AIML Engineer Interview Prep (Q&A — Detailed Answers)

Conmove Private Limited — Entry-Level Role

---

1. Core Machine Learning

Q1. What is the difference between supervised, unsupervised, and reinforcement learning?

A:

- **Supervised Learning:**
  Uses labeled data. The model learns a mapping from input → output.
  Examples: spam detection, price prediction.
- **Unsupervised Learning:**
  Works with unlabeled data. The model finds hidden patterns or groupings.
  Examples: customer segmentation, anomaly detection.
- **Reinforcement Learning:**
  An agent learns actions by interacting with an environment and receiving rewards/penalties.
  Examples: robotics, route optimization, gaming.

Q2. Explain the steps of a typical ML workflow.

A:

1. **Problem Definition:** Understand what business problem ML can solve.
2. **Data Collection:** Databases, APIs, logs — quantity and quality matter.
3. **Preprocessing:** Missing values, duplicates, inconsistent formats.
4. **Feature Engineering:** Selecting/creating meaningful features.
5. **Model Selection:** Try baseline models first, then advanced ones.
6. **Training:** Optimize model parameters.
7. **Evaluation:** Use appropriate metrics — avoid accuracy for imbalanced data.

8. **Deployment:** Serving via FastAPI, Docker, or cloud services.
9. **Monitoring:** Check performance drift and retrain periodically.

Q3. What is overfitting? How do you prevent it?

A:
Overfitting occurs when a model fits training data too closely, learning noise instead of patterns.
It performs well on training data but poorly on unseen data.

Prevention techniques:

- Regularization (L1/L2)
- Dropout in neural networks
- Cross-validation
- Early stopping
- Data augmentation
- Using simpler models
- Providing more data

Q4. How do you choose the right evaluation metric?

A:
Depends on the problem and business need.

- **Imbalanced classification:** F1-score, Precision, Recall
- **Regression:** RMSE if large errors matter, MAE for general accuracy
- **Ranking applications:** NDCG, MAP
- **Real-time systems:** Latency + accuracy importance

Always tie the metric to business goals — not just model performance.

Q5. When would you prefer Precision or Recall over Accuracy?

A:
Accuracy is misleading with imbalance.

- **High Precision needed:** When false positives are bad (loan approval, fraud flagging).
- **High Recall needed:** When false negatives are bad (disease detection, missing delays in logistics).

---

2. Data Preprocessing & Feature Engineering

Q6. How do you handle missing values?

A:

- **Numeric:** Mean/median imputation, interpolation, or model-based imputation.
- **Categorical:** Mode or "Unknown" bucket.
- **Time-series:** Forward/backward fill.
- **Drop:** Only if missing % is low and doesn't introduce bias.

Q7. Why is feature scaling needed?

A:
Some algorithms rely on distance or gradient calculations. If features are on different scales, one feature may dominate.
Scaling methods:

- **Standardization (z-score)**
- **Min–Max scaling**

Models like Logistic Regression, SVM, KNN, Neural Networks benefit the most.

Q8. Difference between one-hot encoding and label encoding?

A:

- **One-hot encoding:**
  Turns categories into binary vectors. Best for unordered categories.
- **Label encoding:**
  Maps categories to integers — only suitable when order exists (e.g., small/medium/large).

Q9. How do you detect and treat outliers?

A:
Detection:

- IQR method
- Z-score
- Boxplots
- Isolation Forest

Handling:

- Remove (if they're errors)
- Cap values using winsorization
- Apply log/Box–Cox transformations
- Investigate domain-specific reasons for outliers

Q10. What steps would you take to clean logistics data?

A:
Logistics data is often messy due to manual entries or real-time sensors. Cleanup steps:

- Standardize date formats (ISO 8601).
- Normalize location names and port codes.
- Remove duplicated timestamps/events.
- Align event sequences (gate-in → loading → transit → unloading).
- Fix inconsistent units (hours, days).
- Validate container codes using ISO 6346 rules.

---

3. Deep Learning

Q11. Explain CNNs, RNNs, and Transformers.

A:

- **CNNs:**
  Capture spatial hierarchies — great for images. Use filters and pooling.
- **RNNs:**
  Process sequential data. Maintain hidden states. LSTM/GRU solve long-term memory issues.
- **Transformers:**
  Replace recurrence with self-attention, enabling parallel processing. Good for NLP, CV, and multimodal tasks.

