Data Science Notes

Bhavjot Khurana

November 6, 2025

Contents

1 Introduction

Using the freecodecamp.ord video as a reference (link: `https://www.youtube.com/watch?v=XU5pw3QRYjQ`). This document contains notes on data science topics covered in the video.

2 Linear Regression

2.1 Definition

Linear regression models the expected value of a response variable y as a linear function of a single predictor x:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim \text{Normal}(0, \sigma^2)$$

It assumes the errors ε_i are independent, normally distributed, and have constant variance.

2.2 Key Formulas

Ordinary least squares (OLS) estimates the slope and intercept by minimizing the sum of squared residuals:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

2.3 Error Function

The residual for observation i captures the prediction error:

$$e_i = y_i - \hat{y}_i$$

Residuals tell us how far each point lies from the regression line; examining their pattern helps spot outliers or violations of model assumptions.

2.4 Mean Squared Error (MSE)

Mean Squared Error is the average of squared residuals:

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

Errors are squared to penalize large deviations more heavily, eliminate sign cancellation, and produce a smooth, differentiable loss function that calculus-based solvers can optimize.

2.5 How the Line is Fitted

Fitting the line involves solving the optimization problem:

$$\min_{\beta_0,\beta_1}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

Setting the partial derivatives to zero yields the normal equations (matrix form: $(X^\top X)\hat{\beta} = X^\top y$). Software uses closed-form solutions for small problems or matrix decompositions (QR or SVD) for numerical stability. Gradient methods (e.g., gradient descent) offer scalable alternatives for very large datasets.

2.6 Hypothesis Testing and p-values

To test whether x helps explain y, use the t-test for the slope. The null hypothesis $H_0 : \beta_1 = 0$ indicates no linear relationship. The statistic

$$t = \frac{\hat{\beta}_1}{\text{SE}(\hat{\beta}_1)}$$

follows a t-distribution with $n - 2$ degrees of freedom. A small p-value suggests the predictor provides statistically significant explanatory power.

2.7 Residual Standard Error (RSS and TSS)

Two sums of squares quantify variability:

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \qquad \text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

Residual Standard Error (RSE) estimates the typical size of residuals in the units of y:

$$\text{RSE} = \sqrt{\frac{\text{RSS}}{n-2}}$$

Lower RSE indicates tighter fit around the regression line.

2.8 Interpretations

- **Slope** $\hat{\beta}_1$: Expected change in y for a one-unit increase in x.
- **Intercept** $\hat{\beta}_0$: Expected value of y when $x = 0$, useful when $x = 0$ is meaningful.
- $R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$: Proportion of variance in y explained by the model.
- **Practical use**: Combine coefficient estimates with confidence or prediction intervals to communicate both central tendency and uncertainty in predictions.

2.9 Example: Simple Linear Regression Output

Using the `Advertising` dataset (TV spend predicting sales), the `statsmodels` summary reports the following key statistics:

Statistic	Estimate	Interpretation
Intercept ($\hat{\beta}_0$)	7.03	Baseline sales when TV spend is \$0k.
TV coefficient ($\hat{\beta}_1$)	0.0475	Each \$1k on TV adds approximately 0.048k units sold.
R^2	0.612	61.2% of variance in sales explained by TV.
F-statistic	312.1 (p $< 10^{-40}$)	The model is highly significant overall.

The printed OLS table also shows small standard errors and very large t-statistics for the slope, so the p-value for the TV coefficient is essentially zero. Residual diagnostics (Omnibus, Jarque-Bera, Durbin-Watson) indicate roughly normal residuals and limited autocorrelation for this example.

2.10 Interview Questions and Answers

1. **Question:** What are the four core OLS assumptions in simple linear regression, and how can you diagnose violations?

Solution: (i) Linearity: inspect scatterplots or residuals vs. fitted plots for curvature; (ii) Independence: examine residual autocorrelation plots or the Durbin-Watson statistic; (iii) Homoscedasticity: look for constant residual spread in residuals vs. fitted plots or run a Breusch-Pagan test; (iv) Normality of errors: review QQ-plots or apply the Shapiro-Wilk test. If violations appear, consider transforming variables, adding features, or using robust methods.

2. **Question:** The following summary statistics are reported for a dataset: $\sum(x_i - \bar{x})(y_i - \bar{y}) = 120$ and $\sum(x_i - \bar{x})^2 = 30$. Compute $\hat{\beta}_1$ and interpret it.

   Solution: $\hat{\beta}_1 = 120/30 = 4$. On average, a one-unit increase in x is associated with a 4-unit increase in y.

3. **Question:** If $\hat{\beta}_0 = 1.5$ and $\hat{\beta}_1 = 2.0$, what is the fitted value and residual for an observation with $x = 6$ and $y = 12$?

   Solution: $\hat{y} = 1.5 + 2.0 \times 6 = 13.5$. The residual is $e = y - \hat{y} = 12 - 13.5 = -1.5$, indicating the model over-predicts this point by 1.5 units.

4. **Question:** Given TSS = 200 and RSS = 60, compute R^2. What does the result imply?

   Solution: $R^2 = 1 - (60/200) = 0.70$. The model explains 70% of the variance in y; the remaining 30% is unexplained noise or structure not captured by the single predictor.

5. **Question:** During diagnostic review, the residual vs. fitted plot shows a funnel shape (variance increases with fitted values). What issue does this suggest and how can you address it?

