Question 1

Which distribution has the best balance between bias and variance?

The Figure C has the best balance. It's moderately complex, balancing some bias to avoid oversimplification and some variance to avoid overfitting.

Sparse data limits how much complexity you can trust — too simple, like A, and you're underfitting; too flexible, like B, and you're overfitting noise.

C's moderate approach likely generalizes best, which I'd confirm with cross-validation in practice. It's about finding that trade-off where bias and variance minimize total error.

Question 2

What is the purpose of this graph and its name?

This is a Receiver Operating Characteristic curve, or ROC curve. Its purpose in model selection and evaluation is to show how well a binary classification model distinguishes between classes.

It plots the trade-off between catching true positives and avoiding false positives across different decision thresholds, helping us assess model performance holistically.

What kind of model result does the dashed line represent?

It's a random guess model. The baseline where the true positive rate equals the false positive rate.

Which curve represents a better fit, the red or the green? Why?

The green curve's path suggests it's correctly distinguishing classes, likely with a solid AUC above 0.5.

The red curve, despite a bigger AUC, being below the line implies it's worse than random—its AUC might be high in absolute terms, but if it's below 0.5 relative to the random baseline, it's not a practical fit.

A model under the line could mean the labels are flipped or the scoring's off, but either way, green outperforms in a real-world sense.

Question 3

Can we say that the model has a good performance in the test evaluation?

The confusion matrix shows correct predictions at 50% for A, 45% for B, and 50% for C, averaging around 48% accuracy if classes are balanced.

That beats random guessing (33% for 3 classes), but it's far from good—70-80% or higher is the benchmark for solid test performance.

Graph A, training accuracy, rises in a curve and flattens near 1—say, 95-100%—while Graph B, training loss, crashes fast then drifts toward (infinity, 0), likely nearing 0.

This screams success on training data, but the test matrix tells a different story. A 48% test accuracy against near-perfect training isn't 'good' — it's a red flag for poor generalization.

What phenomenon happened during the test evaluation?

The gap between training and test is glaring. The model's over-specialized, memorizing training patterns instead of learning robust features.

The B-C mix-up could hint at feature overlap or imbalance too, but the training-test chasm is the dominant signal here. This is textbook overfitting.

The curves suggest it kept optimizing past the point of usefulness—after that initial loss crash, it should've stopped.

I'd recommend early stopping around the loss inflection point or adding regularization like dropout to tame it.

The B-C confusion might warrant a feature check too, but overfitting's the core issue. Test performance isn't good—it's a wake-up call to recalibrate.