



AI / ML Revision Notes

Machine Learning • Deep Learning • NLP • Generative AI
Comprehensive revision guide for students — theory, examples & diagrams


1. Machine Learning (ML)

1.1 What is Machine Learning?

Machine Learning is a branch of Artificial Intelligence where computers learn from data to make decisions or predictions — without being explicitly programmed with fixed rules. Instead of a programmer writing "if price > 500,000 then luxury home", the model discovers these patterns itself from thousands of examples.

Think of it this way: a child learns to recognise a cat not by memorising rules like "four legs, fur, whiskers" but by seeing hundreds of cats and non-cats. ML works the same way — feed it labelled data and it figures out the patterns on its own.


- **Training:** feeding labelled data to the model so it can learn patterns.
- **Model:** the mathematical function the algorithm builds from training data.
- **Prediction:** using the trained model on new, unseen data to get an output.
- **Features / Input:** the columns used to make a prediction (e.g. house size, location).
- **Label / Output:** the value we want to predict (e.g. house price, spam or not).

 **Key Point:** The three pillars of ML are: Data (examples to learn from), Algorithm (the learning method), and Model (the learned result).

1.2 Types of Machine Learning

There are three main types of ML, each suited to different kinds of problems:

- **Supervised Learning:** trains on labelled data (input + known output). Goal: predict output for new inputs. Examples: regression, classification.
- **Unsupervised Learning:** no labels — the algorithm finds hidden structure or groupings in data by itself. Example: clustering.
- **Reinforcement Learning:** an agent learns by trial and error, receiving rewards for good actions. Example: game-playing AI, robotics.

 **Remember:** Most real-world ML problems use Supervised Learning. Unsupervised Learning is great for exploratory analysis when you don't know the categories yet.

1.3 Regression — Predicting a Number

Regression predicts a continuous numerical value. The output can be any number on a scale — it is not limited to fixed categories. Use regression whenever you want to predict 'how much' or 'how many'.

Real-world analogy

Imagine you are a real estate agent. You look at a house's size, location, age, and condition, and then estimate its price. You're not choosing between categories — you're predicting a specific number. That's regression.

Example Dataset — House Price Prediction


INPUT Features (what we know): Size, Bedrooms, Age, Location, Condition.

OUTPUT / Target (what we predict): Price (\$) — a continuous number.

Size (sqft)	Bedrooms	Age (yrs)	Location	Condition	Price (\$)
850	2	10	Suburbs	Good	180,000
1200	3	5	City	Excellent	320,000
950	2	20	Suburbs	Average	160,000
1500	4	2	City	Excellent	450,000
700	1	30	Rural	Poor	90,000
1100	3	8	Suburbs	Good	240,000
1800	4	1	City	Excellent	520,000
600	1	25	Rural	Average	75,000

The model learns: larger houses in the city with excellent condition command higher prices. Given a new house (Size=1300, Bedrooms=3, Age=4, City, Good) it predicts ~\$295,000.

 **Key Point:** Regression algorithms: Linear Regression, Decision Tree Regressor, Random Forest, XGBoost, SVR.

 **Remember:** Key metric: RMSE (Root Mean Squared Error) — measures how far predictions are from actual values. Lower is better.

1.4 Classification — Predicting a Category

Classification predicts a category or class label — the output is one of a fixed set of options. Unlike regression (a number), classification answers 'which bucket does this belong to?' The categories are defined before training.

Real-world analogy

Think of a spam filter. It reads an email and decides: Spam or Not Spam? It doesn't predict a price — it chooses between predefined classes. That's classification.

Example Dataset — Email Spam Detection


INPUT Features: Sender, Has Link, Uses Capitals, Has Exclamation, Known Sender.

OUTPUT / Target (what we predict): Spam or Not Spam — a fixed category.

