

Artificial Intelligence (AI)

AI refers to the simulation of human intelligence in machines. AI enables machines to perform tasks that typically require human intelligence, like recognizing speech, solving problems, or making decisions.

Example: Voice assistants like Siri or Alexa use AI to understand spoken commands and respond in ways that mimic human conversation.



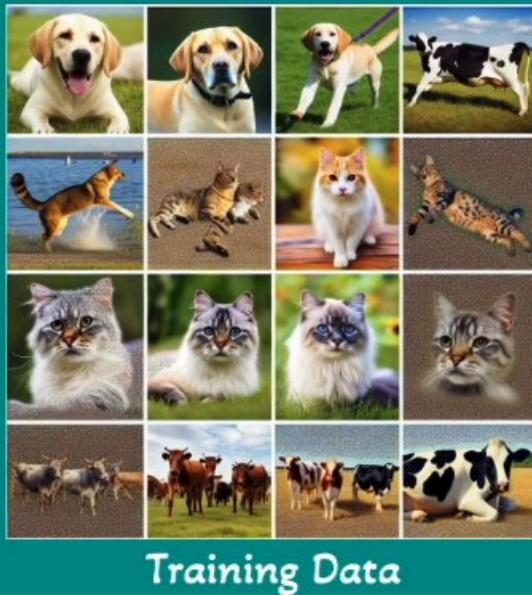
Machine Learning(ML)

ML is a subset of AI focused on creating algorithms that allow computers to learn from data and improve their performance over time without being explicitly programmed for each specific task.

Example: Netflix uses ML to recommend shows based on your viewing history, continuously learning what kinds of shows you like.



How does AI/ML work?



Model Training

Train Model

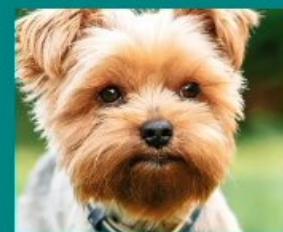


AI Model

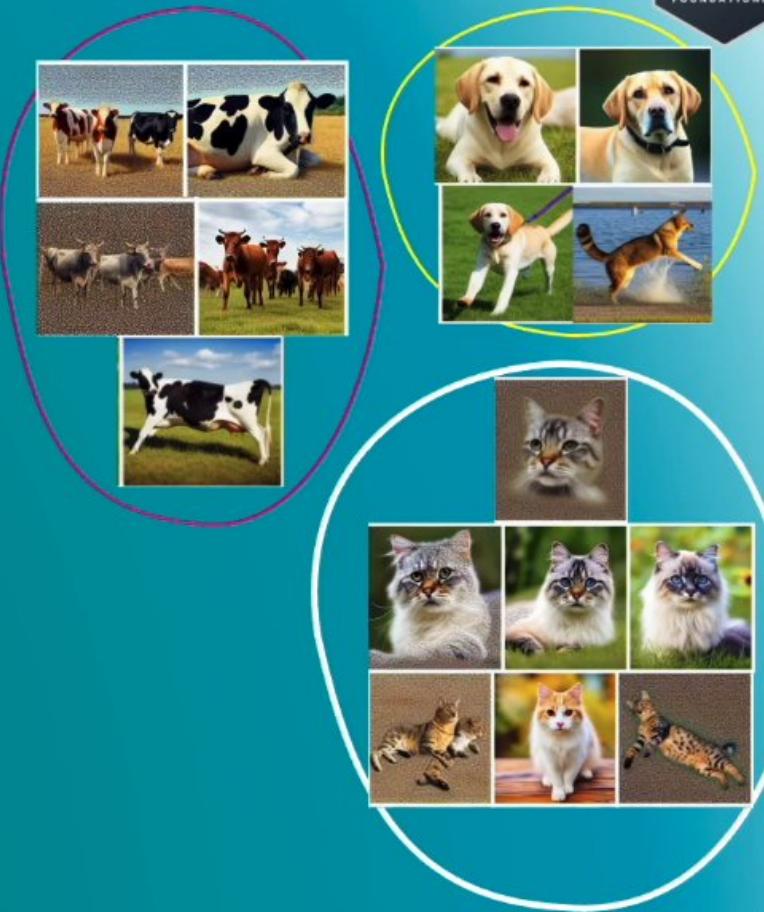
What is this?

Dog

Inferencing



New Object



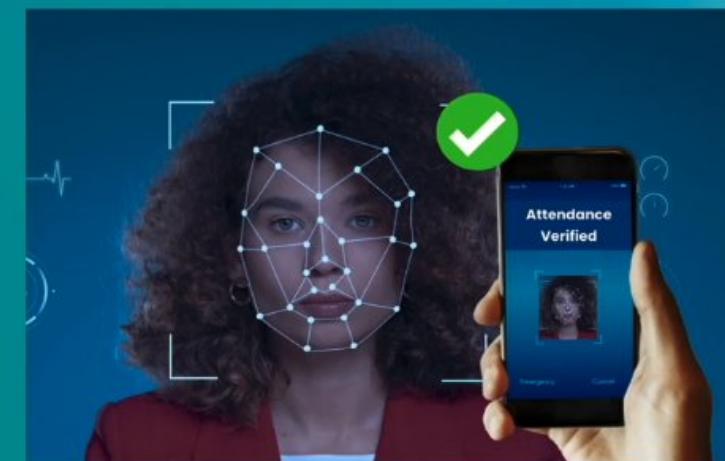
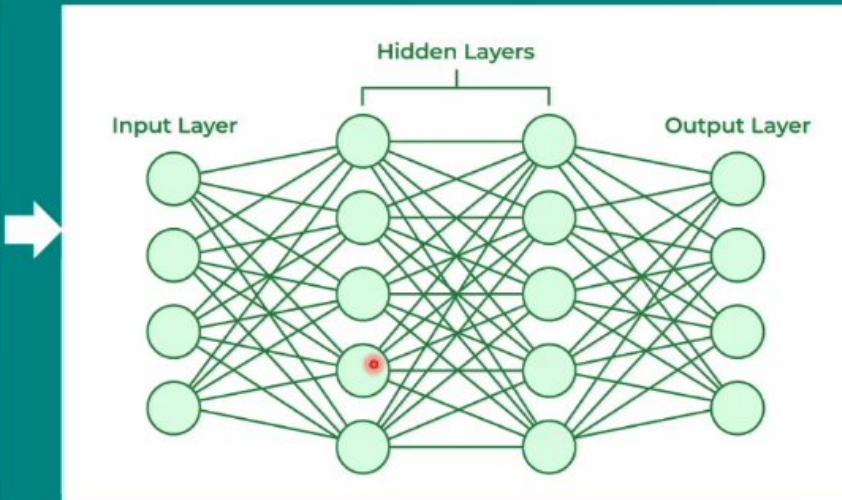
Deep Learning & Neural Network



$$A|a \rightarrow A|a \quad \mathcal{H}|a$$

Neural Network are models inspired by the human brain's structure, made up of layers of connected "neurons" that process data. Each neuron adjusts itself based on the data it receives to recognize patterns over time.

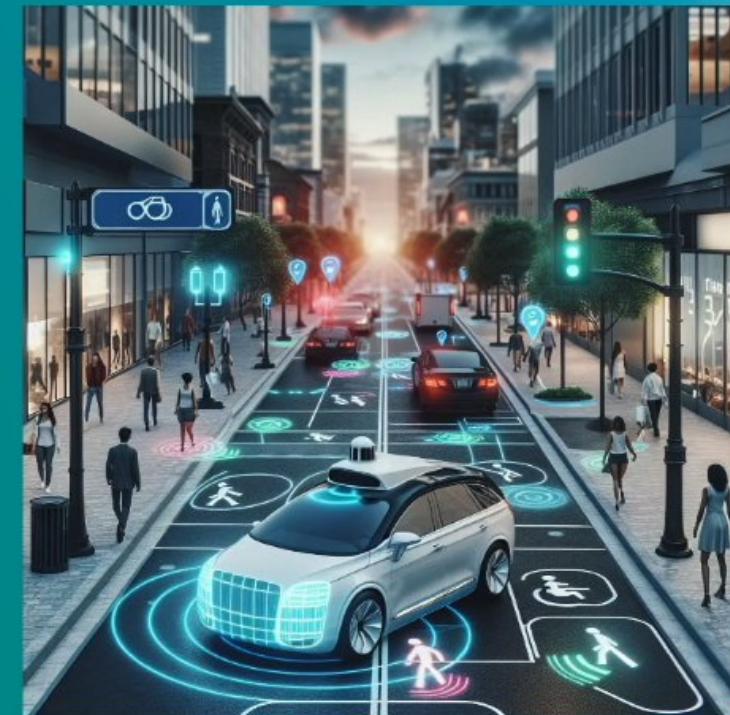
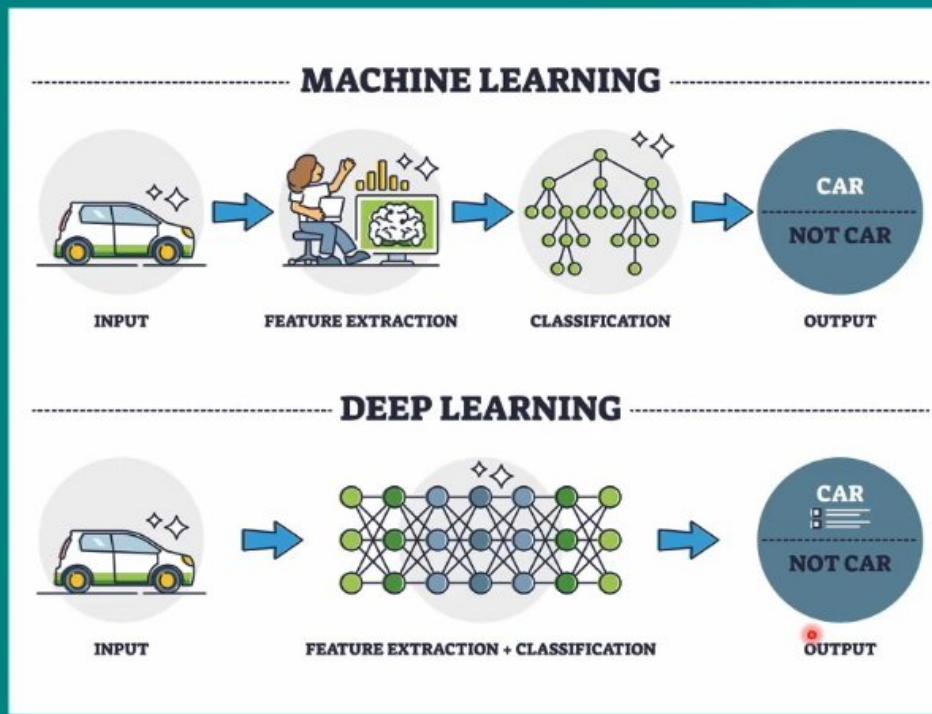
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9



Example: Facial recognition software to recognize and match faces in photos

Deep Learning & Neural Network

Deep learning is a branch of ML that uses neural networks with many layers (often called "deep" networks) to process data in complex ways, allowing for more sophisticated tasks like image and speech recognition.



