Title: Complete Guide to Machine Learning for AI Engineers

Page 1: Introduction & Math Foundations

AI Engineers design intelligent systems that learn from data. Mastering Machine Learning (ML) is key for building scalable AI products.

Mathematics:

- Linear Algebra: Supports vector operations & neural networks (vectors, matrices, dot product).
- Calculus: Helps in optimizing models using gradients (derivatives, chain rule).
- Probability & Statistics: Essential for model prediction confidence (Bayes' theorem, distributions).
- Optimization: Guides learning process (Gradient Descent, Stochastic methods).

Page 2: Core Machine Learning Algorithms

Supervised Learning:

- Regression: Predict continuous values (Linear, Polynomial, Ridge).
- Classification: Categorize inputs (Logistic Regression, SVM, k-NN, Decision Trees, Random Forest).

Unsupervised Learning:

- Clustering: Group similar data (K-Means, DBSCAN, Hierarchical).
- Dimensionality Reduction: Simplify data (PCA, t-SNE, UMAP).

Model Evaluation:

- Metrics: Measure performance (Accuracy, Precision, Recall, F1, ROC-AUC).
- Cross Validation: Prevent overfitting by rotating test/train splits.

Page 3: Deep Learning & Architectures

Neural Networks:

- Perceptron & MLP: Basic building blocks with multiple layers.
- CNN: Image processing via filters and feature extraction.
- RNN/LSTM/GRU: Handle sequences and time-series data.
- Transformers: Parallel processing with self-attention (BERT, GPT).

Training:

- Loss Functions: Measure error (MSE, CrossEntropy).
- Optimizers: Update weights (SGD, Adam).
- Regularization: Avoid overfitting (Dropout, L2, Early Stopping).

Page 4: Tools, Libraries, and Data Handling

Python Libraries:

- NumPy/Pandas: Matrix operations and data manipulation.
- Matplotlib/Seaborn: Visualization of trends and patterns.
- Scikit-learn: Classical ML algorithms and utilities.

Deep Learning Frameworks:

- TensorFlow/Keras: High-level model design and training.
- PyTorch: Dynamic computation graphs for research and deployment.

Data Handling:

- Data Cleaning: Handle missing/outlier values.
- Feature Engineering: Create useful input variables.

Page 5: MLOps & Model Deployment

MLOps:

- Version Control: Track code and model changes (Git, DVC).
- Experiment Tracking: Monitor runs and metrics (MLflow, Weights & Biases).
- CI/CD: Automate testing and deployment (GitHub Actions, Jenkins).

Deployment:

- FastAPI: Lightweight Python framework to serve models via APIs.
- Docker: Containerize models for reproducibility and scalability.
- Kubernetes: Orchestrate multiple containers in production.
- Streamlit/Gradio: Rapidly build ML web apps for demos and testing.

Page 6: Advanced Topics & Resources

Natural Language Processing (NLP):

- Text processing: Tokenization, Lemmatization.
- Embeddings: Represent text numerically (Word2Vec, GloVe, BERT).
- Transformers: State-of-the-art in NLP (T5, GPT, LLaMA).

Reinforcement Learning:

- Agents learn via trial and error (Q-Learning, DQN, PPO).

Ethics & Safety:

- Bias & Fairness: Prevent discriminatory predictions.
- Explainability: Use SHAP/LIME to interpret model decisions.

Resources:

- arXiv, NeurIPS, ICLR, GitHub open-source repos (Hugging Face, LangChain).