because each training set uses $n-1$ observations, LOOCV closely mimics training on the full dataset, which explains its low bias. However, its variance is higher than that of k-fold cross-validation because the training sets differ by only one observation, so the fitted models are highly correlated. In general, LOOCV can be computationally expensive, but for linear models there exists an efficient shortcut formula based on leverage values that allows computing the LOOCV error from a single model fit.

8. k-fold cross-validation

In k-fold cross-validation, the data are split into k approximately equal-sized folds. Each fold is used once as a validation set while the remaining $k-1$ folds are combined as the training set, and the resulting test errors from each fold are averaged to produce an overall error estimate. This method offers an excellent compromise between computational efficiency, bias, and variance: it has lower variance than LOOCV and lower bias than the simple validation set approach. In practice, common choices are $k = 5$ or 10, which tend to provide stable and reliable estimates of test error and are widely used for choosing among models and tuning parameters in applied work.

9. The Bootstrap

The bootstrap is a general resampling method used to estimate the variability of a statistic by repeatedly sampling with replacement from the observed data and computing the statistic across these bootstrap samples. The empirical distribution of the bootstrap estimates approximates the true sampling distribution of the estimator, enabling estimation of standard errors, confidence intervals, and bias without requiring strong parametric assumptions about the underlying data-generating process. Bootstrap methods are extremely flexible and particularly valuable when analytic standard errors are difficult or impossible to derive, as in complex estimators, nonstandard models, or highly nonlinear functionals of the data.

SET B

10. Best Subset Selection

Best subset selection evaluates all 2^p possible models involving any combination of the p predictors. For each model size k, it chooses the model with the lowest training RSS or highest R^2, and then uses cross-validation or information criteria such as AIC, BIC, or adjusted R^2 to select the best overall model. Although exhaustive and conceptually appealing because it searches the entire model space, best subset selection becomes computationally infeasible for large p, and the selected model may still suffer from high variance due to overfitting, especially when many predictors are available. However, it is effective at removing redundant predictors and can be valuable in moderate-dimensional settings where interpretability and careful variable selection are important.

11. Forward Stepwise Selection

Forward stepwise selection begins with no predictors and adds them one at a time, choosing at each step the variable that leads to the greatest reduction in RSS or improvement in a chosen criterion. It drastically reduces computational burden compared to best subset selection because it does not revisit earlier choices, and in many situations it yields models with similar predictive performance. However, because the search path is greedy and irreversible, it may miss the globally optimal model that best subset selection could find. Forward selection is more stable and scalable when p is large and is especially useful when the analyst wants a relatively small, interpretable model and must balance model quality against computational feasibility.

12. Regression trees

Regression trees partition the predictor space into non-overlapping rectangular regions using recursive binary splitting: at each node, the algorithm selects the predictor and cutpoint that minimize the within-node RSS among all possible splits. Predictions for new observations are obtained by sending them down the tree according to the splitting rules and taking the average response in the terminal node. Trees are highly interpretable because they correspond to simple, hierarchical if–then rules and automatically capture nonlinearities and interactions among predictors. However, they tend to overfit the training data and exhibit high variance, so cost-complexity pruning combined with cross-validation is used to select an optimally sized subtree with better generalization performance.

13. Classification trees

Classification trees use the same recursive binary splitting mechanism as regression trees but evaluate candidate splits based on node impurity measures such as the Gini index or cross-entropy rather than RSS. Each terminal node is assigned a predicted class equal to the most frequent class (mode) among the training observations in that node, and associated class probabilities can also be estimated. Classification trees are easy to interpret and can capture complex nonlinear decision boundaries and interactions between predictors, but, like regression trees, they are prone to high variance and overfitting. Pruning, cross-validation, and ensemble methods such as bagging, random forests, and boosting are therefore crucial for achieving high predictive accuracy.

14. Aggregation methods: Bagging, Random Forests, Boosting

Aggregation methods improve tree-based model accuracy primarily by reducing variance and, in some cases, bias. Bagging (bootstrap aggregating) generates many bootstrap samples, fits a separate tree to each, and averages their predictions (or takes a majority vote), thereby stabilizing the inherently unstable single-tree estimator. Random forests extend bagging by randomly selecting a subset of predictors at each split, which reduces correlation among trees and leads to a stronger variance reduction and higher accuracy. Boosting takes a different approach: it fits trees sequentially, with each new tree focusing on the residuals or errors of

the previous ensemble, and combines them using shrinkage; this can substantially reduce bias and often yields state-of-the-art performance but requires careful tuning of the learning rate, number of trees, and tree depth. Together, these ensemble methods overcome the instability and limited predictive power of individual trees.