Generative AI (GenAI)

Generative AI is a subset of artificial intelligence focused on creating new content—like text, images, audio, video, and even code—by "learning" from existing data patterns.



Train Model



GenAI Model

Generate a cartoon bird
singing with microphone



Artificial Intelligence (AI)

Machine Learning
(ML)

Deep Learning

Generative AI
(GenAI)



Foundation Model



- A general-purpose model trained on a massive dataset
- Pre-trained models
- It can be adapted to perform many different tasks
- Wide selection of Foundation Models:
 - GPT (OpenAI)
 - DALL-E (OpenAI)
 - Titan (Amazon)
 - Llama (Meta)
 - Claude (Anthropic)
 - BERT (Google)
- Training a foundation model requires significant time and compute capability
- Example: GPT foundation model trained on diverse text from the internet and codes from Git Hub.

What foundation model you are using?

I'm based on OpenAI's GPT-4, specifically a variant known as GPT-4-turbo. This model is part of OpenAI's lineup of large language models designed for a range of natural language tasks, from answering questions and generating creative content to reasoning and coding. GPT-4-turbo is optimized to be faster and more cost-effective than the original GPT-4, with some differences in training and architecture that OpenAI hasn't fully disclosed.

Switch model

Auto

GPT-4o
Great for most tasks

GPT-4o mini
Faster for everyday tasks

Try again

GPT-4o

Large Language Models (LLM)



- Specific type of Foundational Model focuses exclusively on language-based tasks
- Examples of language-based tasks:
 - Writing essays / Content Creation
 - Answering Questions
 - Language Translation
- All LLMs are Foundational Model, but not all foundational models are LLMs
- Trained on large corpus of text data – books, articles, websites, other text data
- Examples of LLMs:
 - GPT-4 (Open AI)
 - BERT (Google)
 - Llama (Meta)

A screenshot of a Microsoft Copilot AI interface. It shows a conversation between "You" and "Copilot".

You
Are you a LLM?

Copilot
Yes, I am! I'm a large language model (LLM) designed to assist with a wide range of tasks and provide information on various topics.
How can I help you today?

AI-generated content may be incorrect

Generative AI - Prompt



- **Prompt** – a piece of text or instruction provided by the user to guide the AI model in generating a response or output.
- Can be a question, a statement or an instruction
- **Examples of Prompts:**
 - **Question** – “What are the benefits of yoga?”
 - **Creative Request** – “Write a short story about a princesses and a dragon”
 - **Instruction** – “Summarize the key points of the article on climate change.”
- **Non-Deterministic** – the generated text may be different for every users even though everyone use the same prompt.



User

What is LLM?



what is LLM?

A Large Language Model (LLM) is an artificial intelligence system designed to understand and generate human-like text. Built with billions of parameters, LLMs learn language patterns by analyzing vast amounts of text data, like books, websites, and articles. Through training, they develop skills in context comprehension, language generation, and nuanced conversation, making them valuable for tasks like summarizing, translating, and answering questions. The "large" aspect refers to their massive size, allowing them to handle complex language tasks. Notable LLMs include OpenAI's GPT models, Google's BERT, and Meta's LLaMA, each used for diverse applications in language processing.

...

What is LLM?

A Large Language Model (LLM) is a type of artificial intelligence designed to process and generate human language. Built with billions of parameters, these models learn language by training on vast text datasets, enabling them to predict words, answer questions, summarize information, and engage in complex conversations. The "large" refers to the model's size, which helps it understand language intricacies. Examples of LLMs include OpenAI's GPT series, Google's BERT, and Meta's LLaMA, each used in applications like chatbots, content creation, translation, and text analysis.

...



Generative AI for Images

Generates images from text

Prompt - “Can you generate an image where bird is singing holding a microphone?”

Diffusion Model (ex: Stable Diffusion)



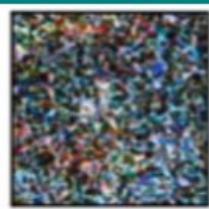
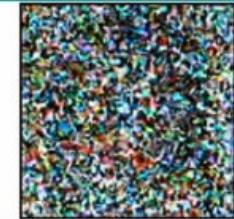
Slightly
Noise



Pretty
Noise



Noise



You

Can you generate an image where bird is singing holding a microphone?

Copilot

I'll try to create that.



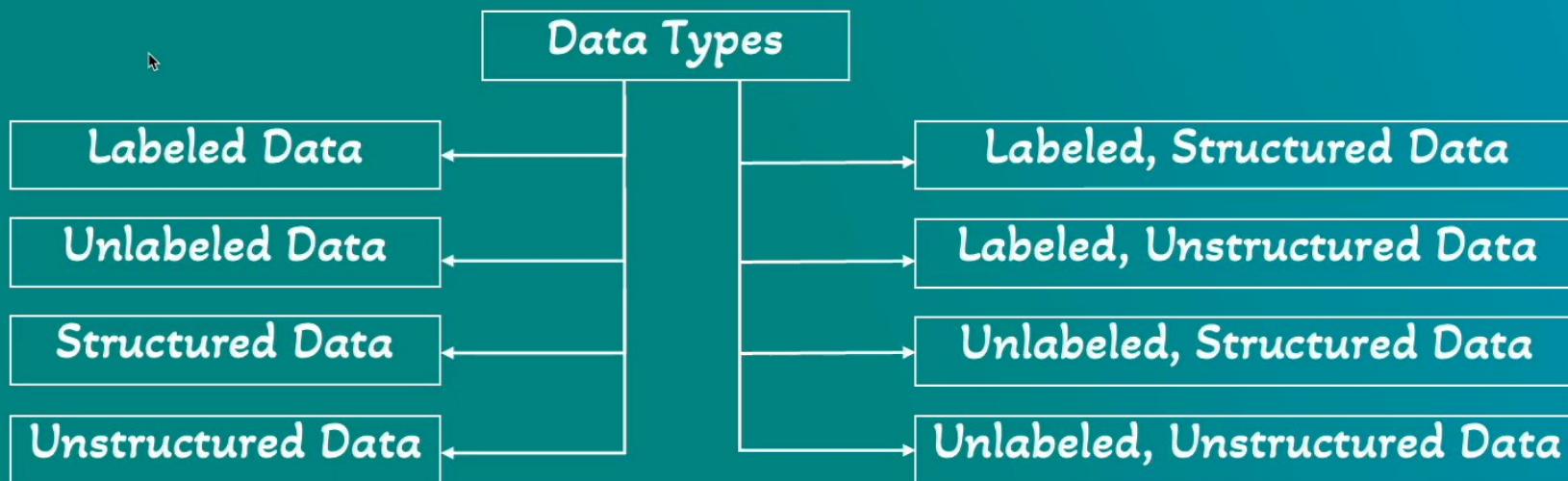
AI-generated content may be incorrect

Designer Preview



Training Data

- Dataset used to train an AI model to recognize patterns, make predictions or generate content.
- Key Characteristics:
 - Diversity – data should cover a wide range of representative
 - Quality – accurate data leads to better and more reliable prediction



Labeled vs. Unlabeled Data

Labeled Data

- Data that comes with tags or “labels” indicating the correct answer or category for each data point.*



Dog



Dog



Cat



Cat

- Use Cases:** Essential for supervised learning, where the model is trained to predict the label based on the input.
- Pros:** Enables the model to learn mappings from input to output.
- Cons:** Expensive and time-consuming, especially for large datasets.

Unlabeled Data

- Data that lacks tags or specific answers, meaning there is no predefined category or outcome.*



- Use Cases:** Common in unsupervised learning, where the model looks for patterns or grouping in the data without guidance.
- Pros:** Easier and cheaper to gather in large amounts since no labeling is required.
- Cons:** Less targeted than labeled data; models may require additional interpretation to extract useful insights.

Structured vs. Unstructured Data

Structured Data

- Highly organized data, typically formatted in rows and columns.
- Stored in table or database with a consistent schema (e.g., spreadsheets or SQL database).



- Use Cases:** ML task requiring numerical or categorical data, financial analysis, inventory management, etc..
- Pros:** Easy to analyze, manipulate and search due to the consistent data schema.
- Cons:** Limitation in capturing complex data types like images, videos or text documents.

Unstructured Data

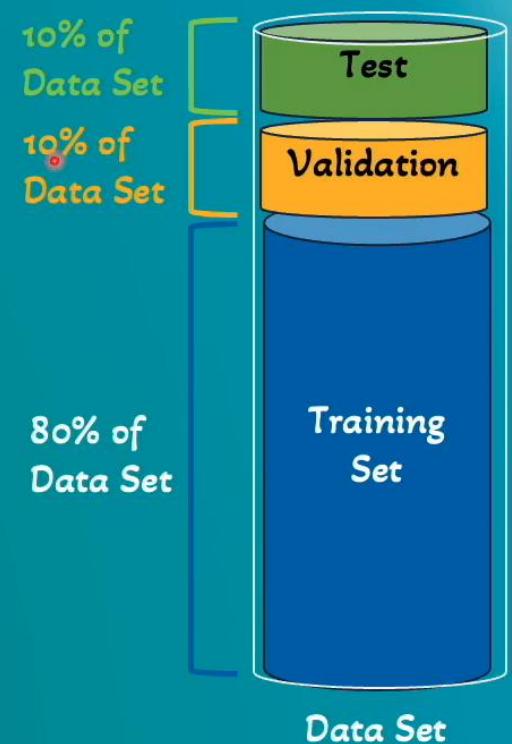
- Data without a predefined format, making it more challenging to organize and analyze directly.
- Free-form, can be text, images, audio or video files.

Amazon Customer
★★★★★ Absolutely gorgeous
Reviewed in the United States on July 31, 2022
Verified Purchase
Looks like the picture, size correctly listed. Gorgeous. Highly recommend it.

