**Data Science Interview Questions - Machine Learning**

*Source: Interview Preparation*

Common Machine Learning Interview Questions:  
   
 1. What is the difference between supervised and unsupervised learning?  
 Supervised learning uses labeled training data to learn a mapping from inputs to outputs.   
 Unsupervised learning finds hidden patterns in unlabeled data.  
   
 2. Explain overfitting and how to prevent it.  
 Overfitting occurs when a model learns the training data too well, including noise.   
 Prevention methods include: cross-validation, regularization, more training data, feature selection.  
   
 3. What is cross-validation and why is it important?  
 Cross-validation splits data into multiple folds to evaluate model performance more robustly.   
 It helps prevent overfitting and provides better performance estimates.  
   
 4. Explain the bias-variance tradeoff.  
 Bias is the model's assumptions about the data. Variance is sensitivity to training data changes.   
 High bias = underfitting, high variance = overfitting. The goal is to balance both.  
   
 5. What are the main evaluation metrics for classification?  
 Accuracy, precision, recall, F1-score, ROC-AUC, confusion matrix.  
   
 6. How do you handle imbalanced datasets?  
 Techniques include: resampling (oversampling/undersampling), SMOTE, class weights,   
 different evaluation metrics, ensemble methods.  
   
 7. What is feature engineering and why is it important?  
 Feature engineering creates new features from raw data to improve model performance.   
 It can include: scaling, encoding categorical variables, creating interaction terms.  
   
 8. Explain regularization techniques.  
 L1 (Lasso) and L2 (Ridge) regularization add penalty terms to prevent overfitting.   
 L1 can zero out coefficients (feature selection), L2 shrinks coefficients.  
   
 9. What is the difference between bagging and boosting?  
 Bagging (Bootstrap Aggregating) trains multiple models independently and averages predictions.   
 Boosting trains models sequentially, each focusing on previous errors.  
   
 10. How do you choose the right algorithm for a problem?  
 Consider: data size, data type, problem type (classification/regression),   
 interpretability needs, computational constraints, performance requirements.