Email Sender	Has Link?	Caps?	Exclaim?	Known?	Label (Output)
unknown@xyz	Yes	Yes	Yes	No	Spam
boss@company	No	No	No	Yes	Not Spam
promo@deals	Yes	Yes	Yes	No	Spam
friend@gmail	No	No	No	Yes	Not Spam
win@lottery	Yes	Yes	Yes	No	Spam
hr@office	Yes	No	No	Yes	Not Spam
sale@unknown	Yes	Yes	No	No	Spam

The model learns: emails from unknown senders with links, excessive capitals, and exclamation marks are almost always spam. A new email matching those features gets classified as 'Spam' automatically.

 **Key Point:** Classification algorithms: Logistic Regression, Decision Tree, Random Forest, SVM, KNN, Naive Bayes.

 **Remember:** Key metrics: Accuracy, Precision, Recall, F1-Score. When classes are imbalanced (e.g. 95% Not Spam), accuracy alone is misleading — always check Precision and Recall too.

1.5 Clustering — Discovering Groups (Unsupervised)

Clustering groups data points into clusters based on similarity — with no predefined labels. The algorithm discovers the natural groupings entirely on its own. You don't tell it what the groups are — it figures them out from the data patterns.

Real-world analogy

A retail company has thousands of customers. They don't know their customer types in advance. A clustering algorithm automatically discovers groups like 'Budget Shoppers', 'Tech Enthusiasts', and 'Luxury Buyers'. The business then tailors marketing for each group.


Example Dataset — Customer Segmentation


INPUT Features: Age, Income, Spending, Purchase Frequency — NO labels given.

OUTPUT (discovered by algorithm): Cluster label (the names are added by humans after analysing the groups).

Customer	Age	Income (\$k)	Spend (\$)	Freq/mo	Cluster (Output)
Alice	25	30	50	15	Budget Shopper
Bob	35	80	500	8	Tech Enthusiast
Carol	22	28	45	18	Budget Shopper
David	45	150	2000	5	Luxury Buyer
Eve	30	75	480	9	Tech Enthusiast
Frank	50	200	3500	4	Luxury Buyer
Grace	20	25	40	20	Budget Shopper
Henry	38	90	550	7	Tech Enthusiast

The algorithm was never told about 'Budget Shopper' or 'Luxury Buyer' — it grouped customers by similarity in income, spending, and frequency. We then named the clusters after examining their shared characteristics.

 **Key Point:** Most popular clustering algorithm: K-Means. You specify K (number of clusters) and it iteratively assigns each point to the nearest cluster centre.

 **Remember:** Key difference from Classification: in classification you know the labels upfront. In clustering there are no labels — the model discovers the groups itself.

2. Deep Learning (DL)

2.1 What is Deep Learning?

Deep Learning is a specialised subset of Machine Learning that uses artificial neural networks with many layers (hence 'deep') to learn from large amounts of data. It is inspired by the structure of the human brain — neurons connected in layers, each passing signals to the next.

Traditional ML requires humans to manually engineer features (e.g. manually detecting edges in images). Deep Learning skips this step — it automatically discovers the most useful features directly from raw data. This is why DL revolutionised image recognition, speech, and language understanding.

- **Shallow ML:** human extracts features → feeds into algorithm → prediction.
- **Deep Learning:** raw data → neural network learns features automatically → prediction.

💡 **Key Point:** Deep Learning needs large datasets and significant computing power (GPUs/TPUs). For small datasets, traditional ML often performs better, faster, and is easier to interpret.

2.2 What is a Neural Network?

A neural network is a series of layers of 'neurons' (mathematical units). Each neuron receives inputs, multiplies them by weights (importance scores), sums them up, applies an activation function, and passes the result to the next layer. Through training, the network adjusts these weights to minimise prediction errors.

Layers in a Neural Network

- **Input Layer:** receives raw input features — one neuron per feature.
- **Hidden Layers:** one or more layers that learn intermediate representations. More layers = 'deeper' network. This is where feature learning happens.
- **Output Layer:** produces the final prediction — one neuron for regression; one per class for multi-class classification.
- **Weights:** numbers controlling the strength of connections between neurons. These are what the model learns during training.
- **Activation Function:** a mathematical function (e.g. ReLU, Sigmoid) that adds non-linearity, enabling the network to learn complex patterns.

