XGBoost Interview Questions Cheat Sheet

Basic Level

Q: What is XGBoost? How is it different from Gradient Boosting?

A: XGBoost (Extreme Gradient Boosting) is an optimized version of gradient boosting. Differences include built-in regularization, parallelization, better handling of missing values, and more speed.

Q: What are the advantages of XGBoost?

A: High performance, regularization, missing value handling, cross-validation, parallel computing, and multiple objective functions.

Q: How does XGBoost handle missing values?

A: XGBoost automatically learns the best direction to split when missing values are encountered.

Q: What are the key hyperparameters in XGBoost?

A: eta, max_depth, subsample, colsample_bytree, n_estimators, gamma, lambda, alpha.

Q: What types of objectives does XGBoost support?

A: Regression, classification, ranking, and custom objectives.

Intermediate Level

Q: Explain how XGBoost works internally.

A: XGBoost builds trees sequentially using second-order Taylor approximation (gradients and Hessians) for better optimization.

Q: What is the role of the eta parameter?

A: Controls learning rate. Smaller values slow learning but improve generalization.

Q: Whats the effect of max_depth?

A: Controls complexity. Higher values can lead to overfitting; lower values might underfit.

Q: Whats the difference between gamma, lambda, and alpha?

A: gamma: min loss reduction for split; lambda: L2 regularization; alpha: L1 regularization.

XGBoost Interview Questions Cheat Sheet

Q: How does XGBoost perform regularization?

A: Adds L1 (alpha) and L2 (lambda) terms to the objective to prevent overfitting.

Advanced Level

Q: How does XGBoost handle overfitting?

A: Through regularization, subsampling, tree constraints, early stopping, and tuning eta.

Q: What is the difference between GOSS and XGBoosts tree-growing strategy?

A: XGBoost uses level-wise growth; GOSS (LightGBM) uses leaf-wise growth, which can be faster but risk overfitting.

Q: What is scale_pos_weight and when should you use it?

A: Used for imbalanced classes. Set as: num_negative / num_positive to balance class weights.