   Solution: The pattern suggests heteroscedasticity. Remedies include transforming y (log or Box-Cox), modeling the variance directly (weighted least squares), or adopting heteroscedasticity-robust standard errors when inference is the focus.

3 Multiple Linear Regression

3.1 Definition

Multiple Linear Regression (MLR) extends the linear model to p predictors:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

Coefficients measure the expected change in y for a one-unit change in the corresponding predictor while holding others constant.

3.2 F-statistic and Interpretation

The model-wide F-test compares the explained variance to unexplained variance:

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n - p - 1)}$$

A large F-statistic (with a small p-value) indicates that, taken together, the predictors explain significantly more variation than an intercept-only model. If the F-test is not significant, it implies the collective set of predictors may not offer meaningful predictive power beyond the mean of y.

3.3 Example: Multiple Linear Regression Output

Fitting TV, radio, and newspaper spend simultaneously to predict sales yields the following summary highlights:

Statistic	Estimate	Interpretation
Intercept	2.94	Baseline sales when all spends are zero.
TV coefficient	0.0458	Positive, highly significant effect (p $< 10^{-50}$).
Radio coefficient	0.1885	Strong positive effect (p $< 10^{-40}$).
Newspaper coefficient	-0.0010	Not significant ($p \approx 0.86$).
R^2	0.897	89.7% of sales variance explained jointly.
F-statistic	570.3 (p $< 10^{-90}$)	Model is overwhelmingly significant.

Compared with the simple-regression model, R^2 jumps from 0.61 to 0.90, showing the value of additional predictors. However, the newspaper coefficient's large p-value signals it may not contribute meaningfully, suggesting a model refinement (e.g., dropping the variable) could be appropriate.

3.4 Interview Questions and Answers

1. **Question:** In the model $\hat{y} = 2.9 + 0.046\,\text{TV} + 0.189\,\text{Radio} - 0.001\,\text{Newspaper}$, how do you interpret the radio coefficient?

   Solution: Holding TV and newspaper spend fixed, an additional \$1k on radio is associated with an average increase of 0.189k units sold. The ceteris paribus clause is critical in multiple regression interpretations.

2. **Question:** Suppose $n = 60$, $p = 3$, and $R^2 = 0.82$. Compute the adjusted R^2.

   Solution: $R^2_{\text{adj}} = 1 - (1 - 0.82)\frac{n-1}{n-p-1} = 1 - 0.18 \times \frac{59}{56} \approx 1 - 0.1896 = 0.8104$. The slight drop reflects the penalty for adding predictors relative to the sample size.

3. **Question:** How can you detect multicollinearity, and what are two mitigation strategies?

   Solution: Examine variance inflation factors (VIFs) or the condition number of $X^\top X$. Mitigations include removing or combining correlated predictors, applying dimensionality reduction (PCA), or using regularization methods such as ridge regression.

4. **Question:** You fit a baseline model with TV and radio predictors ($\text{RSS}_0 = 120$, $p_0 = 2$) and an extended model that adds newspaper ($\text{RSS}_1 = 110$, $p_1 = 3$) on $n = 200$ observations. Conduct the partial F-test for the added variable.

   Solution: $F = \frac{(\text{RSS}_0 - \text{RSS}_1)/(p_1 - p_0)}{\text{RSS}_1/(n-p_1-1)} = \frac{10/1}{110/196} \approx 17.82$. Compare to an $F_{1,196}$ distribution; the large value yields a tiny p-value, supporting inclusion of the newspaper predictor.

5. **Question:** Why might you standardize predictors before fitting an MLR model, and how does that affect coefficient interpretation?

   Solution: Standardization removes scale differences, aiding numerical stability and making coefficient magnitudes comparable (useful for feature importance and regularization). After standardization, a coefficient represents the change in y for a one standard deviation increase in that predictor, holding others fixed.

4 Logistic Regression

4.1 Problem Setup

Binary logistic regression models the log-odds of a positive outcome as a linear function of input features:

$$\log \frac{\Pr(y_i = 1 \mid \mathbf{x}_i)}{1 - \Pr(y_i = 1 \mid \mathbf{x}_i)} = \beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}$$

Applying the logistic function $\sigma(z) = 1/(1+e^{-z})$ converts log-odds to probabilities. Each observation is classified by comparing $\hat{p}_i = \sigma(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})$ to a decision threshold.

4.2 Estimation and Decision Rules

Parameters maximize the log-likelihood

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)]$$

via iteratively reweighted least squares or gradient-based solvers. Regularized variants (L1, L2) shrink coefficients to curb overfitting and handle correlated features. The default classification rule uses $\tau = 0.5$, but business costs often motivate custom thresholds or probability calibration.

4.3 Interpreting Coefficients

Coefficient β_j represents the change in log-odds from a one-unit increase in x_{ij} holding others fixed. Exponentiating yields an odds ratio: $e^{\beta_j} > 1$ increases the odds of the event, $e^{\beta_j} < 1$ decreases them. Wald tests and likelihood-ratio tests assess whether predictors contribute significantly.

4.4 Model Diagnostics and Metrics

Model quality is summarized with the confusion matrix:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Key metrics derived from these counts include

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

ROC and precision-recall curves help visualize threshold trade-offs, while calibration plots verify that predicted probabilities align with observed frequencies.

4.5 Handling Class Imbalance

Class imbalance can mask poor minority-class performance. Common strategies include resampling, synthetic sample generation (SMOTE), class-weighted loss functions, or adjusting the decision threshold to prioritize recall or precision as required.