Q12. What is backpropagation?

A:
An algorithm used to compute gradients of the loss function w.r.t model weights. These gradients are used to update weights during training with optimizers like Adam or SGD.

Q13. Why are activation functions needed?

A:
Without activation functions, neural networks behave like linear models. Activation functions (ReLU, sigmoid, tanh) introduce non-linearity that allows learning complex relationships.

Q14. Explain epochs, batch size, and learning rate.

A:

- **Epoch:** One full pass through the training set.
- **Batch size:** Number of samples processed before one weight update.
- **Learning rate:** Controls how big each update step is.

A good learning rate avoids overshooting and slow convergence.

Q15. What is the vanishing/exploding gradient problem? Solutions?

A:
When gradients become too small or too large during backpropagation, training becomes unstable.

Solutions:

- ReLU activation
- Batch normalization
- Gradient clipping
- LSTM/GRU cells
- Residual connections (ResNet)

---

4. Natural Language Processing

Q16. What is tokenization?

A:
Splitting text into smaller units (words, subwords, characters). Modern models prefer subword tokenization (BPE, WordPiece) for vocabulary efficiency.

Q17. What are word embeddings?

A:
Numerical vector representations of words that capture meaning and context. Examples: Word2Vec, GloVe, FastText, BERT embeddings.

Q18. Difference between BERT and GPT models?

A:

- **BERT:** Bidirectional encoder. Good for classification, extraction tasks.
- **GPT:** Autoregressive decoder. Designed for text generation. Transformers use attention to capture long-range relationships.

Q19. How do you perform text classification?

A:

1. Clean text (remove noise)
2. Tokenize
3. Convert tokens to embeddings
4. Feed to classifier (BERT, LSTM, CNN)
5. Evaluate using F1-score

Q20. How to extract structured info from shipping documents?

A:

- OCR to extract raw text
- Named Entity Recognition for fields like container numbers
- Regex patterns for codes (e.g., container number format AB12 345678)
- Post-processing to fix OCR errors
- Use domain knowledge (port codes, vessel names)

---

5. Computer Vision

Q21. How do convolutions work?

A:
A convolutional filter slides over an image computing dot products.
Each filter detects a particular feature (edges, textures).
Stacking layers creates a hierarchy from low-level to high-level features.

Q22. What is transfer learning?

A:
Using a pre-trained model's learned features for a new task.
Benefits: faster training, less data needed, higher accuracy.

Q23. Difference between image classification and object detection?

A:

- **Classification:** Whole image → single label
- **Detection:** Identify *what* and *where* using bounding boxes (YOLO, Faster R-CNN)

Q24. How would you build a model to read container numbers?

A:

1. Detect container region (object detection).
2. Extract text via OCR (Tesseract/CRNN).
3. Validate using ISO container code rules (pattern: XXXX1234567).
4. Correct common OCR mistakes (O→0, I→1).

---

6. Statistics & Mathematics

Q25. Explain the bias-variance tradeoff.

A:

- **High bias:** Model too simple → underfitting
- **High variance:** Model too complex → overfitting
  Goal is to find a middle ground through regularization and proper model complexity.

Q26. What is a probability distribution?

A:
A function describing the likelihood of different outcomes.
Examples: Normal, Bernoulli, Poisson, Uniform.

Q27. What is hypothesis testing?

A:
Technique to determine if data supports a hypothesis.
Steps: set null hypothesis → choose significance level → compute p-value → accept/reject.

Q28. When is a t-test used?

A:
Comparing means of two samples when population variance is unknown or sample size is small.

Q29. What is PCA and when would you use it?

A:
Dimensionality reduction that projects data onto orthogonal components capturing max variance.
Useful when:

- Data has many correlated features
- Reducing noise
- Speeding up model training

---

7. Python, Debugging & Coding

Q30. Write a function to remove duplicates from a list.

A:

```
def remove_duplicates(lst):
   return list(set(lst))
```

Q31. Why does training loss decrease but validation loss increase?

A:
Model is overfitting — memorizing training data instead of generalizing.

Q32. How do you organize code for an ML project?

A:
A clean structure:

- data/ – raw & processed files
- src/preprocessing.py
- src/models.py
- notebooks/ – experimentation
- models/ – saved weights
- app/ – FastAPI for deployment

Q33. What libraries do you use for different tasks?