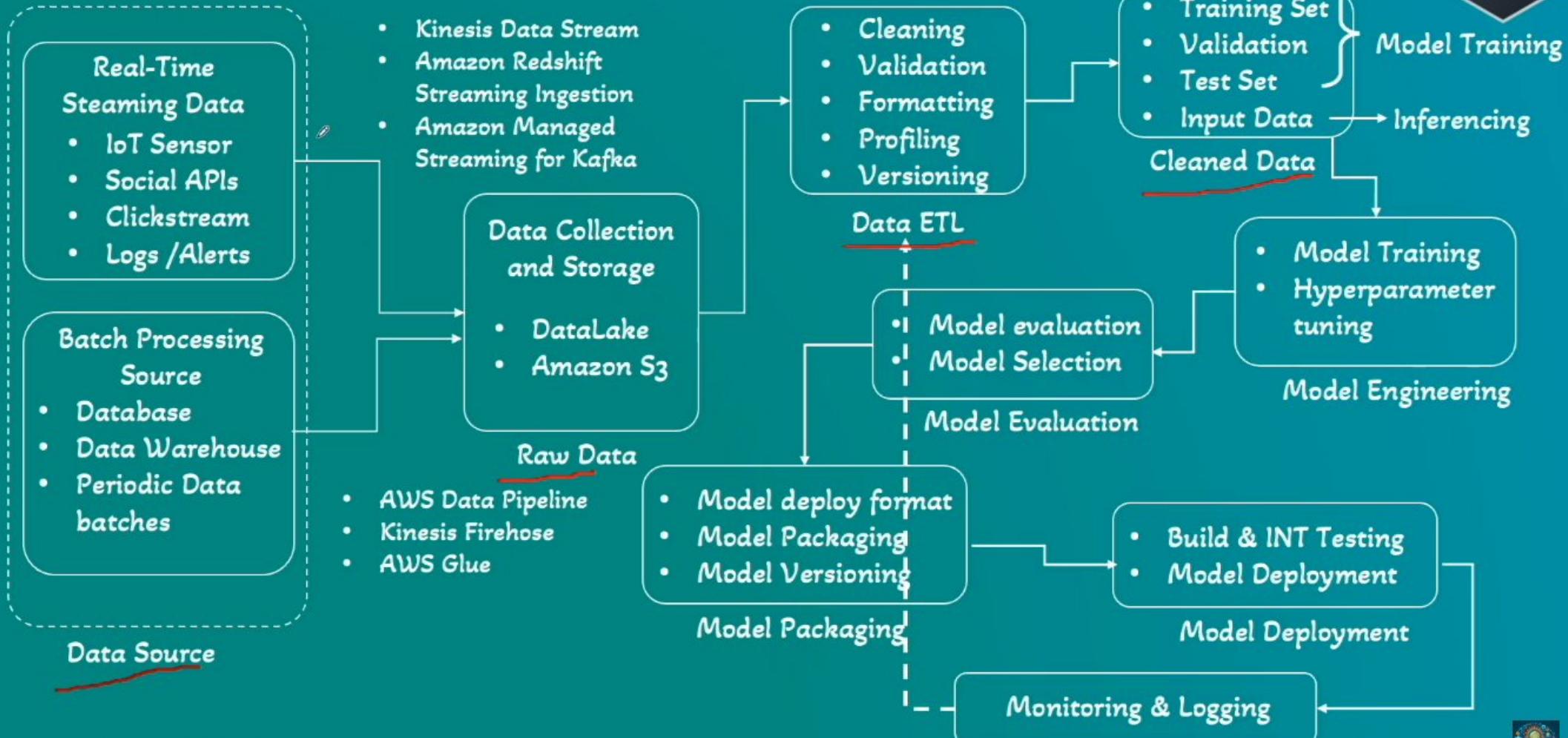
- Use Cases:** Common in natural language processing (NLP), image recognition, customer feedback analysis, and social media monitoring.
- Pros:** Richer data, can capture complex, real-world information.
- Cons:** Requires resource intensive preprocessing to make it useful for analysis.

Training Data – Training Set vs Validation Set vs Test Set

- **Training Set:** Used to train the model.
- **Size:** Largest portion of the data (usually 60-80%) of the data set.
- **Goal:** Help the model to find relationship between input and output data.
- **Validation Set:** Used to tune the model's hyperparameters.
- The model doesn't see this data during training
- **Size:** Typically, 10-20% of the data set.
- **Goal:** Provide feedback on how the model is likely to perform on unseen data and prevent overfitting.
- **Test Set:** Final set used to evaluate model's performance after training.
- The model doesn't see this data during training and validation
- **Size:** Typically, 10-20% of the data set.
- **Goal:** Provide unbiased evaluation of the model's generalization ability on truly unseen data.



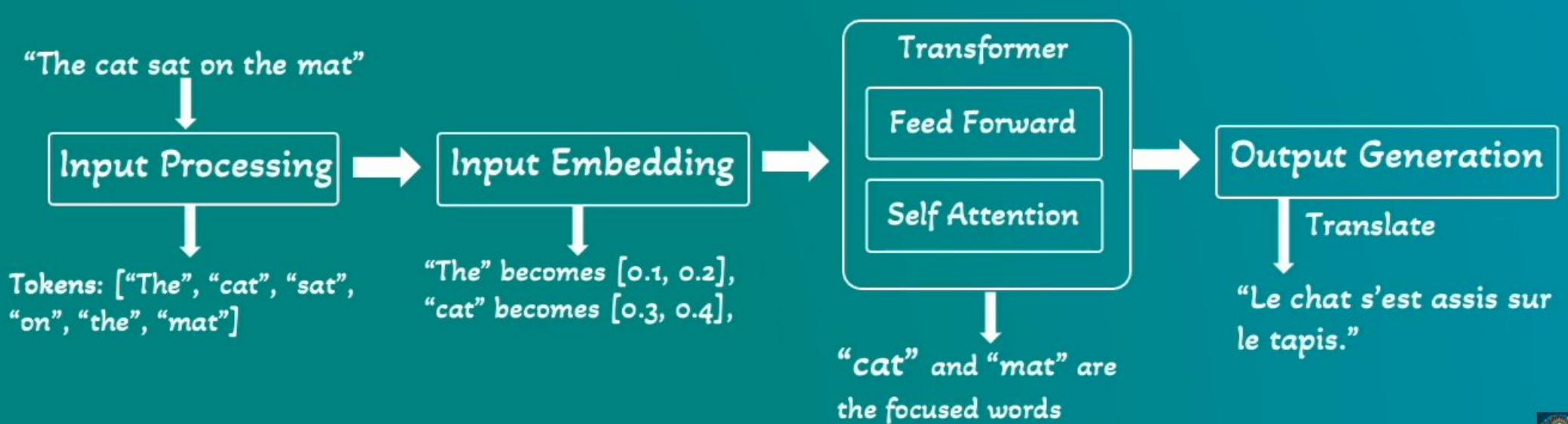
Machine Learning Workflow





Transformer Model (LLM)

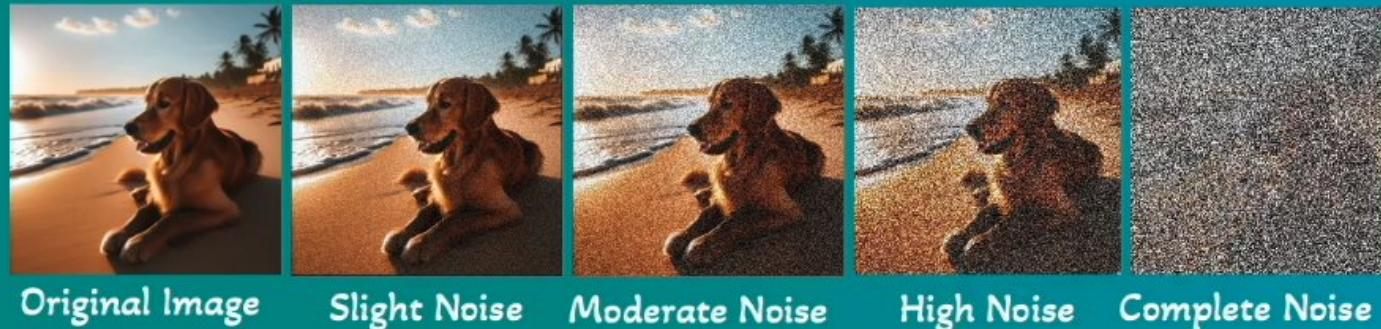
- Used to understand and generate human-like text
- Translate sentences (English to French), text summary, continuation of the text
- Examples:
 - ChatGPT (Chat Generative Pre-trained Transformer)
 - Google BERT (Bidirectional Encoder Representations from Transformers)
- Able to process a full sentence instead of word by word



Text to Image: Diffusion Models



Forward Diffusion → Model learns how image degrade with noise



Original Image Slight Noise Moderate Noise High Noise Complete Noise

Reverse Diffusion → Model generates new images from random noise

Training: Model learns to predict the noise added at each step and how to remove effectively

Generation: Text Prompt → “A dog sitting on a bench”

Starts with a noisy image and uses the learned reverse diffusion process to generate the image



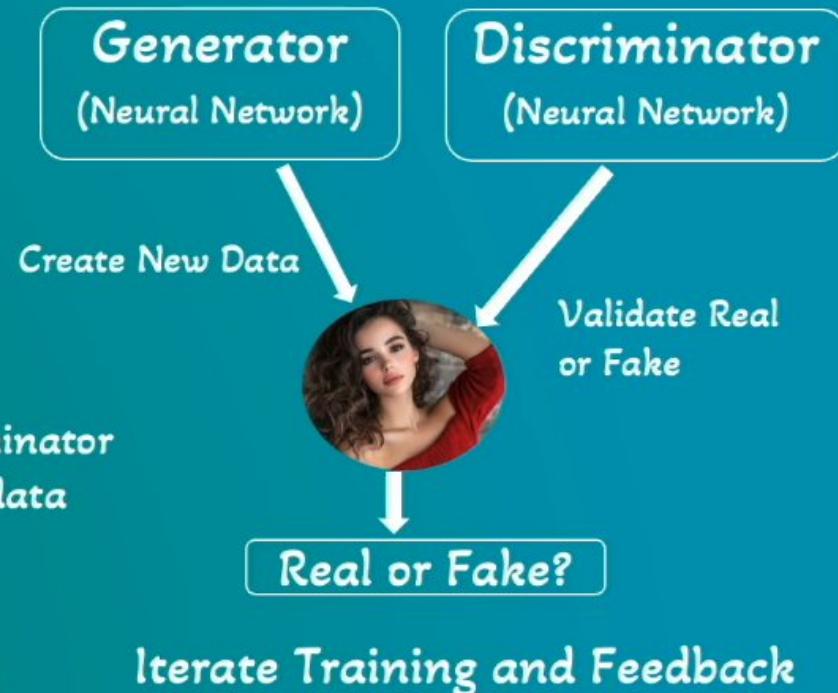


Multimodal Models in GenAI - GANs

- Can generate image from a text prompt
- Can create video content from text scripts
- Can combine text, image, audio and output a mix of video and audio

GANs → Generative Adversarial Networks

- Combination of two Neural Networks
- Training Process:
 - Generator tries to create realistic data to fool the discriminator
 - Discriminator tries to get better at spotting real vs fake data
- Feedback Loop:
 - Discriminator spot a fake → feedback to Generator
 - Generator fooled a discriminator → learns from mistake
- Applications:
 - Creates realistic images, such as faces, animals and landscapes, used in art and music
 - Can generate additional training data for machine learning model