15. Dissimilarity and similarity measures

Clustering begins by defining how to measure similarity or dissimilarity between observations, because the choice of metric determines which structures are detectable. Common dissimilarity measures include Euclidean and Manhattan distances, which capture geometric closeness; Mahalanobis distance, which accounts for the covariance structure of the variables; more general Minkowski distances; and Canberra distance, which is useful in the presence of outliers or highly variable scales. Similarity measures, such as cosine similarity or correlation-based measures, quantify how alike two observations are in terms of direction or profile rather than magnitude. Selecting an appropriate measure is essential because it influences the shape, separation, and interpretation of clusters and must be chosen with regard to scale, units, and correlation among variables.

16. Hierarchical clustering: single & complete linkage

Hierarchical agglomerative clustering starts by treating each observation as its own cluster and then repeatedly merges the two closest clusters according to a chosen linkage criterion, producing a tree-like representation called a dendrogram. Single linkage defines the distance between two clusters as the minimum distance between any pair of points in them, which can detect long, chain-shaped or non-convex clusters but suffers from the chaining effect, where clusters grow by adding individual points or small groups connected by single links. Complete linkage defines the inter-cluster distance as the maximum distance between any pair of points, leading to compact, roughly spherical clusters and reducing chaining but making it harder to detect elongated structures. Unlike k-means, hierarchical methods do not require specifying the number of clusters in advance and provide a full multiscale view of the data structure, but they are more computationally intensive and do not allow reassigning points once merged.

17. Hierarchical clustering: average linkage & Ward's method

Average linkage merges clusters based on the average pairwise distance between all points in the two clusters, providing a compromise between the extremes of single and complete linkage and often resulting in clusters that are neither too elongated nor overly compact. Ward's method instead chooses the pair of clusters to merge that leads to the smallest increase in the total within-cluster sum of squares (or equivalently, the smallest loss of homogeneity), producing clusters that are typically very compact and spherical, similar in spirit to k-means. Compared to non-hierarchical methods, these hierarchical procedures generate an entire clustering hierarchy and do not require pre-specifying the number of clusters, but they are computationally heavier, especially for large datasets, and cannot reassign points once a merge has been made.

18. Non-hierarchical clustering: k-means & comparisons

k-means clustering seeks to partition observations into K clusters by minimizing the within-cluster sum of squares. It starts from initial centroids, alternates between assigning each observation to the nearest centroid and updating each centroid as the mean of the points assigned to it, and iterates until convergence. This algorithm is computationally efficient and scales well to very large datasets, but it assumes roughly spherical clusters with similar variance, is sensitive to initialization and outliers, and requires the choice of K in advance. Unlike hierarchical clustering, k-means allows observations to move between clusters during the algorithm, but it does not provide a dendrogram or multilevel cluster structure, making it less informative about relationships among the clusters themselves.

SET C

19. The Ridge Regression estimation method

Ridge regression addresses instability in ordinary least squares estimates by adding an ℓ^2 penalty to the loss function, shrinking coefficients toward zero without setting them exactly to zero. This reduces variance and can improve prediction accuracy, particularly in the presence of multicollinearity or when the number of predictors p is large relative to the sample size n. The ridge solution $(X'X + \lambda I)^{-1}X'Y$ stabilizes estimates by inflating the diagonal of the information matrix, effectively damping the impact of directions in the predictor space that are poorly determined by the data. The amount of shrinkage depends on the tuning parameter λ, which is typically chosen via cross-validation to balance bias and variance.

20. The Lasso estimation method

Lasso regression minimizes the residual sum of squares with an ℓ^1 penalty on the coefficients, producing sparse solutions by shrinking some coefficients exactly to zero. This yields both regularization and variable selection in a single procedure, allowing the model to automatically exclude irrelevant or redundant predictors and improving interpretability. Geometrically, the ℓ^1 constraint region has corners aligned with the coordinate axes, encouraging solutions that lie on these axes and thus contain zeros. The tuning parameter λ controls the degree of sparsity: larger values yield fewer nonzero coefficients. As with ridge, λ is usually selected via cross-validation to achieve good predictive performance without overfitting.

21. Geometrical interpretation of Ridge vs Lasso (two predictors)

Geometrically, ridge regression constrains the coefficient vector to lie within a circular (or spherical) region, while the lasso constrains it within a diamond-shaped region (an ℓ^1 ball) with sharp corners on the coordinate axes. The ordinary least squares solution corresponds to the center of elliptical contours of constant residual sum of squares, and the penalized estimate is found where these ellipses first touch the constraint region. For ridge, the smooth circular boundary means that the solution typically lies in the interior of a quadrant, shrinking all coefficients toward zero but rarely making any exactly zero. In contrast, the corners of the lasso constraint region cause the ellipses to often touch at a vertex, forcing one or more coefficients to be precisely zero. This geometric difference explains why ridge performs continuous shrinkage without variable selection, whereas lasso performs both shrinkage and true variable selection.

