Introduction to Data Science

Multiple-Choice Questions (A–D)

Select the best answer. Correct answers are marked in **bold**.

1. Which scenario is the MOST likely indicator of data leakage?

(A) High training error but low test error

(B) High variance across CV folds

(C) **Validation performance extremely high but real-world performance collapses**

(D) Low correlation between features

Explanation: (C) is correct because leakage often produces unrealistically high validation scores that do not generalize; the model has seen information it should not have (e.g. from the future or from the test set). (A) suggests underfitting or optimization issues, not leakage. (B) high variance across folds usually indicates instability or small sample sizes, not specifically leakage. (D) low correlation between features is a modeling characteristic, not a symptom of leakage.

2. Why can accuracy be higher for a worse classifier on imbalanced data?

(A) Accuracy inversely correlates with model complexity

(B) **Majority-class predictions dominate the metric**

(C) Accuracy penalizes false negatives heavily

(D) Accuracy is threshold-independent

Explanation: (B) is correct because, on imbalanced data, a model that always predicts the majority class can achieve high accuracy while completely failing on the minority class. (A) is false: accuracy does not systematically depend on model complexity. (C) is wrong: accuracy weights all errors equally, not false negatives more. (D) is incorrect: accuracy *does* depend on the chosen classification threshold when probabilities are used.

3. Which CV strategy prevents leakage when multiple rows belong to the same entity?

(A) Standard K-Fold

(B) Stratified K-Fold

(C) **Group K-Fold**

(D) TimeSeriesSplit

Explanation: (C) is correct because Group K-Fold keeps all observations from the same entity (e.g. customer, patient) in a single fold, preventing identity-based leakage between train and validation. (A) standard K-Fold randomly splits individual rows, so the same entity can appear in both train and validation. (B) stratified K-Fold balances label proportions but still splits entities across folds. (D) TimeSeriesSplit respects time order but does not guarantee grouping by entity.

4. A model performs extremely well on the training data but much worse on new, unseen data. What does this usually mean?

(A) The model is too simple

(B) **The model memorized the training data but did not learn general patterns**

(C) The training data is too small to use

(D) The test data must contain errors

Explanation: (B) is correct: this describes overfitting—the model remembers examples instead of learning patterns. (A) a simple model tends to perform poorly on both train and test. (C) small data can contribute but is not the main explanation. (D) poor test performance is rarely caused by test-set errors.

5. Which situation MOST indicates high model variance?

(A) Training and validation errors both high

(B) **Training error low, validation error high**

(C) Both errors low

(D) Validation error lower than training error

Explanation: (B) is correct because low training error and high validation error are characteristic of overfitting / high variance: the model memorizes training data but does not generalize. (A) high errors on both sets indicate underfitting/high bias. (C) low errors on both sets indicate a good fit. (D) validation error lower than training error can happen due to regularization or sampling noise, but does not by itself indicate high variance.

6. Lasso tends to outperform Ridge when:

(A) Predictors are all highly correlated

(B) **Many features are irrelevant and sparsity is desired**

(C) All features have equal marginal contributions

(D) The dataset is extremely small

Explanation: (B) is correct because Lasso (L1) shrinks some coefficients exactly to zero, performing feature selection and yielding sparse models when many predictors are irrelevant. (A) with strong multicollinearity, Ridge (L2) is often preferable. (C) if all features truly contribute similarly, neither sparsity nor feature selection is particularly advantageous. (D) dataset size alone does not make Lasso strictly better than Ridge; both can overfit when data are extremely limited.

7. What is the main difference of Gradient Boosting over Random Forests?

(A) Lower risk of overfitting

(B) Models are trained independently

(C) **Sequential learning focuses on correcting previous errors**

(D) It requires no hyperparameter tuning

Explanation: (C) is correct: Gradient Boosting trains trees sequentially, each new tree focusing on the residual errors of the current ensemble, which can yield very strong performance. (A) is false; boosting actually can overfit more easily than Random Forests and often needs regularization. (B) describes Random Forests (bagging), not boosting. (D) is wrong: Gradient Boosting typically requires substantial hyperparameter tuning (learning rate, depth, number of estimators, etc.).

8. Which preprocessing step MOST commonly introduces leakage?

(A) Scaling features using training-set statistics

(B) **Scaling before splitting the dataset**

(C) Dropping missing values

(D) One-hot encoding

Explanation: (B) is correct because scaling before splitting uses information from the entire dataset (including validation/test) to compute statistics (mean, variance), leaking future information into training. (A) is actually good practice, as long as statistics are computed only on the training set. (C) dropping missing values is not inherently leakage; it depends on when and how it is done. (D) one-hot encoding is safe if applied within the train-only pipeline or consistently across splits.

9. A model with perfect training RMSE and poor test RMSE MOST likely suffers from:

(A) High bias

(B) **High variance**

(C) Underfitting

(D) Class imbalance

Explanation: (B) is correct: near-zero training error but high test error is classic overfitting, which corresponds to high variance. (A) and (C) indicate underfitting, which would show high error on both training and test sets. (D) class imbalance is mainly an issue for classification metrics, not RMSE in regression.

10. Why do we use cross-validation when training a model?

(A) To guarantee the model never overfits

(B) To make the model train faster

(C) **To check how well the model generalizes to unseen data**

(D) To automatically fix missing values

Explanation: (C) is correct: cross-validation tests the model on multiple splits of the data, giving a better idea of how well it will perform on new, unseen examples. (A) CV cannot guarantee no overfitting. (B) CV usually makes training slower, not faster. (D) CV does not handle missing values; that's a separate preprocessing step.

