Machine Learning Engineer Interview Q&A

1. Core ML Knowledge

Q: What's the difference between supervised and unsupervised learning?

A: Supervised learning uses labeled data to train models for tasks like classification or regression. Unsupervised learning finds patterns in unlabeled data, such as clustering or dimensionality reduction.

Q: What metrics do you use to evaluate classification models?

A: Common metrics include Accuracy, Precision, Recall, F1-Score, and AUC-ROC. Cross-validation is also used to ensure generalization.

2. Python, scikit-learn, XGBoost, pandas, NumPy

Q: How would you handle missing values in a dataset?

A: Drop rows/columns, impute using mean/median/mode, or create a missing flag feature.

Q: How would you use XGBoost in a pipeline?

A: Integrate with sklearn.pipeline using preprocessing and XGBClassifier/XGBRegressor. Tune hyperparameters with GridSearchCV.

Q: How do you optimize performance with pandas and NumPy for large datasets?

A: Use vectorized operations, avoid apply where possible, and use efficient slicing or query methods.

3. ML Workflow (Exploration Modeling Validation)

Q: How do you perform feature engineering?

A: Use domain knowledge, interaction terms, transformations, time-series features, and encoding methods.

Q: How do you ensure your model generalizes well?

A: Use cross-validation, monitor training/validation loss, and apply regularization or early stopping.

Q: What's your approach to model interpretability?

A: Use SHAP, LIME, feature importance, and communicate with visualizations for non-technical audiences.

Machine Learning Engineer Interview Q&A

4. Deployment, MLOps, and Production

Q: How have you deployed ML models into production?

A: Wrap models in Flask/FastAPI, serialize with joblib, and use CI/CD pipelines and Docker.

Q: What is MLflow and how have you used it?

A: MLflow is used for experiment tracking and model management. I log parameters, metrics, and models.

Q: What would you monitor in production?

A: Monitor model drift, latency, prediction confidence, and error rates with dashboards and alerts.

5. Collaboration and Documentation

Q: How do you collaborate with backend/data engineers?

A: Define data contracts, use version control, align APIs, and document clearly.

Q: How do you explain a model to a non-technical stakeholder?

A: Use simple language, analogies, visualizations, and focus on business impact.