4.6 Example: Email Spam Detection

Using a spam detection dataset, the logistic regression summary highlights the effect of key tokens on the log-odds of the spam class: Coefficients quantify how tokens push messages toward spam or ham. Reviewing ROC curves and calibration plots ensures probabilities match observed frequencies.

Feature	Coefficient	Interpretation
Intercept	-1.92	Base log-odds favor “ham” when no spam tokens appear.
‘‘free’’ count	0.78	Each occurrence multiplies spam odds by $e^{0.78} \approx 2.18$.
‘‘money’’ count	0.55	Positive association with spam likelihood.
email length	-0.21	Longer emails slightly decrease spam odds.
ROC-AUC	0.947	Excellent separability between spam and ham.

4.7 Interview Questions and Answers

1. **Question:** How does logistic regression differ from linear regression in terms of model output and optimization objective?

   Solution: Logistic regression produces probabilities through the sigmoid link and maximizes log-likelihood (cross-entropy), whereas linear regression outputs unbounded continuous predictions and minimizes squared error. The probabilistic formulation enables classification metrics and likelihood-based inference.

2. **Question:** Interpret a coefficient $\beta_j = 0.4$ in a standardized logistic regression model.

   Solution: A one standard deviation increase in feature x_j multiplies the odds of the positive class by $e^{0.4} \approx 1.49$ (a 49% increase), holding all other predictors fixed.

3. **Question:** You observe 90% accuracy but only 40% recall on the positive class. What actions do you take?

Solution: The model misses many positives. Lower the threshold, tune class weights, resample the minority class, or optimize directly on recall/F$_1$ to capture more positives while monitoring precision and business cost.

4. **Question:** Why is feature scaling important for penalized logistic regression?

   Solution: Penalties such as L1/L2 act on coefficient magnitudes. Without scaling, predictors with larger numeric ranges shrink more aggressively, biasing feature selection. Standardization ensures the penalty treats predictors comparably.

5. **Question:** When would you apply probability calibration methods like Platt scaling?

   Solution: Apply them when predicted probabilities are poorly calibrated—for example, scores near 0.8 correspond to positives only 60% of the time. Calibrated probabilities support better risk-based decisions and cost-sensitive thresholds.

5 Multiple Logistic Regression & Multiclass Classification

5.1 Multiple Logistic Regression

Multiple logistic regression retains the binary outcome but includes several predictors:

$$\Pr(y_i = 1 \mid \mathbf{x}_i) = \sigma\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}\right)$$

Interpretation still relies on odds ratios, but with the "all else equal" proviso. Assess collinearity (e.g., variance inflation factors) and consider interaction terms when domain knowledge suggests multiplicative effects.

5.2 Feature Engineering and Regularization

Creating indicator variables for categorical features, standardizing numeric predictors, and engineering interaction terms often improve fit. Regularization choices matter: L1 (lasso) performs feature selection, L2 (ridge) stabilizes coefficients, and elastic net blends both—use cross-validation to tune penalty strength.

5.3 Multiclass Logistic Regression

When $K > 2$ classes are present, multinomial logistic regression employs the softmax function:

$$\Pr(y_i = k \mid \mathbf{x}_i) = \frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta}_k)}{\sum_{j=1}^{K} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_j)}, \qquad k = 1, \dots, K$$

Training minimizes multiclass cross-entropy. Coefficients compare class k to a reference class, so inspect pairwise log-odds to interpret feature effects.

5.4 One-vs-Rest and One-vs-One Strategies

An alternative is to decompose the multiclass problem into binary subproblems. One-vs-rest trains K classifiers, each discriminating one class against all others. One-vs-one trains a classifier for each class pair, combining votes at prediction time. Choose based on class imbalance and computational cost.

5.5 Evaluation for Multiclass Problems

Extend the confusion matrix to $K \times K$ and compute per-class precision/recall. Macro-averaging treats classes equally by averaging per-class metrics, highlighting minority-class performance. Micro-averaging aggregates counts before computing metrics, reflecting overall accuracy in imbalanced datasets.

5.6 Interview Questions and Answers

1. **Question:** Differentiate multiple logistic regression from simple logistic regression.

   Solution: Simple logistic regression uses a single predictor; multiple logistic regression incorporates several predictors simultaneously, enabling “all else equal” interpretations and capturing combined effects, but requiring vigilance for multicollinearity.

2. **Question:** Explain the one-vs-rest strategy and when it may struggle.

   Solution: One-vs-rest trains a binary classifier per class. It can falter when classes are heavily imbalanced or overlap, because each classifier faces skewed data and conflicting boundaries.

3. **Question:** How do macro-averaged and micro-averaged F_1 scores differ?

Solution: Macro averages treat each class equally by averaging per-class F_1 scores, emphasizing minority classes. Micro combines all TP/FP/FN before computing F_1, weighting metrics by class frequency for an overall view.

4. **Question:** Why should features be standardized before fitting k-NN or SVM classifiers for multiclass tasks?

   Solution: Both methods rely on distance calculations; unscaled features dominate the distance metric, distorting boundaries. Standardization balances feature influence, ensuring fair contribution across predictors.

5. **Question:** How would you evaluate a multiclass classifier when one class is rare yet critical?

   Solution: Inspect the per-class confusion matrix, monitor macro-averaged recall or precision-recall curves for the rare class, and adjust decision thresholds or class weights to prioritize that class's recall while constraining false positives.