11. Which example BEST represents target leakage?

(A) Including a lagged target variable

(B) Using imputation on training folds only

(C) **Including "amount overdue" to predict loan default**

(D) Using regularization to penalize coefficients

Explanation: (C) is correct because "amount overdue" is typically observed only after default behavior has occurred, so it contains information from the future relative to the prediction. (A) including lagged targets can be valid if the lag is strictly in the past and respects temporal order. (B) using imputation on training folds only is the correct, leakage-safe approach. (D) regularization affects model complexity, not data flow, and is unrelated to leakage.

12. What makes PCA an example of *unsupervised* learning?

(A) It uses the target variable to find the best components

(B) It learns rules from labeled training data

(C) **It finds structure in the features without using any target labels**

(D) It automatically improves model accuracy

Explanation: (C) is correct: PCA looks only at the input features and finds directions of maximum variation, without using the target variable. (A) is false because using the target variable would make the method supervised. (B) describes supervised learning, not PCA. (D) PCA may help but does not guarantee accuracy improvements.

13. Why is it important to apply SMOTE *after* the train/test split?

(A) SMOTE automatically improves model accuracy

(B) It removes the need for cross-validation

(C) **To ensure that synthetic examples are created only from the training data**

(D) To reduce the number of features

Explanation: (C) is correct: SMOTE creates new minority-class examples by combining existing ones. If it is applied before the train/test split, the test set may contain synthetic points created using information from the training data, which leads to data leakage. (A) SMOTE does not guarantee accuracy improvements. (B) SMOTE does not replace cross-validation. (D) SMOTE affects sample size, not the number of features.

14. What is a key advantage of using a Random Forest compared to a single decision tree?

(A) It always produces perfect accuracy

(B) It requires no training data

(C) **It combines many trees to make more stable and reliable predictions**

(D) It only works for classification, not regression

Explanation: (C) is correct: a Random Forest averages the results of many decision trees, which reduces the risk of relying on the mistakes of any single tree and leads to more stable predictions. (A) is false—no model guarantees perfect accuracy. (B) is impossible; every model needs training data. (D) is incorrect—Random Forests work for both classification and regression tasks.

15. Which method MOST reliably detects nonlinear feature interactions?

(A) Linear Regression

(B) L1 regularization

(C) **Random Forest / Boosted Trees**

(D) K-Means

Explanation: (C) is correct: tree-based ensemble methods naturally capture nonlinearities and interactions between features via hierarchical splits. (A) Linear Regression cannot capture interactions unless they are manually encoded. (B) L1 regularization affects sparsity but does not, by itself, model nonlinearity. (D) K-Means is an unsupervised clustering method and is not intended for modeling feature interactions with a target.

16. Why is R^2 sometimes negative on the test set?

(A) Model predictions are better than the baseline mean

(B) Test labels contain noise

(C) **The model performs worse than predicting the mean**

(D) R^2 cannot handle linear models

Explanation: (C) is correct: by definition, R^2 compares the model to a baseline that always predicts the mean; if the model's SSE exceeds the baseline's, R^2 becomes negative. (A) would give a positive R^2. (B) noise alone does not guarantee negative R^2; it depends on performance. (D) is false: R^2 is standard for linear models.

17. Which metric is MOST sensitive to a few very large errors in regression?

(A) MAE

(B) **RMSE**

(C) Accuracy

(D) R^2

Explanation: (B) is correct: RMSE squares error terms, so very large errors have a disproportionately large impact. (A) MAE is linear in the error and therefore less sensitive to outliers. (C) accuracy is used for classification, not regression. (D) R^2 is related to SSE but is a normalized measure of explained variance, not as directly interpretable as sensitivity to individual large errors.

18. Which of the following is LEAST appropriate for time series forecasting?

(A) TimeSeriesSplit

(B) Expanding window CV

(C) **Random train/test shuffling**

(D) Train on past $\rightarrow$ validate on future

Explanation: (C) is correct: random shuffling breaks temporal order and allows future information to leak into the training set, which is invalid for forecasting tasks. (A) and (B) both maintain chronological order in different CV schemes and are appropriate. (D) explicitly respects the temporal direction (past to future), which is the right principle for forecasting.

19. Why should any preprocessing step that learns from the data (e.g., scaling or imputation) be done inside cross-validation?

(A) It makes cross-validation run faster

(B) It guarantees perfect accuracy

(C) **To avoid using information from the validation set during training**

(D) It automatically fixes class imbalance

Explanation: (C) is correct: preprocessing steps like scaling or imputing missing values must be fit only on the training fold. If they use the entire dataset, then information from the validation fold leaks into the training process, giving an overly optimistic score. (A) CV usually becomes slower, not faster. (B) No method guarantees perfect accuracy. (D) These preprocessing steps do not handle class imbalance.

20. A model with low training error, low validation error, but wrong predictions on specific subgroups indicates:

(A) Overfitting

(B) Underfitting

(C) **Fairness/representation bias**

(D) Poor calibration

Explanation: (C) is correct: good overall performance but systematically worse performance on specific groups (e.g. demographic subgroups) points to fairness or representation bias. (A) overfitting would usually show a train–validation performance gap. (B) underfitting would produce high errors everywhere. (D) poor calibration means predicted probabilities do not match observed frequencies, but here the issue is uneven performance across subgroups.