ML Models Name and Purpose for the Exam

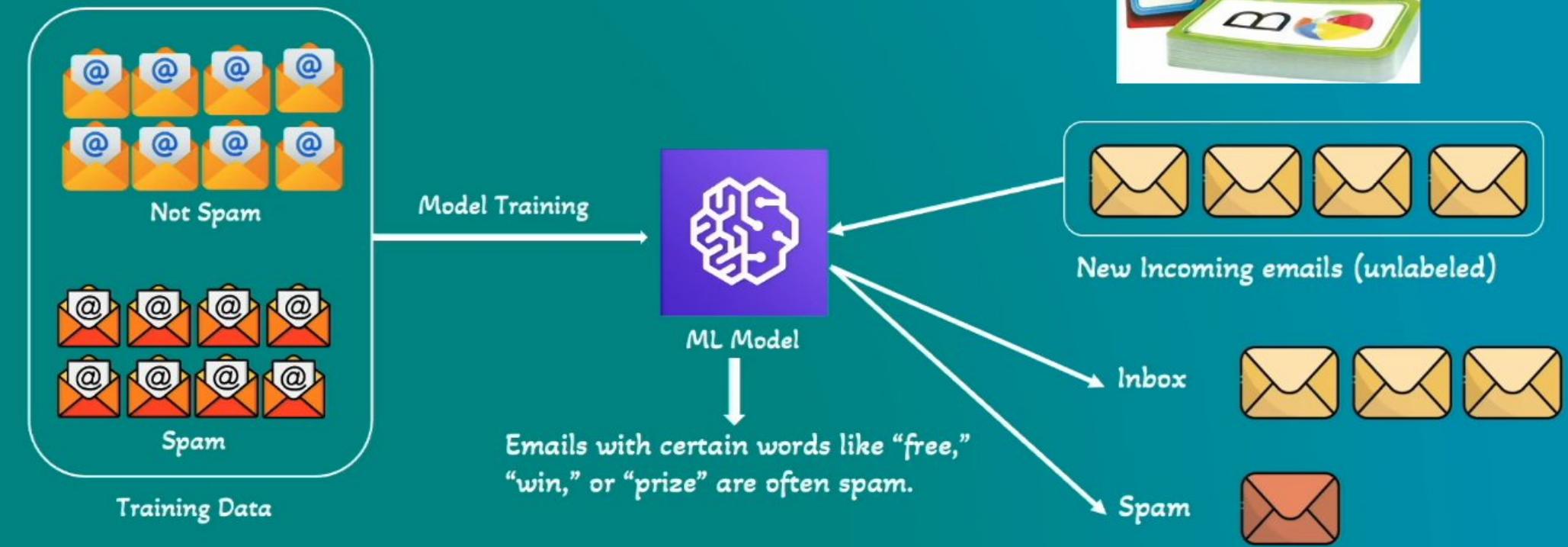


- GPT (Generative Pretrained Transformer) – generates human-like text or source codes
- BERT (Bidirectional Encoder Representations from Transformers) – like GPT but can understand the context of words in a sentence bidirectionally.
- SVM (Support Vector Machine) – ML algorithm for classification and regression
- WaveNet – generates raw audio waveform (Used in Speech Synthesis)
- RNN (Recurrent Neural Network) – used in speech recognition, time-series prediction
- ResNet (Residual Network) – used for image recognition, object detection, facial recognition
- GANs (Generative Adversarial Networks) – used to generate synthesis data. Helpful in data augmentation
- XGBoost (Extreme Gradient Boosting) – enhance performance and speed of gradient boosting models

Supervised Learning

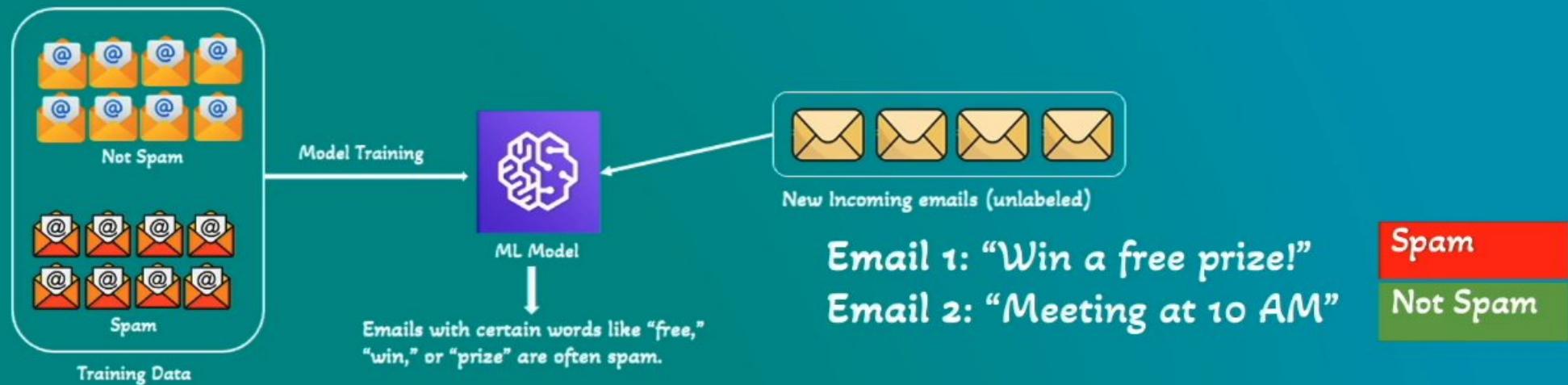
Model is trained on labeled data. This means the data includes both the input and the correct output.

Example: Create a ML model to recognize spam emails.



Supervised Learning - Classification

- Used to predict the categorical variables
- Output variable is discrete, and it will fall into a specific category or class



Classification Use Cases:

- Prediction whether a patient has breast cancer or not
- Predict whether a credit card transaction is fraudulent or legit
- Predicting whether a loan application will be approved or rejected
- Classifying images of crops to detect diseases early.

Supervised Learning - Regression

- Used to predict numerical values based on one or more input features
- Examples:
 - Predicting house price based on house size, location and number of bedroom
 - Predicting the future price of a stock based on historical data

Regression Modeling Steps:

Data Collection

House Size (SQ Ft)	Bedroom	Location	Year of Built	Price
2000	3	City A	2002	\$ 500,000.00
1500	2	City B	1985	\$ 300,000.00
2500	4	City A	1997	\$ 600,000.00
1800	3	City B	2022	\$ 350,000.00
2000	2			\$ 450,000.00

Data Processing

House Size (SQ Ft)	Bedroom	Location	House Age	Price
2000	3	City A	22	\$ 500,000.00
1500	2	City B	39	\$ 300,000.00
2500	4	City A	27	\$ 600,000.00
1800	3	City B	2	\$ 350,000.00

Model Training

Model Training



Model Evaluation

Mean Squared Error

R-Squared

Prediction



House Size (SQ Ft)	Bedroom	Location	House Age
2200	3	City B	22
1600	2	City A	15

Trained Model

Use Case: When the goal is to predict a quantity or a real value



Unsupervised Learning ML Algorithms

- ML Models trained on unlabeled data
- ML Models discover hidden structures, learn patterns in data
- Benefits of Unsupervised Learning
 - Discover hidden patterns in stored data
 - Finds structures in data without predefined labels
 - Ideal for data exploration/ exploratory analysis
- Common Techniques/Algorithms
 - Clustering (K-Means Clustering)
 - Dimensionality Reduction (Principal Component Analysis)
 - Anomaly Detection (Isolation Forest)





Unsupervised Learning - Clustering

- Groups unlabeled data into K cluster based on feature similarities
- Algorithm used → K-Means Clustering
- K in K-means clustering refers to the number of cluster

Use Case:

A retailer wants to create targeted marketing campaign for each customer segments, to offer personalized promotions and enhance CSAT

→ Segmenting customers into different groups based on purchasing behavior.

Process:

- Data: Customer purchase history, including annual income, and spending

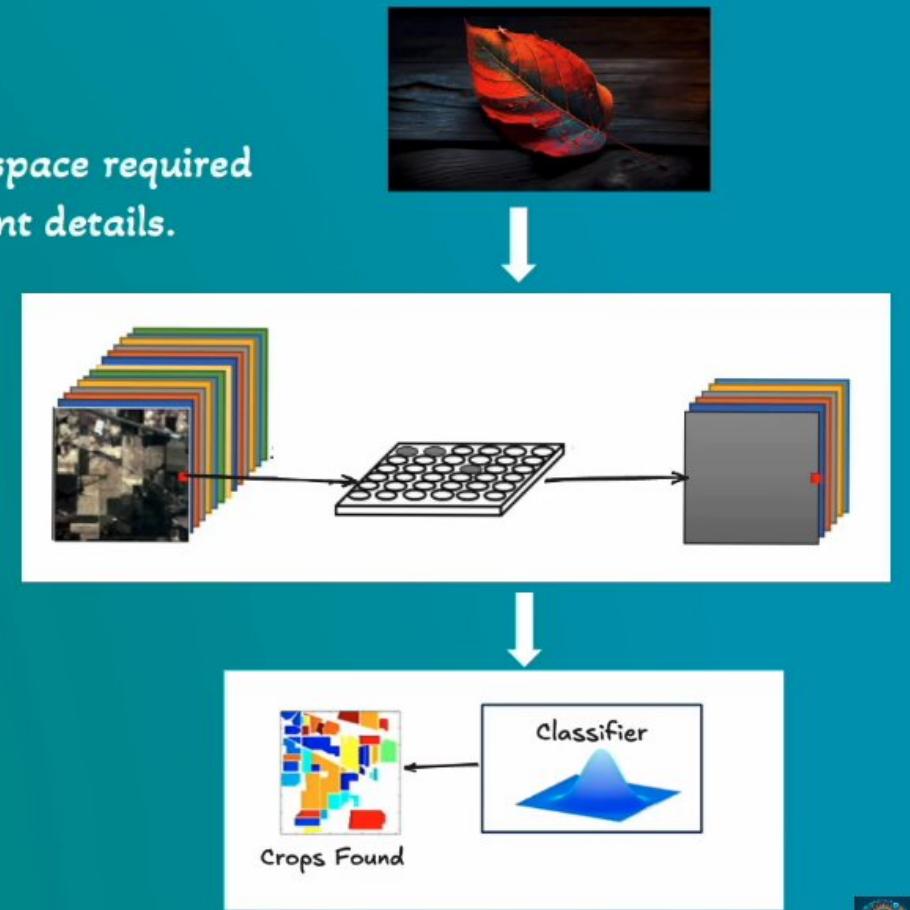
Outcome:

Groups the customers into customer segments, such as high spenders, budget shoppers and occasional buyer.