6 Linear Discriminant Analysis

6.1 Motivation and Assumptions

Linear Discriminant Analysis (LDA) models class-conditional feature distributions as multivariate Gaussians with class-specific means $\boldsymbol{\mu}_k$ but a shared covariance matrix Σ. The assumptions imply homogeneous scatter within each class and linear decision boundaries between classes.

6.2 Derivation of the Discriminant Function

Bayes' rule yields class posterior probabilities

$$\Pr(y = k \mid \mathbf{x}) \propto \pi_k \cdot \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma),$$

where π_k is the prior probability of class k. Taking logs and discarding constants leads to a linear discriminant score:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \Sigma^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^\top \Sigma^{-1} \boldsymbol{\mu}_k + \log \pi_k.$$

Assign $\mathbf{x}$ to the class with the largest $\delta_k(\mathbf{x})$. The resulting boundary between any two classes k and j is a hyperplane.

6.3 Parameter Estimation

Estimate class priors via sample proportions $\hat{\pi}_k = n_k/n$. Sample means follow $\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i:y_i=k}\mathbf{x}_i$. The pooled covariance estimator

$$\hat{\Sigma} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^\top$$

captures within-class scatter. LDA is statistically efficient when assumptions hold, but sensitive to outliers and class covariance differences.

6.4 Dimensionality Reduction via Discriminant Axes

For K classes, LDA identifies up to $K-1$ discriminant directions maximizing between-class variance relative to within-class variance. Projecting data onto these axes visualizes class separation and provides compact features for downstream classifiers (e.g., logistic regression or k-NN on the discriminant scores).

6.5 Comparison with Logistic Regression

Both methods yield linear decision boundaries under their respective assumptions. Logistic regression focuses on conditional probabilities $\Pr(y \mid \mathbf{x})$ without distributional assumptions on $\mathbf{x}$; LDA models $\Pr(\mathbf{x} \mid y)$ and can outperform logistic regression when the Gaussian assumption is valid and sample size is limited. Conversely, logistic regression is more robust when covariance structures differ across classes or the Gaussian model is misspecified.

6.6 Example: Iris Species Classification

Applying LDA to the classic Iris dataset with four petal/sepal measurements: Plotting the first discriminant scores clearly separates Setosa from the other species, while the second discriminant axis aids in distinguishing Versicolor vs. Virginica.

6.7 Interview Questions and Answers

1. **Question:** What assumptions does LDA make about the data generating process?

Statistic	Value	Interpretation
Training accuracy	0.973	Strong in-sample separation along discriminant axes.
First discriminant variance ratio	0.991	Nearly all separability captured in a single axis.
Mean vector difference (Setosa vs. Versicolor)	Large in sepal length	Confirms visual separation.
Pooled covariance determinant	0.14	Shared covariance assumption reasonable.

Solution: Each class follows a multivariate Gaussian distribution with its own mean but a shared covariance matrix, and class priors are fixed. Violations (e.g., unequal covariance) push the method toward Quadratic Discriminant Analysis.

2. **Question:** How do you interpret the discriminant function $\delta_k(\mathbf{x})$?

   Solution: It scores how likely $\mathbf{x}$ belongs to class k. The first term measures alignment with the class mean, the second penalizes distance from the mean, and $\log \pi_k$ incorporates class prevalence. Largest score wins.

3. **Question:** When might you prefer LDA over logistic regression?

   Solution: When Gaussian assumptions are reasonable, classes have similar covariance, and the dataset is small relative to feature count. LDA leverages structure to reduce variance and can outperform discriminative models.

4. **Question:** How do you handle LDA when the number of predictors exceeds observations?

   Solution: The covariance estimate becomes singular. Apply regularized LDA (shrink covariance toward the identity), perform dimensionality reduction (PCA) before LDA, or switch to penalized discriminant analysis.

5. **Question:** How would you evaluate whether the equal covariance assumption holds?

   Solution: Compare sample covariance matrices, use Box's M test, or inspect residual plots along discriminant axes. Large discrepancies suggest considering QDA or heteroscedastic models.

7 Quadratic Discriminant Analysis

7.1 Motivation and Assumptions

Quadratic Discriminant Analysis (QDA) relaxes LDA's equal covariance assumption by allowing each class k to have its own covariance matrix Σ_k. As a result, class-conditional densities remain Gaussian but decision boundaries become quadratic surfaces, capturing curved separations between classes.

7.2 Discriminant Function

Starting from Bayes' rule with class-specific covariances,

$$\Pr(y = k \mid \mathbf{x}) \propto \pi_k \cdot \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k).$$

Taking logs yields

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \log \pi_k,$$

which is quadratic in $\mathbf{x}$. Predict by choosing the class with the largest $\delta_k(\mathbf{x})$.

7.3 Parameter Estimation and Complexity

Class priors and means match LDA: $\hat{\pi}_k = n_k/n$, $\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i:y_i=k} \mathbf{x}_i$. Covariance matrices use class-specific sample covariances

$$\hat{\Sigma}_k = \frac{1}{n_k - 1}\sum_{i:y_i=k}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^\top.$$

Estimating a full covariance per class introduces $\mathcal{O}(p^2K)$ parameters, so QDA demands larger sample sizes than LDA to remain stable. Regularized QDA shrinks $\hat{\Sigma}_k$ toward a diagonal or pooled estimate to mitigate variance.