Neural Network Architecture — Diagram

The diagram below shows a fully-connected feedforward neural network — 3 input neurons (yellow), 3 hidden layers of 4 neurons each (teal/dark), and 1 output neuron (orange). Every neuron in one layer connects to every neuron in the next layer.

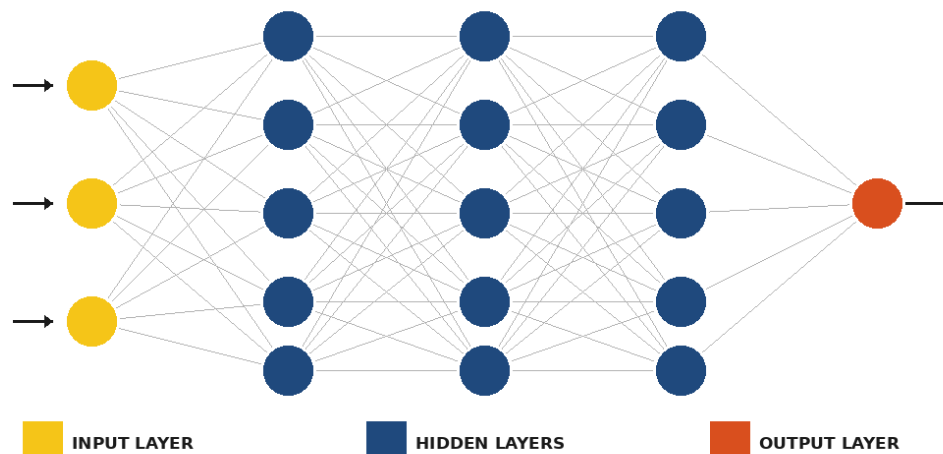


Figure 1: Feedforward Neural Network — Input Layer → Hidden Layers → Output Layer

The arrows represent weighted connections. During training, a process called Backpropagation sends errors backwards through the network, adjusting each weight until predictions improve. This is repeated for many epochs (passes through the data).

💡 **Key Point:** Activation functions: ReLU (hidden layers — fast, avoids vanishing gradient), Sigmoid (binary output — probability 0-1), Softmax (multi-class output — probabilities sum to 1).

2.3 How Does a Neural Network Learn?

Learning is an iterative cycle of three steps repeated thousands of times:

- **1. Forward Pass:** input data flows through the network layer by layer, producing a prediction.
- **2. Loss Calculation:** a loss function (e.g. MSE for regression, Cross-Entropy for classification) measures how wrong the prediction is.
- **3. Backpropagation + Gradient Descent:** the error is sent backwards; each weight is nudged slightly in the direction that reduces the error. This is repeated for every batch of training data.

🔑 **Remember:** An 'epoch' is one complete pass through the entire training dataset. Training usually requires tens to hundreds of epochs before the model converges.

2.4 Common Deep Learning Architectures

Architecture	Best For	Example Use Case
CNN (Convolutional Neural Network)	Images & visual data	Face recognition, Medical image diagnosis, Self-driving cars
RNN (Recurrent Neural Network)	Sequential / time-series data	Stock price prediction, Sensor data forecasting
LSTM (Long Short-Term Memory)	Long sequences with long context	Text generation, Speech recognition, Music generation
Transformer	Language, long-range dependencies	GPT, BERT, ChatGPT, Google Translate

GAN (Generative Adversarial Network)	Generating realistic new data	Image synthesis, Deepfakes, AI Art
Autoencoder	Compression & anomaly detection	Fraud detection, Data denoising, Recommendation


💡 **Key Point:** Transformers are the backbone of modern GenAI (ChatGPT, Gemini, Claude). They use 'self-attention' — a mechanism that weighs the importance of every word relative to every other word simultaneously.

3. Natural Language Processing (NLP)

3.1 What is NLP?

Natural Language Processing (NLP) is a field of AI that enables computers to understand, interpret, and generate human language — written text or spoken words. Language is the most natural way humans communicate, but it is extremely complex for machines: full of ambiguity, context, sarcasm, and cultural idioms.