A:

- **Data:** Pandas, NumPy
- **ML:** Scikit-learn
- **DL:** PyTorch, TensorFlow
- **Visualization:** Matplotlib, Plotly
- **Deployment:** FastAPI, Docker

8. Tools & Deployment

Q34. What is a preprocessing pipeline?

A:
A chain of transformations (scaling, encoding, cleaning) applied consistently during training and prediction.
Ensures no mismatch between training and deployment preprocessing.

Q35. How do you package a model using joblib/pickle?

A:

```
import joblib

joblib.dump(model, "model.pkl")
```

Later load it in API:

```
model = joblib.load("model.pkl")
```

Q36. How would you deploy a model using FastAPI?

A:

1. Create FastAPI app
2. Load model on startup
3. Create /predict endpoint
4. Accept JSON input
5. Return prediction
6. Containerize using Docker
7. Deploy on AWS/GCP

Q37. Why is Docker useful in ML?

A:

- Ensures consistent environment across machines
- Eliminates dependency/version issues
- Simplifies deployment and scaling

Q38. What cloud tools help with ML deployment?

A:
AWS: EC2, Lambda, S3, SageMaker
GCP: Cloud Storage, Vertex AI, Compute Engine

---

9. Logistics Domain Knowledge

Q39. What ML opportunities exist in container logistics?

A:

- Route optimization
- ETA prediction

- Delay detection
- OCR from shipping documents
- Container tracking analysis
- Anomaly detection in movement patterns

Q40. How would you predict delays in logistics?

A:
Use features such as:

- Historical transit times
- Weather conditions
- Traffic patterns
- Vessel schedules
- Loading/unloading delays

Models: Gradient Boosting, Random Forests, LSTMs (time series).

Q41. How can ML improve route planning?

A:

- Predict congestion
- Optimize delivery routes
- Use RL to dynamically select best paths
- Use clustering for demand grouping

Q42. What data sources are common in logistics?

A:

- GPS logs (latitude/longitude/time)
- Port event timestamps
- Container movement status

- Weather/traffic APIs
- Scanned documents (invoices, bills of lading)

Q43. How do you manage noisy/incomplete logistics data?

A:

- Impute missing events
- Standardize timestamps
- Remove duplicates
- Smooth GPS trajectories
- Cross-check event sequences
- Validate using business rules

---

10. Behavioral & Team Fit

Q44. Describe an end-to-end ML project you've worked on.

A:
Provide:

- Problem statement
- Data source & size
- Preprocessing steps
- Model selection & training
- Evaluation metrics
- Deployment method
- Final outcome
- Challenges + improvements

Q45. What challenges have you faced with a dataset?

A:
Mention examples like noise, imbalance, inconsistent formatting, missing values — and how you systematically fixed them.

Q46. How do you stay updated with ML trends?

A:
Follow research papers, tech blogs, newsletters, GitHub repos, Kaggle, and online ML communities.

Q47. Describe a time you learned a new tool quickly.

A:
Explain your approach:

1. Read documentation
2. Try a small example
3. Build a mini-project
4. Apply to the main project

Q48. How would you explain a complex model to non-technical people?

A:
Use analogies, focus on business impact (e.g., "This model predicts delays so operations can save time."). Avoid jargon.

---

11. Scenario-Based Questions

Q49. What steps do you take when a model is stuck at 70% accuracy?

A:

- Improve data quality

- Add better features
- Try other algorithms
- Tune hyperparameters
- Handle imbalance
- Add relevant domain-specific data
- Increase model complexity if underfitting

Q50. How would you handle unstructured logistics data?

A:

- Classify data types (text/images/logs)
- Apply respective cleanup pipelines
- Extract structured features
- Combine into a unified dataset
- Train multimodal or ensemble models

Q51. What if the model must go live in 1 week?

A:
Focus on essentials:

- Simple baseline model
- Clean essential data only
- Build minimal FastAPI service
- Dockerize
- Deploy
- Add basic monitoring
  Cut all non-critical tasks.

Q52. What would you check if a production model performs poorly?

A:

- Data drift
- Feature pipeline mismatch
- Incorrect API input formats
- Differences between training and live environment
- Model versioning errors
- Infrastructure latency issues