7.4 When to Prefer QDA over LDA

QDA shines when classes exhibit distinct covariance structures or nonlinear boundaries that LDA cannot capture. However, variance grows quickly in high dimensions with limited data, so cross-validation should confirm that added flexibility improves out-of-sample accuracy. If covariances are nearly equal, QDA may overfit and LDA can be superior.

7.5 Example: Credit Risk Segmentation

Applying QDA to a credit dataset with features like income, debt-to-income ratio, and credit score: The richer boundary reduces high-risk false negatives after tuning the shrinkage parameter via grid search.

Statistic	Value	Interpretation
Validation accuracy	0.842	QDA outperforms LDA (0.803) due to curved boundaries.
Average log determinant difference	1.27	Covariance volume differs meaningfully by class.
Regularization parameter	0.05	Mild shrinkage stabilizes covariance estimates.
Misclassified high-risk loans	Reduced by 18%	Better capture of heteroskedastic patterns.

7.6 Interview Questions and Answers

1. **Question:** How do the decision boundaries of QDA differ from LDA?

   Solution: QDA produces quadratic (curved) boundaries because each class has its own covariance matrix, while LDA yields linear boundaries due to a common covariance assumption.

2. **Question:** What data conditions motivate QDA despite its higher variance?

   Solution: Use QDA when class covariance matrices differ substantially or the Bayes-optimal boundary is curved. Sufficient data per class mitigates the increased parameter count.

3. **Question:** How can you regularize QDA when sample size is limited relative to predictors?

   Solution: Shrink each $\hat{\Sigma}_k$ toward the identity or pooled covariance (e.g., using a convex combination), apply diagonal covariance assumptions, or reduce dimensionality with PCA before fitting.

4. **Question:** Compare QDA with logistic regression in terms of assumptions and flexibility.

   Solution: QDA models $\Pr(\mathbf{x} \mid y)$ with Gaussian densities and allows class-specific covariance, capturing curved decision boundaries. Logistic regression models $\Pr(y \mid \mathbf{x})$ directly, assuming a linear log-odds relationship; it is less flexible for nonlinear boundaries unless augmented with feature engineering.

5. **Question:** How would you evaluate whether QDA is overfitting?

   Solution: Monitor cross-validated performance, inspect confusion matrices for variance across folds, and compare against simpler models (LDA, regularized QDA). Large train-accuracy gains without validation gains signal overfitting.

8 Resampling and Regularization

8.1 Cross-Validation Strategies

Resampling evaluates model performance on unseen data by repeatedly partitioning the dataset. Popular schemes include:

- **Holdout**: Split once into train/validation; fast but variance can be high.
- **k-fold cross-validation**: Partition into k folds, train on $k-1$ folds, validate on the holdout fold, and average metrics; balances bias and variance.
- **Stratified k-fold**: Preserves class ratios in each fold, critical for imbalanced classification.
- **Leave-one-out (LOOCV)**: Special case with $k = n$; low bias but computationally expensive and high variance.
- **Time-series splits**: Maintain chronological order (rolling-origin) to respect temporal dependencies.

Use cross-validation to compare models, tune hyperparameters, and estimate generalization error before final training on all data.

8.2 Bootstrap for Uncertainty

The bootstrap samples the data with replacement to approximate the sampling distribution of an estimator. For model assessment:

1. Draw B bootstrap samples of size n.
2. Train the model on each sample and evaluate on the out-of-bag observations.
3. Aggregate metrics (mean, standard deviation) to quantify variability and construct confidence intervals.

Bootstrap methods are especially helpful when analytic variance formulas are complex or unavailable.

8.3 Regularization Overview

Regularization adds penalty terms to the loss function, trading bias for reduced variance to prevent overfitting:

$$\text{Loss}_{\text{reg}} = \text{Loss}_{\text{empirical}} + \lambda \cdot \Omega(\boldsymbol{\beta}),$$

where λ controls penalty strength and Ω encodes the desired constraint.

- **Ridge (L2)**: $\Omega(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2$. Shrinks coefficients toward zero, stabilizing models with correlated predictors.
- **Lasso (L1)**: $\Omega(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$. Encourages sparsity, performing feature selection by driving some coefficients exactly to zero.
- **Elastic Net**: Combines L1 and L2 penalties; useful when predictors are correlated and sparsity is desired.
- **Regularized logistic/LDA/QDA**: Adds penalties to classification models to control decision boundary complexity or shrink covariance estimates.

Select λ using cross-validation or information criteria (AIC, BIC) to balance fit and complexity.

8.4 Bias-Variance Trade-off

Resampling reveals how training vs. validation error evolves with model flexibility. Regularization increases bias slightly but reduces variance, often lowering validation error. Plotting learning curves or cross-validation error against λ helps diagnose underfitting and overfitting.

8.5 Model Selection Workflow

1. Define candidate models and features.
2. Standardize or normalize predictors where required (e.g., before applying L1/L2 penalties).

3. Use stratified k-fold cross-validation to tune hyperparameters such as λ, penalty mix, or tree depth.

4. Evaluate final model on a held-out test set to obtain an unbiased estimate of performance.

5. Retrain on the full dataset with chosen hyperparameters before deployment.

8.6 Case Study: Ridge Regression

Ridge regression extends linear regression by adding an L2 penalty:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_2^2.$$

The normal equations become $(X^\top X + \lambda I)\boldsymbol{\beta} = X^\top y$, ensuring invertibility even when predictors are collinear or $p > n$. Ridge shrinks coefficients smoothly toward zero, reducing variance while retaining all predictors. Key diagnostics:

- **Coefficient paths**: Plot β_j vs. λ to observe shrinkage behavior and assess stability.
- **Effective degrees of freedom**: $\mathrm{df}(\lambda) = \mathrm{tr}\left(X(X^\top X + \lambda I)^{-1} X^\top\right)$; larger λ lowers model complexity.
- **Cross-validated error**: Select λ minimizing validation error or the "one-standard-error" rule for a simpler model.