NLP bridges the gap between human communication and machine understanding. It combines linguistics, statistics, and machine learning to process text and speech at scale.

 **Key Point:** NLP powers Google Search, autocorrect, voice assistants (Siri, Alexa), chatbots, translation services, and modern LLMs like ChatGPT and Claude.

3.2 Traditional NLP Pipeline

Before deep learning, NLP relied on a step-by-step text processing pipeline. Understanding this pipeline shows why modern DL-based NLP is such an improvement — it automates and improves every step.

Step 1 — Tokenisation

Breaking raw text into individual units (tokens) — usually words or subwords. This is the first step in every NLP system.

```
Text: "I love Machine Learning!"  
Tokens: ["I", "love", "Machine", "Learning", "!"]
```

Step 2 — Stop Word Removal

Removing common words that carry little meaning (like 'the', 'is', 'a', 'and'). This reduces noise and focuses the model on meaningful content words.

```
Before: ["I", "love", "the", "study", "of", "AI"]  
After: ["love", "study", "AI"]
```

Step 3 — Stemming(root form) / Lemmatisation(dictionary form)

Reducing words to their root form so 'running', 'ran', 'runs' are all treated as 'run'. This shrinks vocabulary size and helps the model generalise across word forms.

```
running → run      |      better → good      |      studies → study
```


Step 4 — Feature Extraction (TF-IDF / Bag of Words)

Converting text into numerical vectors that a machine learning model can process. Computers cannot understand words — they need numbers.

```
Bag of Words: counts how many times each word appears in a document.  
TF-IDF:       weights words by uniqueness across all documents.  
               (common words like 'the' get low weight; rare words get high weight)
```

Step 5 — Model Training & Prediction


The numerical vectors are fed into a traditional ML algorithm (Naive Bayes, SVM, Logistic Regression) to train a model and make predictions on new text.

 **Key Point:** The big limitation of traditional NLP: it loses word order and context. 'Dog bites man' and 'Man bites dog' produce the same Bag-of-Words vector — yet mean completely different things!

3.3 Traditional NLP Use Cases

These are the classic, well-established NLP tasks that were solved using traditional ML + NLP pipelines — before the era of large language models. Every modern LLM can do all of these tasks in a single model.

NLP Task	What It Does	Example
Sentiment Analysis	Determines emotional tone (positive / negative / neutral)	"Great product!" → Positive
Text Classification	Assigns text to predefined categories	News article → Sports / Politics / Tech
Spam Detection	Identifies unwanted or malicious messages	Email → Spam / Not Spam
Named Entity Recognition (NER)	Extracts names, places, dates, organisations from text	"Apple in California" → [ORG][LOC]
Machine Translation	Translates text from one language to another	"Bonjour" → "Hello"
Text Summarisation	Produces a shorter version of a long document	5-page article → 3-sentence summary
Question Answering	Finds the answer to a question within a given passage	Q: Who founded Apple? A: Steve Jobs
POS Tagging	Labels each word with its grammatical role	"The cat sat" → [DET, NOUN, VERB]
Keyword Extraction	Identifies the most important words in a document	Article about ML → ['neural network', 'model']

 **Remember:** Modern LLMs (GPT-4, Claude, Gemini) perform all these tasks in a single model without task-specific training — they learn general language understanding from billions of text examples.

4. Generative AI (GenAI)

4.1 What is Generative AI?

Generative AI is a type of Artificial Intelligence that can create new, original content — text, images, audio, video, code, and more — that did not previously exist. It goes beyond analysing or classifying existing data; it generates something entirely new based on learned patterns.

Generative AI models learn the underlying structure of their training data so thoroughly that they can produce new examples indistinguishable from real ones. Ask ChatGPT to write a poem — it has never seen that exact poem before, but it generates one that sounds completely natural and coherent.

💡 Key Point: Examples: ChatGPT / Claude (text), DALL-E / Midjourney (images), Suno (music), Sora (video), GitHub Copilot (code). Common thread: they all generate new content from a prompt.