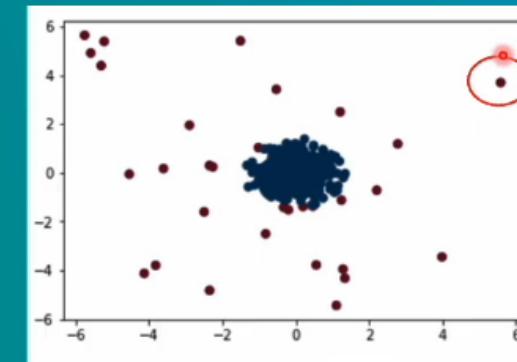
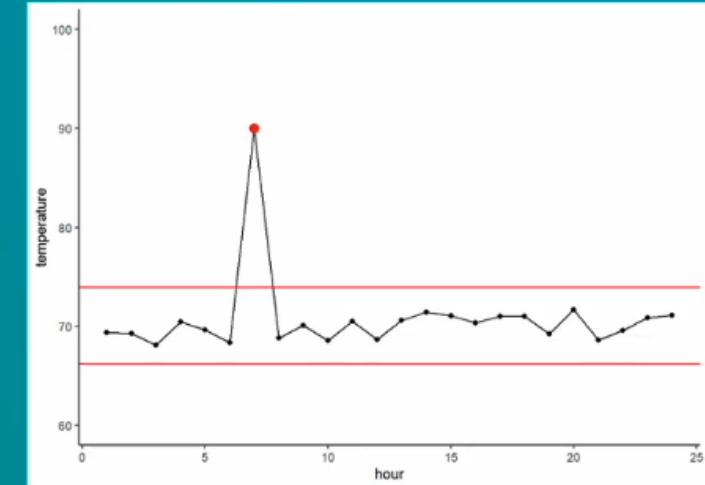
Unsupervised Learning – Dimensionality Reduction

- Used to simplify a dataset with many features for easier visualization and analysis
- Algorithm used → Principal Component Analysis
- Use Case: A tech company wants to reduce the storage space required for high-resolution images/videos without loosing significant details.
- Process:
- Data: High-resolution images (many pixels)
- Principal Component Analysis reduces the number of features while retaining most important information
- Outcome:
 - Compressed images that require less storage space but still maintain visual quality



Unsupervised Learning – Anomaly Detection

- Identify unusual patterns or behaviors that are significantly deviates from the standard behavior pattern.
- Algorithm used → Isolation Forest
- Use Case: A bank wants to detect fraudulent transactions in real-time.
- Process:
- Data: Transaction data, including location, amount, time
- Isolation Forest algorithm identifies transactions that deviates significantly from normal patterns
- Outcome: Flags suspicious transactions for further investigation.



Semi Supervised Learning

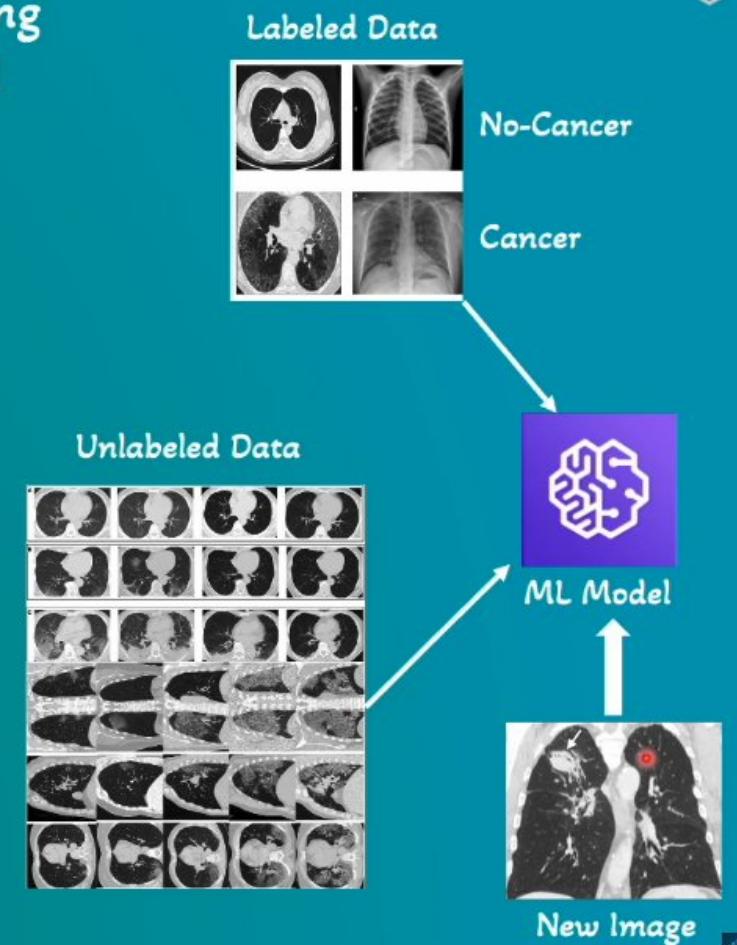
- In between supervised learning and unsupervised learning
- Uses both labeled and unlabeled data to train the model
- Use Case: A hospital wants to develop a ML model to diagnose cancer diseases from medical images.

Process:

- Labeled Data - A small set of medical images labeled by doctors
- Unlabeled Data – A large set of medical images without labels

Training:

- Labeled images helps to learn correct diagnosis
- Unlabeled images helps to understand general structure and features of medical images





Reinforcement Learning

- ML algorithm where an agent learns to make decisions by performing actions in an environment to maximize cumulative reward.
- Key Components
 - Agent: the learner or decision-maker The Dog
 - Environment: external system the agent interacts with The backyard
 - State: a specific situation (e.g., current state) The dog's current position in the yard.
 - Action: a move the agent can make The dog can run, sit or fetch the stick.
 - Reward: feedback from the environment The dog gets a treat if it fetches the stick correctly.
- Process:
 - Exploration: The dog tries different actions to see what works
 - Reward: A treat when the dog fetches the stick (+VE reward)
 - Learning: Over time, the dog learns that fetching the stick leads to treat and is more likely to repeat this action.



Teaching a Dog

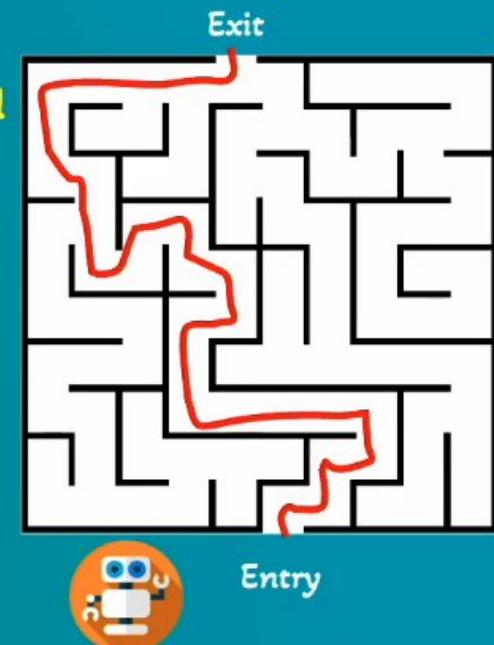


Reinforcement Learning

- A maze game where the robot needs to exit the maze with highest point.

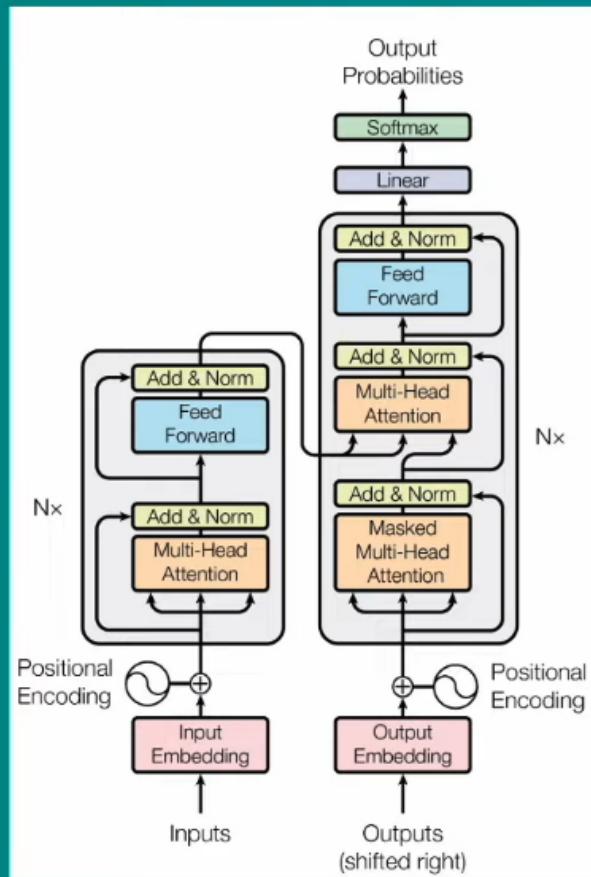
Key Components

- Agent: the learner or decision-maker **The Robot**
- Environment: external system the agent interacts **The gameboard**
- State: a specific situation (e.g., current state) **Current position + Reward**
- Action: a move the agent can make **Left, right, up and down**
- Reward: feedback from the environment **Collect points and loses for hitting obstacles**
- Process:
 - Exploration: robot tries different routes to exit
 - Reward: Unblocked moves -1, hitting wall -10
 - Learning: Over time, the robot learns to avoid obstacle and collect points to maximize the score.



Transformer Model (LLM)

Transfer Model Architecture



LLMs follow the Transfer Model:

- GPT-3 (OpenAI)
- BERT (Google)
- Llama (Meta)

Machine Learning Inferencing

- *The process of using a trained machine learning model to make prediction on new data.*

Machine Learning Inferencing - Batch

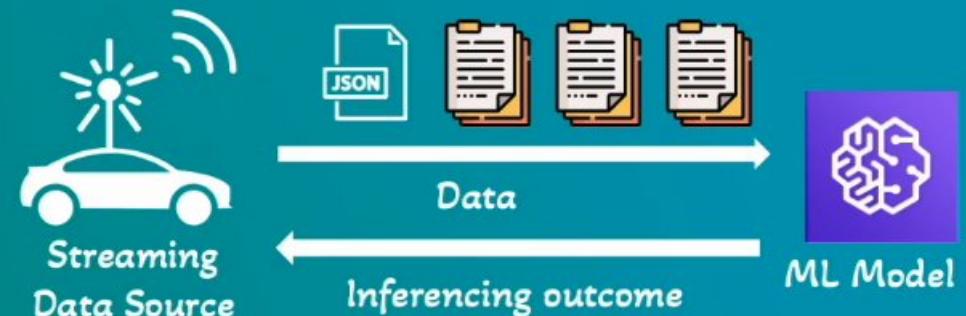
- Bulk processing of large dataset, typically scheduled (daily, weekly, monthly)
- Offline processing or inferencing



- **Use Cases:**
 - Monthly Customer Churn Prediction
 - Monthly sales forecasting
- **Advantages:**
 - Effective resource management
 - Cost effective and efficient
- **Disadvantages:**
 - Latency in getting results
 - Not suitable for real-time decision-making

Machine Learning Inferencing – Real-Time

- Making predictions on individual data points, with minimal delay
- For Latency sensitive workloads
- Use Cases:
 - Fraud Detection
 - Chatbot
 - Personalize recommendations
 - Autonomous Vehicles
- Advantages:
 - Immediate outcome, suitable for latency-sensitive applications
 - Enhance user experience
 - Increased efficiency
- Disadvantages:
 - Requires robust infrastructure to handle high throughput
 - Potentially higher costs due to continuous operations



Real-Time vs Batch Inferencing Comparison

	Real-Time	Batch
Latency	Near to zero (in Milliseconds)	High Latency (in Hours)
Data Volume	Small (in MBs)	Large (in GBs / TBs)
Resource Utilization	Continuous (24 X 7)	Scheduled
Use Cases	Latency Sensitive	Latency Intensive

Edge Inferencing

Bringing AI closer to the data sources

- Autonomous Vehicles.
- Wearable health devices.