Standardize predictors before fitting to ensure the penalty treats features on comparable scales.

8.7 Interview Questions and Answers

1. **Question:** Why is stratified cross-validation important for classification?

   Solution: It preserves class proportions in each fold, yielding more stable estimates for metrics like recall/precision, especially when classes are imbalanced.

2. **Question:** Compare ridge and lasso regularization in terms of their effect on coefficients.

   Solution: Ridge shrinks coefficients continuously toward zero without eliminating them; lasso can set coefficients exactly to zero, selecting a sparse subset of features. Elastic net blends both behaviors.

3. **Question:** How would you choose the regularization strength λ?

   Solution: Evaluate a grid (or use algorithms like coordinate descent) within cross-validation, selecting the λ that minimizes validation error or lies within one standard error of the minimum to favor simpler models.

4. **Question:** When is the bootstrap preferable to k-fold cross-validation?

   Solution: Bootstrap is useful for estimating parameter uncertainty or when the dataset is too small for reliable folds; it provides confidence intervals for metrics by resampling with replacement.

5. **Question:** How does regularization mitigate multicollinearity?

   Solution: Penalties like ridge constrain coefficient magnitude, reducing sensitivity to correlated predictors and stabilizing estimates, leading to lower variance in predictions.

9 Decision Trees

9.1 Structure and Intuition

Decision trees partition the feature space into axis-aligned regions by recursively splitting on feature thresholds. Each internal node applies a decision rule (e.g., feature$_j \leq t$), and leaves output predicted values (regression) or class labels/probabilities (classification). Trees capture nonlinear relationships and feature interactions without explicit feature engineering.

9.2 Recursive Binary Splitting (Top-Down Greedy)

The CART algorithm grows trees via recursive binary splitting: starting at the root, evaluate all candidate feature-threshold pairs and choose the split that yields the largest impurity reduction. Repeat this top-down greedy procedure on each child node until a stopping rule triggers (minimum samples, maximum depth, or purity). Although greedy, the method performs well in practice and serves as the basis for modern tree ensembles.

9.3 Regression Trees and RSS

Regression trees predict continuous outcomes by minimizing the residual sum of squares within each node. For a node with observations $\mathcal{D}$, the impurity is

$$\mathrm{RSS}(\mathcal{D}) = \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \bar{y}_{\mathcal{D}})^2,$$

where $\bar{y}_{\mathcal{D}}$ is the mean response in the node. Candidate splits are scored by the drop in RSS between the parent and its children. Leaves store the average response of training points that fall into the region.

9.4 Classification Trees and Error Metrics

Classification trees rely on class proportions in each node. Common impurity measures include:

- **Classification error rate**: $1 - \max_k p_k$; easy to interpret but less sensitive to class changes.
- **Gini impurity**: $1 - \sum_k p_k^2$; smooth and fast to compute.
- **Entropy**: $-\sum_k p_k \log p_k$; strongly penalizes rare-class mixtures.

During pruning or model comparison, classification error rate provides a high-level accuracy view, while Gini/entropy guide split decisions.

9.5 Pruning and Regularization

Fully grown trees often overfit. Apply pre-pruning (limit depth, minimum samples per leaf, maximum leaf count) or post-pruning (grow a large tree then prune back using cost-complexity pruning with parameter α). Cross-validation selects the pruning parameter that balances bias and variance. Ensembles such as random forests and gradient boosted trees further reduce variance and improve generalization.

9.6 Handling Continuous and Categorical Features

Continuous features are split by thresholds; categorical features require either one-hot encoding or subset splits. When categories are high-cardinality, group infrequent levels or use target encoding carefully to avoid leakage.

9.7 Strengths and Limitations

- **Strengths**: Interpretable structure; handle mixed data types; no scaling needed; nonlinear decision boundaries.
- **Limitations**: High variance; piecewise-constant predictions; biases toward features with many levels; unstable to small data perturbations.

Bagging, random forests, and boosting address many weaknesses by aggregating multiple trees.

9.8 Ensemble Extensions: Bagging, Random Forests, Boosting

- **Bagging (Bootstrap Aggregation)**: Train many deep trees on bootstrap samples and average their predictions. Reduces variance and improves stability.
- **Random Forests**: Bagging plus random feature subsampling at each split, decorrelating trees and yielding better performance than plain bagging.
- **Boosting**: Sequentially fit shallow trees to residuals or gradients (e.g., AdaBoost, Gradient Boosted Trees). Each tree focuses on examples the previous ensemble mispredicted, reducing bias with careful learning-rate control.

Hyperparameters (number of trees, depth, learning rate) are tuned via cross-validation to balance bias and variance.

9.9 Example: Customer Churn Classification

Training a depth-limited tree on customer behavior metrics yields: Tree visualization highlights churn segments (e.g., short-tenure customers with high support call counts).

Statistic	Value	Interpretation
Max depth	5	Controls model complexity.
Validation accuracy	0.812	Competitive with logistic baseline (0.789).
Top feature importances	Tenure, Support calls, Monthly charges	Key churn drivers.
Pruning parameter α	0.01	Selected via cross-validation to reduce overfitting.