4.2 How is GenAI Different from Traditional ML / DL / NLP?

Aspect	Traditional ML	Deep Learning	Traditional NLP	Generative AI
Primary Goal	Predict / classify	Learn complex features	Process & understand text	Generate new content
Output Type	Number or label	Number, label, or features	Classification, entities, tags	Text, image, audio, video, code
Data Required	Small–medium, labelled	Large labelled datasets	Text corpus + task labels	Massive unlabelled data (web-scale)
Training Style	Supervised / Unsupervised	Supervised + self-supervised	Task-specific training	Pre-training + fine-tuning + RLHF
Flexibility	One model = one task	One architecture, similar tasks	Task-specific models	One model = many tasks (general)
Interpretability	High — clear decision rules	Low — black box	Medium	Very low — billions of parameters
Example	House price prediction	Image recognition (CNN)	Spam filter, NER	ChatGPT, DALL-E, GitHub Copilot


🔑 Remember: The key conceptual shift: Traditional ML/DL = discriminative (learns to distinguish between classes). Generative AI = generative (learns to produce new examples). GenAI models like GPT learn to predict the next token — which turns out to be a remarkably powerful way to learn language, reasoning, and knowledge simultaneously.

4.3 How GenAI Models Generate Text

Text generation in an LLM works through token prediction. Given a prompt, the model predicts the most likely next token (word or subword), appends it, and repeats until the response is complete. This is called autoregressive generation.

```
Prompt: "The capital of France is"
Step 1: Predict next token → " Paris" (high probability)
Step 2: "...France is Paris" → predict → "."
Output: "The capital of France is Paris."
```

The model doesn't 'look up' facts from a database — it learned statistical patterns from training data. This is why LLMs can sometimes 'hallucinate' (generate plausible-sounding but incorrect information).

 **Remember:** Hallucination is a critical limitation of GenAI: models can confidently state false information. Always verify important facts from LLM outputs against reliable sources.

4.4 GenAI Applications Across Domains

Domain	GenAI Application	Tool / Model
Text & Language	Chatbots, writing assistance, summarisation, Q&A	ChatGPT, Claude, Gemini
Code Generation	Auto-complete, explain code, debug, write tests	GitHub Copilot, Code Llama
Image Generation	Create images, logos, illustrations from text	DALL-E 3, Midjourney, Stable Diffusion
Audio & Music	Text-to-speech, music composition, voice cloning	ElevenLabs, Suno, Bark
Video Generation	Generate short videos from text descriptions	Sora (OpenAI), Runway
Science	Protein structure prediction, drug discovery	AlphaFold (DeepMind)
Business	Document analysis, contract review, report gen.	GPT-4, Claude, Gemini Pro

5. The AI Umbrella — How Everything Fits Together

5.1 AI Hierarchy Diagram

A common misconception is that ML, DL, NLP, and GenAI are completely separate fields. In reality, they are nested inside each other like Russian dolls — AI is the broadest concept and everything else is a specialised subset of it.