22. Solving multicollinearity with subset selection, stepwise selection, and ridge regression

Best subset and forward stepwise selection help with multicollinearity by excluding redundant predictors from the model, thereby simplifying the design matrix and potentially removing some unstable coefficients. However, these selection methods do not directly stabilize the coefficients of correlated predictors that remain, and the chosen subset may itself be unstable across samples. Ridge regression is specifically designed to address multicollinearity by imposing an ℓ^2 penalty that shrinks correlated coefficients toward each other and reduces their variance without necessarily dropping any predictors. For strong collinearity, ridge typically provides the most robust and stable solution, while subset and stepwise selection are more appropriate when interpretability and sparsity are priorities, albeit with greater sensitivity to sampling variation.

23. High-dimensional data: definition and cautions

High-dimensional data refers to settings where the number of predictors p is large relative to, or even greater than, the number of observations n. In such contexts, ordinary least squares fails because $X'X$ is singular when $p > n$, overfitting becomes severe, and spurious correlations among predictors and with the response

proliferate. Interpretation of individual coefficients becomes unreliable, and standard hypothesis tests lose their usual meaning. Analysts must therefore apply regularization methods (such as ridge, lasso, or elastic net), use cross-validation for tuning parameter selection, carefully standardize predictors, and often employ dimensionality reduction techniques like principal component analysis or screening. In high dimensions, emphasis should be placed on predictive accuracy and stability rather than on naive inference about individual coefficients.

24. Regression with step functions and piecewise polynomials

Step functions divide the predictor range into intervals and fit constant values to each, capturing abrupt changes in the relationship between X and Y but lacking smoothness and potentially creating artificial jumps at cutpoints. Piecewise polynomials expand this idea by fitting separate polynomials within each interval, adding flexibility and allowing more complex shapes, but they may introduce discontinuities in the function or its derivatives at knot locations. To avoid unrealistic behavior, continuity and smoothness constraints are imposed across knots, leading toward spline constructions. These models offer interpretable, localized fits but require careful selection of the number and location of cutpoints or knots to avoid overfitting or underfitting the underlying relationship.

25. Regression with linear and cubic splines

Linear splines fit piecewise linear segments joined at knots, producing functions that are continuous but may have kinks where the slope changes abruptly. They are useful for modeling relationships that change slope at specific values of the predictor, yet they may appear too rough if the underlying relationship is smooth. Cubic splines extend this approach by using piecewise cubic polynomials that are continuous along with their first and second derivatives at each knot, resulting in much smoother curves that follow data more naturally. Both linear and cubic splines are implemented via basis functions, allowing them to be estimated within standard linear regression frameworks, with cubic splines generally preferred for their superior smoothness and flexibility.

26. Regression with cubic and natural splines

Cubic splines can fit flexible nonlinear relationships but sometimes behave erratically near the boundaries of the data because there are no constraints on their curvature outside the range of the knots. Natural splines refine this by imposing additional boundary conditions that force the function to be linear beyond the boundary knots, which stabilizes extrapolation and prevents large oscillations at the edges. As a result, natural splines typically require fewer effective degrees of freedom than unconstrained cubic splines for similar fit quality and are often preferred in practice. Both cubic and natural splines provide smooth nonlinear fits, but natural splines offer better behavior at the extremes while retaining interpretable, low-dimensional representations of complex relationships.

27. Regression with smoothing splines

Smoothing splines estimate a smooth function by minimizing the residual sum of squares augmented with a penalty on the integrated squared second derivative, thereby balancing fidelity to the observed data with a preference for smoothness. The tuning parameter λ controls this trade-off: for λ close to zero the fit interpolates the data and is highly wiggly, whereas for large λ the function approaches a straight line. The solution to this optimization problem is always a natural cubic spline with knots at each unique observed x-value, but with adaptively chosen smoothness dictated by λ. In practice, cross-validation is typically used to select λ, making smoothing splines a powerful and principled nonparametric regression tool.

28. Local regression and generalized additive models (GAMs)

Local regression (LOESS) fits simple models, such as local linear regressions, in neighborhoods around each target point, weighting observations by their proximity so as to capture complex nonlinear patterns without specifying a global functional form. The span parameter determines how many nearby observations influence each local fit and thus controls the smoothness of the estimated function. Generalized additive models (GAMs) extend linear models by allowing each predictor to enter the model through its own smooth function, estimated using methods like splines or local regression, while retaining an additive structure. This additivity preserves interpretability and avoids the curse of dimensionality: each component $f_j(x_j)$ can be visualized separately, yet the model can capture rich nonlinear effects and, if desired, interactions via bivariate smooths or tensor products.