9.10 Interview Questions and Answers

1. **Question:** How does Gini impurity differ from entropy when splitting nodes?

   Solution: Both measure class mix; Gini is simpler $(1-\sum p_k^2)$, entropy is $-\sum p_k \log p_k$. They often select similar splits, but entropy penalizes rare classes slightly more.

2. **Question:** Why do decision trees tend to overfit, and how can you mitigate that?

   Solution: Trees can memorize training data by growing deep, low-sample leaves. Mitigate using pruning, depth/min-samples constraints, or ensemble methods (bagging/boosting) that reduce variance.

3. **Question:** What bias might arise when splitting on categorical variables with many levels?

   Solution: Features with many categories can appear more informative because they create numerous small pure splits, leading to biased feature selection. Mitigate by collapsing categories, using feature hashing, or applying corrected splitting criteria.

4. **Question:** When would you prefer a single decision tree over a random forest?

   Solution: When interpretability is paramount and the dataset is small or low-noise. Otherwise, random forests generally outperform single trees by averaging variance.

5. **Question:** How do you evaluate feature importance in tree models?

   Solution: Sum the impurity reduction contributed by each feature across splits, or use permutation importance by measuring performance drop when feature values are shuffled.

6. **Question:** Explain the top-down greedy nature of recursive binary splitting and one consequence of its greediness.

   Solution: The algorithm picks the best immediate split without considering future splits; this can miss globally optimal trees. Pruning and cross-validation help correct overfitting caused by greedy decisions.

7. **Question:** Contrast bagging, random forests, and boosting for tree-based models.

   Solution: Bagging averages many independent trees to lower variance; random forests add feature subsampling to further decorrelate trees; boosting fits trees sequentially to residuals to reduce bias. Choice depends on variance vs. bias trade-offs and computation.

10 Support Vector Machines

10.1 Maximal Margin Classifier

In the separable case, Support Vector Machines (SVMs) seek the hyperplane that maximizes the margin—the distance from the hyperplane to the nearest training points. For labeled data $(\mathbf{x}_i, y_i)$ with $y_i \in \{-1, +1\}$, the maximal margin problem is

$$\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|} \quad \text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \ \forall i.$$

The support vectors are the points lying exactly on the margin boundaries and fully determine the optimal separating hyperplane.

10.2 Hyperplanes and Geometric View

A hyperplane in p-dimensions is described by $\{\mathbf{x} : \mathbf{w}^\top \mathbf{x} + b = 0\}$. Its normal vector $\mathbf{w}$ governs orientation, while $b/\|\mathbf{w}\|$ sets the offset from the origin. The signed distance of a point $\mathbf{x}$ to the hyperplane equals $\frac{\mathbf{w}^\top \mathbf{x}+b}{\|\mathbf{w}\|}$, so maximizing the margin corresponds to minimizing $\|\mathbf{w}\|$ under the classification constraints.

10.3 Soft Margin Support Vector Classifier

Real-world data rarely are perfectly separable. The soft margin SVM introduces slack variables $\xi_i \geq 0$ to allow margin violations:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i.$$

The penalty parameter C controls the trade-off between margin width and misclassification tolerance. In the dual formulation, the optimization depends only on inner products $\mathbf{x}_i^\top \mathbf{x}_j$, paving the way for kernel methods.

10.4 Kernel Trick and Nonlinear Decision Boundaries

Replacing inner products with kernel functions $K(\mathbf{x}_i, \mathbf{x}_j)$ implicitly maps points into a high-dimensional feature space without explicit transformation.

- **Linear kernel**: recovers the soft margin classifier.

- **Polynomial kernel**: $K(\mathbf{x}, \mathbf{z}) = (\gamma \mathbf{x}^\top \mathbf{z} + r)^d$ captures feature interactions up to degree d.
- **Radial Basis Function (RBF)**: $K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2)$ yields flexible, localized decision boundaries.

Kernel choice and hyperparameters (C, γ, d) are tuned via cross-validation.

10.5 Model Selection and Practical Considerations

- **Feature scaling**: Standardize features before training; SVMs are sensitive to feature magnitude because the margin is Euclidean.
- **Class imbalance**: Adjust class weights or decision thresholds (C per class) to balance precision and recall.
- **SVM regression (SVR)**: Uses an ε-insensitive loss to fit regression models with similar margin-based principles.
- **Computational cost**: Training complexity grows with n; linear SVM solvers and approximate kernels help on large datasets.

10.6 Example: RBF-SVM for Handwritten Digits

Using standardized pixel intensities to classify digits 0–9: Grid search over (C, γ) with stratified folds identified the best trade-off between bias and variance.

Setting	Value	Notes
Kernel	RBF ($\gamma = 0.015$)	Captures nonlinear digit boundaries.
Penalty C	10	Balances margin and errors.
Cross-validated accuracy	0.983	Beats logistic baseline (0.944).
Support vectors	3,120 ($\approx$ 30% of data)	Points defining the decision surface.

10.7 Interview Questions and Answers

1. **Question:** Define the maximal margin classifier and explain when it is feasible.

   Solution: It is the hard-margin SVM that maximizes the geometric margin subject to perfect separation; feasible only when data are linearly separable.

2. **Question:** How does the parameter C influence the soft margin SVM?