Fitness Tracker Smartwatches Wearable biosensors Smart contact lenses

- Smart Factories.

What is Edge Inferencing?

Running trained machine learning model directly on local devices

Reduces latency, improves efficiency and enhance privacy

Use Cases or Real-life Examples:

Tesla Autopilot Cars

Apple Watch – Heart rate monitoring

Smart Factory – Industry 5.0

Edge Inferencing Advantages vs Disadvantages

Advantages

- Low Latency
- Enhanced Privacy
- Bandwidth Efficiency
- Scalability

Disadvantages

- Resource Constraints
- Maintenance
- Complexity

Language Models for Edge Inferencing

Small Language Models (SLMs)

- Lightweight and efficient
- Speed
- Lower computational and memory requirements.
- Good for simpler tasks and resource-constrained environments.

Use Cases

Mobile applications

Edge Devices

Example

TinyBERT

DistilBERT

Large Language Models (LLMs)

- High accuracy, handle complex tasks
- Can perform wide range of tasks
- Ideal for apps requires deep understanding and text generation
- Higher computational & memory requirements

Use Cases

Content Creation

Customer Support

Example

GPT-4

Feature Engineering

What is feature?

- Individual measurable property
- Represent a specific piece of information about the data
- Column in a tabular dataset
- Inputs that ML models use to make predictions
- Quality and relevance directly affect the model accuracy

What is feature engineering?

- Process of creating new features from existing data
- To make data more useful and enhance the accuracy

Feature engineering Process

- Feature Creation
- Feature Transformation
- Feature Selection
- Scaling and encoding
- Exploratory Data Analysis

Dataset of houses

Location	Bedroom	Year of Built	House Size (SQ Ft)	Price
City A	3	2002	2000	\$ 500,000.00
City B	2	1985	1500	\$ 300,000.00
City A	4	1997	2500	\$ 600,000.00
City B	3	2022	1800	\$ 350,000.00

Location	Bedroom	Year of Built	House Size (SQ Ft)	Price	Cost per SQ Ft
City A	3	2002	2000	\$ 500,000.00	\$ 250.00
City B	2	1985	1500	\$ 300,000.00	\$ 200.00
City A	4	1997	2500	\$ 600,000.00	\$ 240.00
City B	3	2022	1800	\$ 350,000.00	\$ 194.44

Location	Bedroom	Year of Built	House Age	House Size (SQ Ft)	Cost per SQ Ft	Price
City A	3	2002	22	2000	\$ 250.00	\$ 500,000.00
City B	2	1985	39	1500	\$ 200.00	\$ 300,000.00
City A	4	1997	27	2500	\$ 240.00	\$ 600,000.00
City B	3	2022	2	1800	\$ 194.44	\$ 350,000.00

Location	Bedroom	Year of Built	House Age	House Size (SQ Ft)	Cost per SQ Ft	Price
City A	3	2002	22	2000	\$ 250.00	\$ 500,000.00
City B	2	1985	39	1500	\$ 200.00	\$ 300,000.00
City A	4	1997	27	2500	\$ 240.00	\$ 600,000.00
City B	3	2022	2	1800	\$ 194.44	\$ 350,000.00

Scaling and Encoding

Min-Max scaling to transform the data to a range of 0 to 1.

$$\text{Scaled Value} = \frac{\text{Value} - \text{Min}}{\text{Max} - \text{Min}}$$

Result: [0, 0.33, 0.73, 1.0]

Encoding

Raw Data: ["Detached", "Semi-Detached", "Terraced"]

One-Hot Encoding →

- "Detached" becomes [1, 0, 0]
- "Semi-Detached" becomes [0, 1, 0]
- "Terraced" becomes [0, 0, 1]

Convert house types into binary vectors for model compatibility.

Location	Bedroom	Year of Built	House Size (SQ Ft)	Price
City A	3	2002	500	\$ 50,000.00
City B	2	1985	1500	\$ 300,000.00
City A	4	1997	2000	\$ 600,000.00
City B	3	2022	2500	\$ 800,000.00

Location	Bedroom	Year of Built	House Size (SQ Ft)	Price	House Type
City A	3	2002	500	\$ 50,000.00	Detached
City B	2	1985	1500	\$ 300,000.00	Semi-Detached
City A	4	1997	2000	\$ 600,000.00	Terraced
City B	3	2022	2500	\$ 800,000.00	Apartment

Exploratory Data Analysis

- Analyzing and visualizing data
- Understanding its structure, patterns and relationship
- Create plots like histogram, scatter plots, box plots to review the distribution and relationships
- Correlation Analysis
- Outlier Detection
- Purpose: understand and identify important features, spot and data issues and correct before model training

Location	House Size (SQ Ft)	Price	Cost per SQ Ft
City A	2000	\$ 500,000.00	\$ 250.00
City B	1500	\$ 300,000.00	\$ 200.00
City A	2500	\$ 600,000.00	\$ 240.00
City B	1800	\$ 400,000.00	\$ 222.22
City A	1700	\$ 15,000.00	\$ 8.82
City B	2500	\$ 670,000.00	\$ 268.00



Feature Engineering – Unstructured Data

- Unstructured Data – text, images, audio
- Can't be organized in predefined structure

Use Case – sentiment analysis of product review

- Dataset: Customer review
- Techniques:
 - Tokenization:
 - Perform TF-IDF
- Outcome: Convert the text data into numerical feature

Image Data:

- Edge detection – identify edges within images
- Color Histogram – Analyzing color distribution

★★★★★ Perfect!
Reviewed in the United States on October 26, 2024
Color: Cloud Pink | Style: Solo4 | Set: Without AppleCare+ | Verified Purchase
Love the sound. The charging speed is amazing and the battery lasts as promised. Super easy to pair with the phone. The quality is awesome!

Love the sound. The charging speed is amazing, and the battery lasts as promised. Super easy to pair with the phone. The quality is awesome!

Love the sound. The charging speed is amazing, and the battery lasts as promised.
Super easy to pair with the phone. The quality is awesome!

Copilot

perform TF-IDF on the paragraph: "Love the sound. The charging speed is amazing, and the battery lasts as promised. Super easy to pair with the phone. The quality is awesome!"

✓ Executing Task: amazing, and the battery lasts as promised. Super easy to pair with the phone. The quality is awesome!"

I performed TF-IDF (Term Frequency-Inverse Document Frequency) on the given paragraph. Here are the TF-IDF scores for each word:

- amazing: 0.1459
- and: 0.1459
- as: 0.1459
- awesome: 0.1459
- battery: 0.1459
- charging: 0.1459
- easy: 0.1459

AI Model Fit: Overfitting vs Underfitting

How well a model captures the underlying patterns in the data

Overfitting:

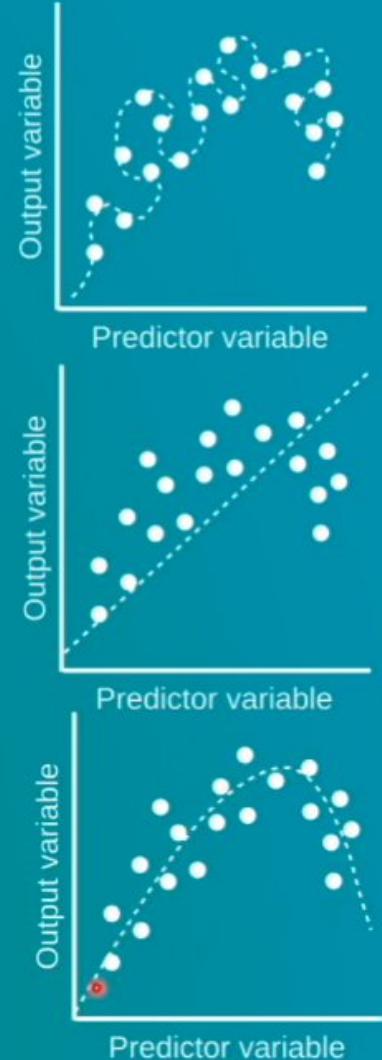
- Captures noise and outliers during training.
- Excellent performance on training data.
- Poor generalization performance on new unseen data.

Underfitting:

- Model is too simple to capture underlying data pattern.
- Poor performance on training data and new data.
- Maybe a problem with training data or Features.

Balanced:

- Strikes the right balance between complexity & simplicity.
- Captures the underline patterns without overfitting or underfitting.



Bias in AI Model

- Systematic errors in AI/ML model that lead to unfair outcomes.

Types of Bias:

- Selection Bias - training data is not representative to the population
- Measurement Bias - Inconsistent or incorrect data collection
- Algorithmic Bias - Model design, or flawed algorithm

Clinical trial of drug

Dataset – Young healthy

Missing Data – older adults
with pre-existing conditions

Biased Training Data

Ensure training data is
inclusive & represent the
entire population

Student Grading System

Mechanism – class-based
grading standards

Dataset – Previous student
grades

Inconsistency – Data
collection mechanism

Collect accurate & consistent
data

AI based hiring

Training data – successful
candidates from a specific gender or
ethnicity

Unfair Prediction – favors
similar candidates

Designing and training models
to avoid perpetuating existing
biases.

How to avoid bias and ensure fairness?

- **Diverse Training Data:** Ensure the training data is representative of all groups.
- **Bias Detection Tools:** Use tools like Amazon SageMaker Clarify to detect and mitigate bias.
- **Regular Audits:** Conduct regular audits of models to check for fairness.
- Increase the number of features.

ML Model Variance

- Model's sensitivity to small changes in the training data.

