**🧠 GENERAL CONCEPTS**

**1. What is Machine Learning?**

Machine Learning (ML) is a branch of Artificial Intelligence that enables systems to **learn from data** and improve their performance **without explicit programming**.

**2. What are the main types of ML?**

* **Supervised Learning** — learn from labeled data (e.g., regression, classification).
* **Unsupervised Learning** — find patterns in unlabeled data (e.g., clustering).
* **Semi-Supervised Learning** — mix of labeled and unlabeled data.
* **Reinforcement Learning** — learn by reward/punishment (e.g., games, robotics).

**3. What is the difference between AI, ML, and Deep Learning?**

* **AI** → any system that acts intelligently.
* **ML** → subset of AI that learns from data.
* **Deep Learning** → subset of ML using neural networks with many layers.

**4. What is a feature and a label?**

* **Feature** = input variable (X).
* **Label** = output variable (Y) that you want to predict.

**📊 DATA PREPARATION**

**5. Why is data preprocessing important?**

Because real-world data is messy — it may have **missing values**, **outliers**, or **inconsistent formats**. Cleaning ensures your model learns the real patterns, not the noise.

**6. What is feature scaling and why is it needed?**

Scaling adjusts numeric values to a common range (e.g., 0–1).  
It improves **training stability and convergence**, especially for algorithms like KNN, SVM, or Gradient Descent–based models.

**7. What is encoding?**

Converting **categorical data** into numeric form, e.g.:

* **Label Encoding:** Red → 0, Blue → 1
* **One-Hot Encoding:** [1,0,0], [0,1,0], [0,0,1]

**8. What is feature engineering?**

Creating or transforming variables to better represent underlying data patterns (e.g., from “date” derive “day of week,” “month,” etc.).

**🤖 MODELS & ALGORITHMS**

**9. What are the most common ML algorithms?**

* **Regression:** Linear, Ridge, Lasso, Decision Tree
* **Classification:** Logistic, KNN, SVM, Random Forest, Naive Bayes
* **Clustering:** K-Means, DBSCAN, Hierarchical
* **Dimensionality Reduction:** PCA, t-SNE
* **Ensemble:** Random Forest, XGBoost, AdaBoost

**10. What’s the difference between regression and classification?**

* **Regression** → predicts continuous values (e.g., house price).
* **Classification** → predicts categories (e.g., spam or not spam).

**11. What is overfitting vs underfitting?**

* **Overfitting:** model learns training data too well, performs poorly on new data.
* **Underfitting:** model is too simple, fails to capture data patterns.

**12. How to prevent overfitting?**

* Use **regularization (L1/L2)**
* Apply **cross-validation**
* Add **more data** or **drop unnecessary features**

**⚙️ TRAINING & TESTING**

**13. Why split data into train and test sets?**

To evaluate model performance on **unseen data**, ensuring generalization.

**14. What is cross-validation?**

Splitting data into multiple folds, training on some and validating on others, to reduce bias in model performance.

**15. What is a confusion matrix?**

A table showing correct vs incorrect predictions for classification tasks.

**16. What are hyperparameters?**

Settings that control how the model learns (e.g., learning rate, tree depth, number of estimators).

**17. What is model tuning?**

Optimizing hyperparameters using **GridSearchCV**, **RandomSearchCV**, or **Bayesian optimization** to improve performance.

**📈 EVALUATION**

**18. What metrics are used to evaluate ML models?**

| **Type** | **Common Metrics** |
| --- | --- |
| Regression | MAE, MSE, RMSE, R² |
| Classification | Accuracy, Precision, Recall, F1, ROC-AUC |
| Clustering | Silhouette Score, Davies–Bouldin Index |

**19. What is R² (R-squared)?**

It measures how much of the variance in target variable is explained by the model. Closer to 1 = better.

**20. What is bias–variance tradeoff?**

Balancing **simplicity (bias)** and **complexity (variance)**:

* High bias → underfitting.
* High variance → overfitting.  
  You need a balance.

**🧩 MODEL IMPROVEMENT**

**21. How can model performance be improved?**

* Clean & scale data properly.
* Add more training data.
* Use feature engineering.
* Tune hyperparameters.
* Use ensemble methods (bagging, boosting).

**22. What is ensemble learning?**

Combining multiple models to improve accuracy.  
Examples: Random Forest, Gradient Boosting, AdaBoost.

**23. What is regularization?**

Adding penalty terms to avoid large coefficients:

* **L1 (Lasso)** → removes irrelevant features.
* **L2 (Ridge)** → reduces large weights smoothly.

**🧮 DEPLOYMENT & MONITORING**

**24. How is a model deployed?**

You can deploy as:

* **API service** (Flask, FastAPI, ASP.NET, Azure Function)
* **Embedded model** (e.g., ML.NET .zip file)
* **Cloud deployment** (Azure ML, AWS SageMaker, GCP AI Platform)

**25. What is MLOps?**

“MLOps” = DevOps for ML — practices for automating training, testing, deployment, and monitoring of ML models.

**26. Why retrain models?**

Because data drifts — patterns change over time (e.g., customer behavior, market prices). Retraining keeps model updated.

**💬 PRACTICAL QUESTIONS**

**27. How is ML different from traditional programming?**

In traditional programming, you code rules manually.  
In ML, you provide **data + output**, and the **algorithm finds the rules**.

**28. What are some real-world ML applications?**

* Predictive maintenance in factories
* Spam detection in Gmail
* Product recommendation (Amazon, Netflix)
* Face recognition
* Credit risk prediction
* Speech-to-text and translation

**29. How much data is needed for ML?**

It depends — but generally, **more data → better accuracy**, as long as data is clean and diverse.

**30. What’s the difference between training, validation, and testing data?**

* **Training** — model learns patterns.
* **Validation** — tune hyperparameters.
* **Testing** — final unbiased evaluation.

**🧱 ADVANCED TOPICS**

**31. What is Deep Learning?**

A subset of ML using **neural networks** with many layers to automatically learn features (used in image, speech, text).

**32. What is Transfer Learning?**

Using a **pre-trained model** and fine-tuning it on your dataset (e.g., use ResNet trained on ImageNet for your custom image set).

**33. What is Reinforcement Learning?**

Learning by interaction — agent learns by **reward and penalty** (e.g., AlphaGo, self-driving cars).

**34. What is Gradient Descent?**

An optimization algorithm to minimize loss function by adjusting weights.

**35. What is the difference between Batch, Mini-batch, and Online Learning?**

* **Batch:** train on entire dataset at once.
* **Mini-batch:** train on small chunks (most common).
* **Online:** train one example at a time (real-time systems).