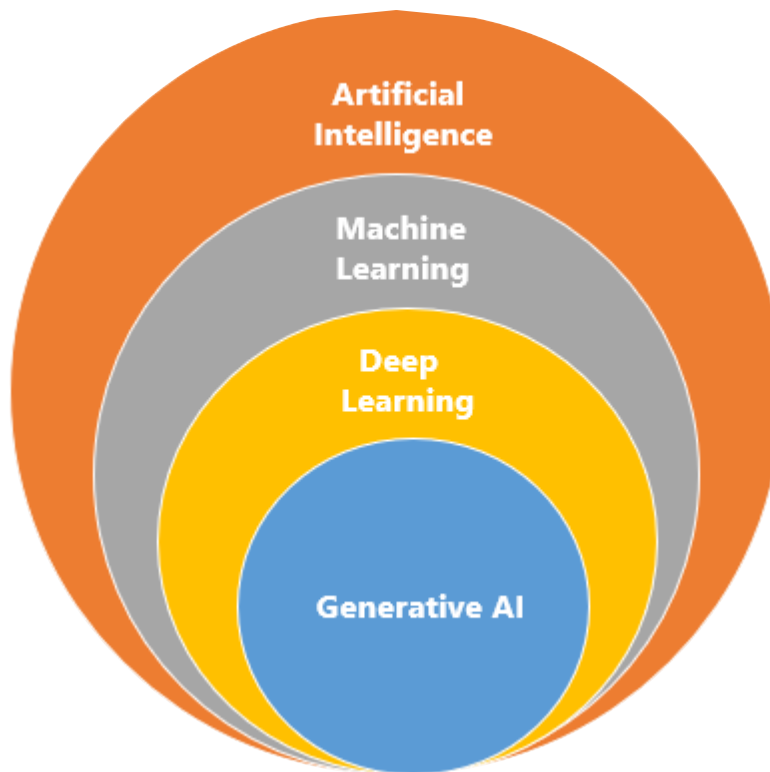


Figure 2: The AI Hierarchy — Artificial Intelligence \supset Machine Learning \supset Deep Learning \supset Generative AI

5.2 Understanding Each Layer

Artificial Intelligence (AI) — The Outer Ring

AI is the broadest field — any technique that enables machines to simulate human intelligence. This includes rule-based expert systems, search algorithms, traditional ML, deep learning, and GenAI. Everything below is a subset of AI.

Machine Learning (ML) — Second Ring

ML is AI that learns from data, rather than following hand-written rules. Instead of programming rules, you provide examples and the algorithm discovers the rules. Includes Linear Regression, Decision Trees, SVM, K-Means, Random Forests, and all of Deep Learning.


Deep Learning (DL) — Third Ring

DL is ML that uses multi-layer neural networks. All DL is ML, but not all ML is DL. DL excels at unstructured data (images, audio, text) but requires large datasets and GPU computing. Traditional ML often works better for structured/tabular data with limited samples.

Generative AI (GenAI) — Inner Circle

GenAI is the newest and most visible layer. Most modern GenAI systems are built on Deep Learning (specifically Transformer architectures) and extend it by focusing on generation rather than classification. GenAI requires the largest datasets, the most compute, and the most sophisticated training procedures.

Layer	Core Idea	Relationship to AI	Example
AI	Machines simulating intelligence	The entire field	Chess engines, Expert systems, ALL of ML
ML	Learning from data without explicit rules	Subset of AI	Random Forest, Linear Regression, K-Means
DL	Neural networks with many layers	Subset of ML	CNN (images), LSTM (sequences), Transformers
NLP	Processing human language	Domain using ML + DL	Spam filter (ML), GPT-4 (DL/GenAI)
GenAI	Creating new content from learned patterns	Subset of DL (mostly)	ChatGPT, DALL-E, Sora, GitHub Copilot

 **Remember:** A common exam question: 'Is ChatGPT an ML system?' Answer: Yes! It is AI → ML → DL → NLP → GenAI. Each label is correct — they describe different levels of the same hierarchy.

5.3 When to Use What — Practical Decision Guide

Situation	Best Approach	Reason
Small labelled dataset, tabular/structured data	Traditional ML (RF, XGBoost)	DL needs large data; ML generalises better with less data
Large labelled dataset, images / audio / video	Deep Learning (CNN, LSTM)	DL automatically extracts complex features from raw data
Text classification with labelled training examples	Traditional NLP or LLM fine-tune	TF-IDF+SVM is fast; fine-tuned LLMs give higher accuracy
No labels — want to find natural groups in data	Unsupervised ML (K-Means, DBSCAN)	Clustering discovers structure without requiring labels
Need to generate text, images, or code	Generative AI (LLM, Diffusion)	Only GenAI can create new, original content from a prompt
Limited compute / need explainability	Traditional ML	Simpler, faster, and interpretable for stakeholders
One model for many different language tasks	LLM (GPT, Claude, Gemini)	Foundation models generalise without per-task retraining