High Variance:

- Model predicts perfectly with training data.
- It fits the training data very closely
- Perform poorly on unseen data

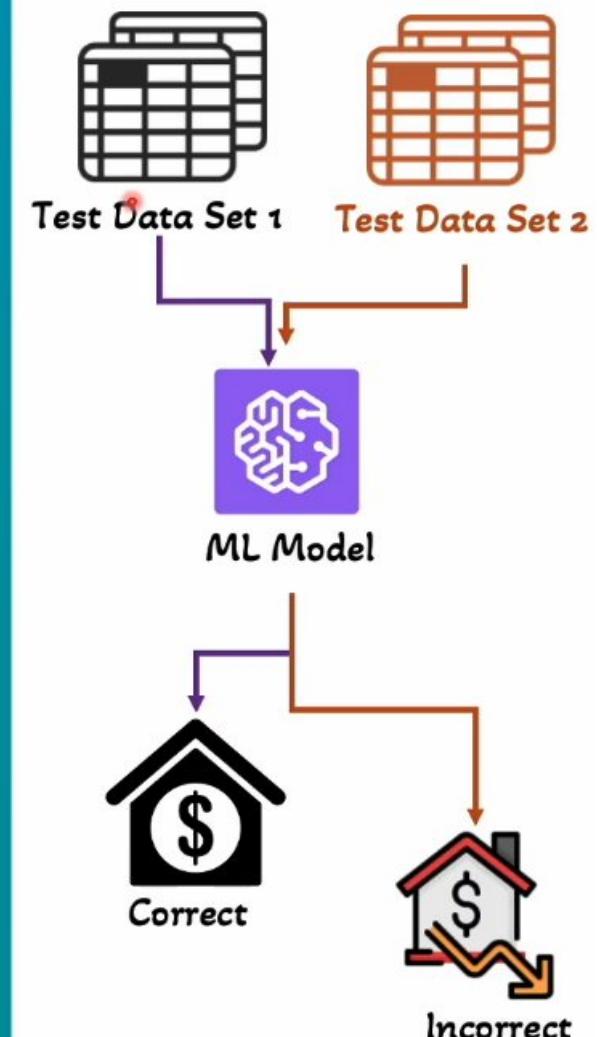
Overfitting

Low Variance:

- Model's predictions are stable across different training set.
- Might not fit the training data perfectly but performs better on new data.

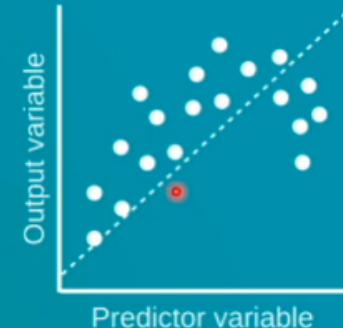
How to mitigate High Variance?

- Increase Training Data
- Cross-Validation
- Feature Selection
- Simplify the model



Balancing Bias and Variance

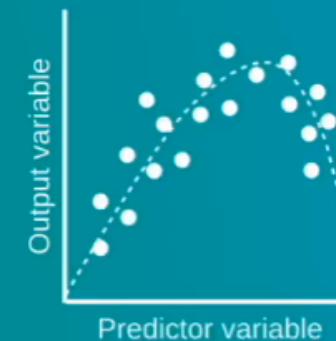
- **High Bias, Low Variance:**
 - Model doesn't capture underlying trend.
 - Model is too Simple
- **Low Bias, High Variance:**
 - It fits the training data very closely
 - Model captures noise and outlier in the training data
 - Model is too complex
- **Low Bias, Low Variance:**
 - Captures underline trends
 - Model well in test data and new data set
 - Model is perfectly balanced



Underfitting



Overfitting

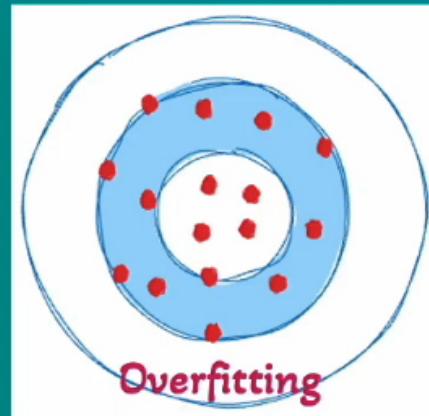


Perfect Balance

Balancing Bias and Variance

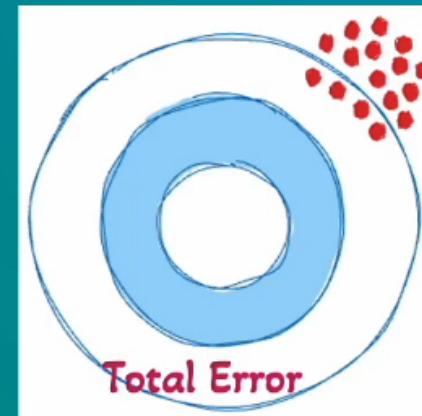
High Variance

Low Bias

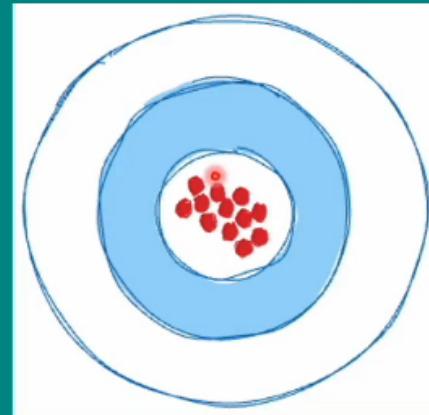


High Bias

Total Error

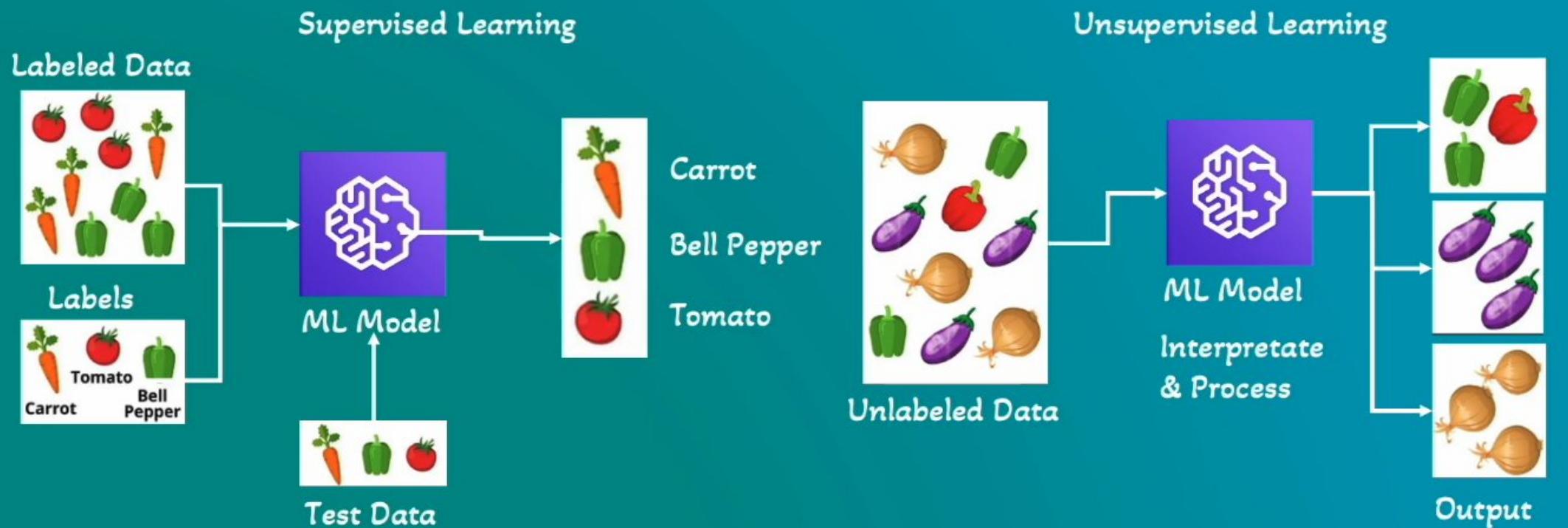


Low Variance



Underfitting





Healthcare and Genomics use cases requires **High-dimensional data**

- Labeling a large data set is time-consuming and expensive
- Human annotations requires domain expertise
- Unsupervised Learning – lack of Task-specific guidance (tumor as benign or malignant)
- No Ground Truth for evaluation
- Limited Generalization.



Self - Supervised Learning

Bridges the gap between unsupervised learning (no labels) and supervised learning (task-specific labels).

- Model learns from the data itself without manually labeled datasets.
- Model generates its own labels from the input data, leveraging inherent structures or relationships.

Key Advantages:

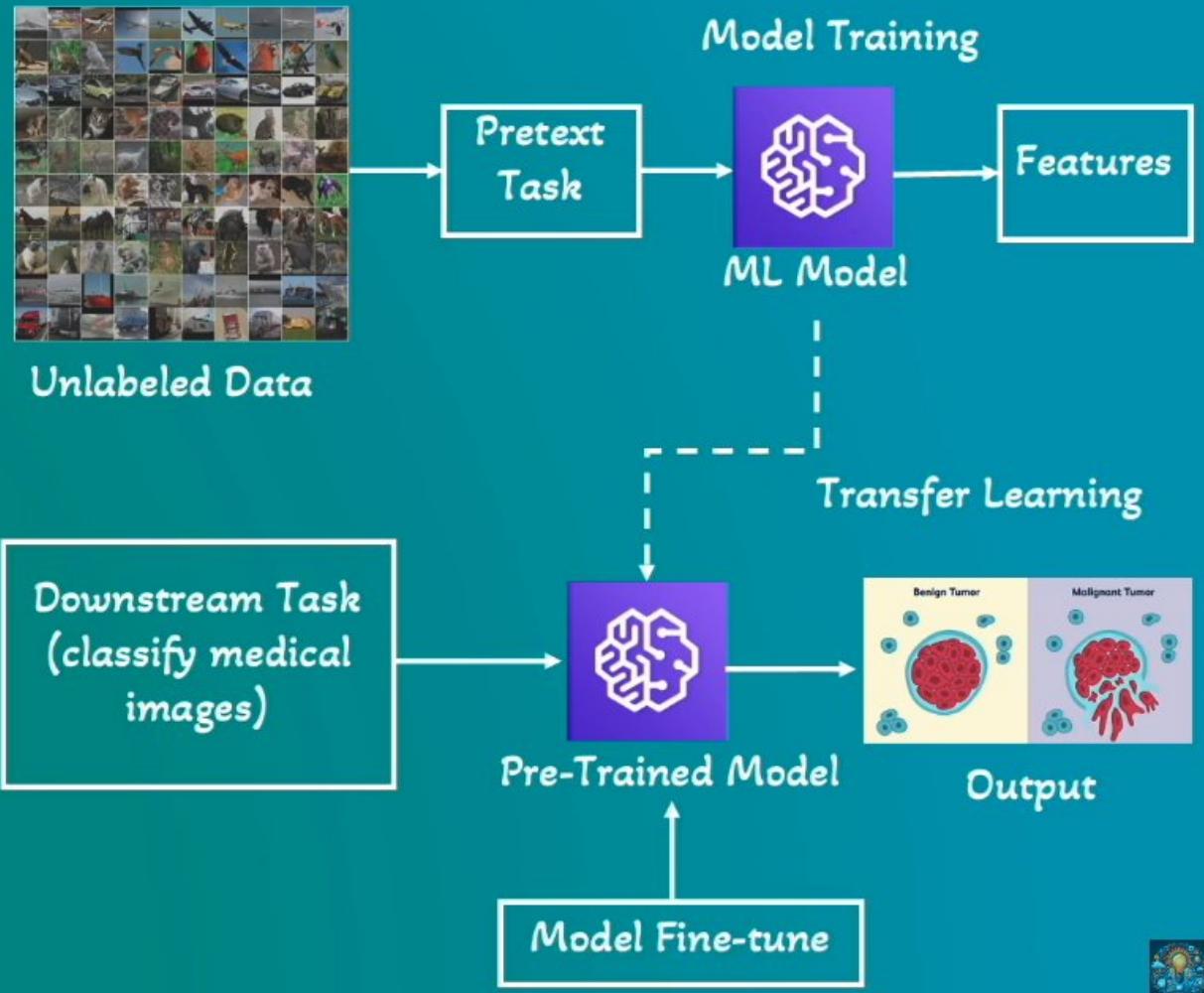
- More time and cost effective compared to supervised learning.
- Eliminates or reduces the need for manual data labeling.
- Reduces reliance on human-annotated datasets.

