Machine Learning Engineer Interview Q&A

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 (e.g., predicting customer churn). Unsupervised learning finds patterns in unlabeled data (e.g., clustering customers by behavior).

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

A: Common metrics include:

- Accuracy: Overall correctness.

- Precision/Recall/F1: For imbalanced datasets.

- AUC-ROC: For probability-based classification.

I also use cross-validation to ensure generalizability.

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

A: Depends on the data:

- Drop rows/columns with excessive missing values.

- Impute using mean/median (numeric), mode (categorical), or predictive models.

- Flag missing values as a feature.

Q: How would you use XGBoost in a pipeline?

A: I use XGBClassifier or XGBRegressor in a sklearn.pipeline, along with preprocessing steps like ColumnTransformer. I tune hyperparameters using GridSearchCV.

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

A: - Use vectorized operations.

- Avoid apply where possible.

- Use df.query() or df.loc[] efficiently.

- For larger data, use Dask or convert to NumPy arrays.

Q: How do you perform feature engineering?

A: - Domain knowledge to create meaningful features.

- Interaction terms, log transformations, buckets/bins.

- Time-series lags/rolling means.

- Encoding categorical features using OneHot or Target Encoding.

Q: How do you ensure your model generalizes well?

A: - Use stratified K-fold cross-validation.

- Monitor for overfitting via training/validation curves.

- Use early stopping with tree-based models.

Q: What's your approach to model interpretability?

A: - For tree models: use feature importance, SHAP, or LIME.

- For linear models: coefficients explain direction/strength.

- Communicate insights using visualizations and examples.

Q: How have you deployed ML models into production?

A: - Wrapped models in Flask APIs or FastAPI.

- Serialized models using joblib or pickle.

- Integrated with backend systems via REST APIs or batch jobs.

- Used Docker and CI/CD pipelines for versioned deployments.

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

A: MLflow helps track experiments, log models and metrics.

I've used:

- mlflow.log_param, log_metric during experiments.

- mlflow.sklearn.log_model() to save model versions.

- mlflow ui to visualize runs.

Q: What would you monitor in production?

A: - Model drift (input/output distribution changes)

- Latency

- Prediction confidence

- Error rates

- Use alerts and dashboards to proactively flag issues.

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

A: - Align on data contracts and schema.

- Define API interfaces for models.

- Use version control (Git) and task boards (JIRA).

- Document assumptions, model choices, and known limitations.

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

A: - Use analogies or examples.

- Visualize inputs vs outputs (e.g., SHAP plots).

- Emphasize business impact over technical details.

- Avoid jargon; simplify explanations.