   Solution: Large C penalizes misclassifications heavily, favoring narrower margins and lower bias but higher variance; small C allows wider margins with more violations, increasing bias and robustness to noise.

3. **Question:** What role do support vectors play in prediction?

   Solution: Only support vectors have non-zero dual coefficients; predictions depend on their weighted contribution via $K(\mathbf{x}_i, \mathbf{x})$, making SVMs resilient to outliers far from the margin.

4. **Question:** Compare linear, polynomial, and RBF kernels in terms of flexibility.

   Solution: Linear kernels build linear boundaries; polynomial kernels add global interactions controlled by degree; RBF kernels generate highly flexible, localized boundaries. Choice depends on domain knowledge and data complexity.

5. **Question:** How would you diagnose overfitting in an SVM model?

   Solution: Monitor cross-validation gap versus training accuracy, inspect number of support vectors (very high proportion may signal overfitting), and visualize decision boundaries or learning curves across C and kernel parameters.

11 Unsupervised Learning

11.1 Overview

Unsupervised learning discovers structure in unlabeled data by modeling feature relationships without target variables. Common goals include dimensionality reduction (capturing dominant variation) and clustering (identifying groups of similar observations). Evaluation relies on internal metrics, visualization, or downstream task performance rather than ground truth labels.

11.2 Principal Component Analysis (PCA)

PCA finds orthogonal directions (principal components) maximizing variance. Given centered data matrix X, PCA solves the eigenvalue problem for the covariance matrix $S = \frac{1}{n-1}X^\top X$:

$$S\mathbf{v}_k = \lambda_k \mathbf{v}_k.$$

Component k explains variance λ_k; projection scores are $X\mathbf{v}_k$. Use PCA to reduce dimensionality, denoise data, and visualize high-dimensional structure. Standardize features beforehand to prevent scale-dominant variables.

11.3 Clustering Fundamentals

Clustering algorithms group points based on similarity metrics (Euclidean distance, cosine similarity). Key considerations:

- Choice of distance metric influences cluster shapes.
- Feature scaling affects similarity; standardization is often required.
- Internal validation metrics (silhouette score, Davies-Bouldin index) assess separation and cohesion.

11.4 k-means Clustering

k-means partitions observations into K clusters by minimizing within-cluster sum of squares:

$$\min_{\{\mathcal{C}_k\}_{k=1}^{K}} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2,$$

where centroid $\boldsymbol{\mu}_k$ is the mean of cluster $\mathcal{C}_k$. The standard algorithm alternates between assigning points to closest centroids and updating centroids. Initialization matters; k-means++ selects diverse seeds to improve convergence. The elbow method or silhouette analysis helps choose K.

11.5 Hierarchical Clustering

Hierarchical methods create nested clusterings visualized via dendrograms:

- **Agglomerative (bottom-up)**: Start with singletons; iteratively merge the closest clusters based on linkage (single, complete, average, Ward).
- **Divisive (top-down)**: Start with all points; recursively split clusters.

Ward linkage minimizes variance increase and behaves similarly to k-means. Cutting the dendrogram at a chosen height yields cluster assignments without pre-specifying K.

11.6 Example: Customer Segmentation Workflow

1. Standardize behavioral features (spend, frequency, tenure).
2. Apply PCA to retain components explaining 85% of variance (e.g., first three components).
3. Run k-means on PCA scores with $K = 4$; silhouette score 0.48 indicates reasonable separation.
4. Compare against hierarchical clustering (Ward linkage) to validate grouping stability.
5. Profile clusters (high-value loyalists, promotion responders, at-risk, new users) for marketing strategy.

11.7 Interview Questions and Answers

1. **Question:** Why is feature scaling important before PCA or k-means?

   Solution: Both rely on Euclidean geometry; unscaled features with large variances dominate principal components or cluster assignments, distorting structure.

2. **Question:** How do you choose the number of principal components to retain?

   Solution: Inspect the scree plot for an elbow, set a cumulative variance threshold (e.g., 90%), or validate with downstream performance (e.g., classification accuracy using reduced features).

3. **Question:** Compare k-means with hierarchical clustering.

   Solution: k-means scales well, assumes spherical clusters, and requires pre-specifying K. Hierarchical clustering handles varied shapes, produces dendrograms aiding interpretation, but is more computationally expensive.

4. **Question:** What challenges arise when clusters have different densities?

   Solution: k-means struggles; dense clusters dominate centroid placement. Alternatives include DBSCAN or Gaussian Mixture Models, which adapt to varying densities and covariance structures.

5. **Question:** How can you assess cluster quality without labels?

   Solution: Use internal metrics (silhouette, Calinski-Harabasz), stability analysis (re-run clustering on bootstrap samples), or evaluate business outcomes (e.g., uplift in targeted campaigns).

12 Conclusion

These notes now cover a broad spectrum of data science fundamentals: linear and logistic regression, discriminant analysis, tree-based models, support vector machines, resampling and regularization, and unsupervised techniques such as PCA and clustering. For interviews, practice articulating model assumptions, diagnostic workflows, and trade-offs across methods—most questions hinge on interpreting results and choosing the right tool for a scenario. Reinforce the material by:

- Reworking small datasets in the companion notebooks (‘data-science/Code/‘) to solidify intuition.
- Creating summary sheets of formulas, metrics, and common interview talking points.
- Building or critiquing end-to-end pipelines that combine feature engineering, model selection, and validation.

Regular review of these topics, plus hands-on experimentation, will keep the concepts fresh and help you communicate confidently during interviews.