How Self - Supervised Learning Works?

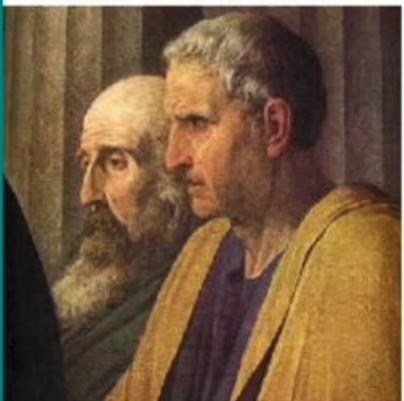
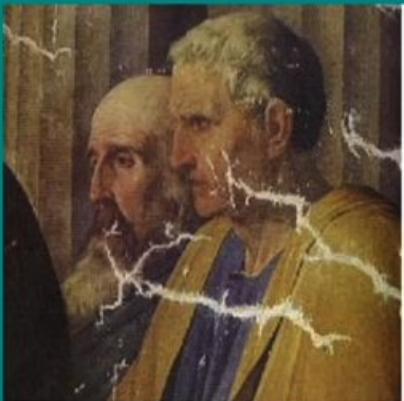
Step 1 : Pretext Task

- **Objective:** Learn meaningful representation from unlabeled data

Trained on an auxiliary task using pseudo-labels generated from the data.



Real-Life Examples:



Natural Language Processing

- BERT and GPT use self-supervised learning to predict missing words in sentences

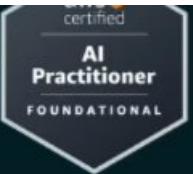
Speech Recognition:

- Facebook's Wav2Vec learns to recognize speech by predicting missing parts of audio signals

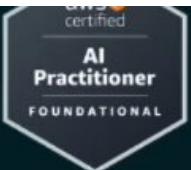
Self-Supervised Learning Use Cases

- **Medical Imaging:**
Can pre-train models on large unlabeled datasets of medical images (e.g., X-rays, MRIs) to learn general features.
- **Genomic Sequencing :**
Can predict missing parts of DNA sequences or infer gene functions from unlabeled genomic data.
- **Electronic Health Records (EHRs):**
Representations from unstructured HER data (e.g., patient notes, lab results) to predict patient outcomes or recommended treatments.
- **Drug Discovery:**
Analyze molecular structures and predict drug-target interactions.

Binary Classification



- Predicts one of two possible outcomes
- Answer questions like:
 - If an email is spam or not?
 - Will the customer churn or stay?
 - Is this transaction fraudulent or legitimate?
- Binary Output: a binary value (0 / 1, TRUE / FALSE, Positive / Negative).
- Real-world Use Cases:
 - Medical Diagnosis: Identifying diseases (Example – Tumor is malignant or benign)
 - Spam Detection: Filtering unwanted emails
 - Fraud Detection: Identifying fraudulent transactions
- Common Algorithms – Logistic Regression, Decision Trees, Support Vector Machine (SVMs) and Neural Networks
- Exam Tips – Please remember the use cases and the common algorithms.



Confusion Matrix

- Visualization tool used to evaluate the performance of a classification models
- Summarize the results of predictions by comparing against the actual, TRUE value

	Predicted : (Spam)	Predicted : (Not Spam)
Actual: (Spam)	True Positive	False Negative
Actual: (Not Spam)	False Positive	True Negative

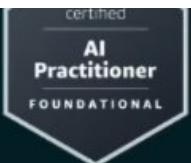
True Positive (TP): Correctly predicted Spam cases.

False Negative (FN): Incorrectly predicted Not Spam

False Positive (FP): Incorrectly predicted as Spam

True Negative (TN): Correctly predicted Not Spam

Allows you to see at a glance how well your model is performing across different classes.
Identifies potential areas for model improvement.



Performance Metrix

- Example: ML Model analyzed 100 emails.

N=100	Predicted : (Spam)	Predicted : (Not Spam)
Actual: (Spam)	TP = 50	FN = 10
Actual: (Not Spam)	FP = 5	TN = 35

Accuracy: Overall, how often is the classifier Correct?

$$\frac{\text{True Positive} + \text{True Negative}}{\text{Total}} = \frac{50+35}{100} = 0.85 \text{ or } 85\%$$

Precision: When it predicts TRUE, how often is it correct?

$$\frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})} = \frac{50}{(50+5)} = 0.91 \text{ or } 91\%$$

Recall: The proportion of actual Positive cases that a model correctly identifies

$$\frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})} = \frac{50}{(50+10)} = 0.83 \text{ or } 83\%$$

F1-Score: Harmonic mean of precision and recall.

$$\frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} = \frac{2*0.91*0.83}{(0.91+0.83)} = 0.86 \text{ or } 86\%$$

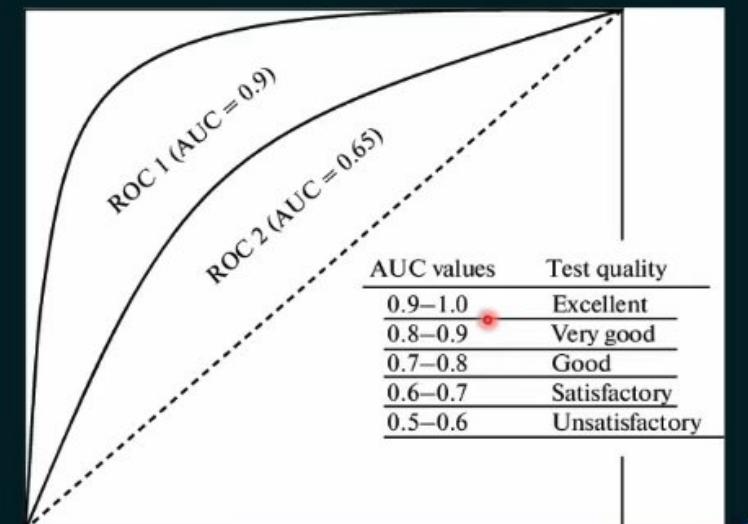


AUC-ROC Curve

- Area Under the Curve – Receiver Operating Characteristic
- Area Under the Receiver Operating Characteristic Curve.
- Used to evaluate the binary classification models
- Measures the ability to distinguish between classes
- ROC Curve: A plot of the TRUE Positive Rate (Sensitivity /Recall) against the FALSE Positive Rate.
- AUC: The area under the curve. It ranges from 0 to 1, with 0.5 representing random guessing and 1.0 representing perfect discrimination.

Interpretation:

A higher AUC indicates a model that can better distinguish between the two classes, meaning it's more likely to correctly classify a randomly selected positive instance higher than a negative one





Regression Metrics

- Used for models predict a continuous value (Example: House Price)

Mean Absolute Error (MAE)

- Represents the average magnitude of errors between predicted and actual values, without considering their direction.

Example: A model predicting the temperature for the next 3 days.

Actual temperatures → 15°C, 20°C, and 25°C

Model's Predictions → 16°C, 18°C, and 23°C respectively.

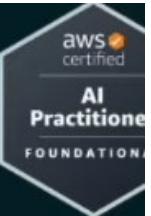
1. Calculate the absolute errors:

- Day 1: $|15 - 16| = 1$
- Day 2: $|20 - 18| = 2$
- Day 3: $|25 - 23| = 2$

2. Sum the absolute errors: $1 + 2 + 2 = 5$

3. Divide by the number of days: $5 / 3 = 1.67$

In simpler terms, it tells you, on average, how far off your predictions are from the true values.



Regression Metrics

Mean Absolute Percentage Error (MAPE)

- Same as MAE but shows error in Percentage terms.
- Example: if your MAPE is 5%, → Model's predictions are 95% accurate (on avg)
- Useful in comparing performance across different scales

Root Mean Squared Error (RMSE)

- Takes square of errors before averaging.
- Useful for penalizing large errors more.
- Use Case: When large errors are more harmful (predicting delivery times)

R-Squared (R^2)

- Measures how well model explains the variance in the target variable.
- $R^2 = 1 \rightarrow$ Perfect Prediction || $R^2 = 0 \rightarrow$ No Predictive Power
- Example: if $R^2 = 0.8$, then 80% of the variability in the prediction

Exam Tips: MAE, MAPE, RMSE – Measures the error (how “accurate” the ML model is)

R-Squared (R^2) – measures the variance



Business Metrics

User Satisfaction

- How happy are users with the model's output?.
- Example: A product recommendation system may not have the best accuracy, but users like the recommendations—this is a success.

Average Revenue Per user (ARPU)

- Measures how much income is generated per user on average.
- Example: Model's personalized recommendation boosts ARPU from \$5 to \$8 – Strong Business Impact

Conversion Rate

- Percentage of users who take a desired action (buy, sign-up, click).
- Useful for ML models in marketing or eCommerce industry

Cross-Domain Performance

- Evaluates models performance across different contexts or user group.
- Example: A voice bot trained in English should generalize well for similar task in Spanish

Efficiency

- Measures computational cost, time saved, or reduced manual effort.
- Example: The ML model reduces fraud investigation time by 40% - Measurable gain in Efficiency

Machine Learning - Anti Pattern

- When there's not enough data
 - ML models learn from patterns in data. No data = No learning.
- When the problem has fixed rules
 - Example: Calculating sales tax. There's a clear formula. No need for ML.
- When a simple program can do the job
 - If you can solve it with a simple if-else or business logic, do that instead.
- When it's just a straightforward probability calculation
 - Example: If you're calculating the probability of flipping a coin or rolling a die, the outcomes are well-defined and can be computed mathematically. No need for ML to estimate something we already know precisely.

Exam Tips: Use services like AWS Lambda for rule-based workflows, and ML only when prediction or pattern discovery is needed.