MLOps – Module 5

Designing

Machine Learning Systems

ML System Design

- Suppose your manager came to know about your rival e-commerce uses ML system to improve their customer service support.
- Instructed your team to build a ML system to speed-up your customer support.
  - **What would you do?**

- Problem statement
- Data pipeline
- Identify ML models
- Training

- Deployment
- monitoring.

Designing ML Systems (Module 5)

ETVL

Course Plan

- Introduction to ML system design
- Model Development & Offline Evaluation
- DataOps & Data Engineering Fundamentals
- LLM Architectures & Model Deployment
- Data Preparation & Feature Engineering
- Scalability, Monitoring & Security in ML Systems

1. Designing Machine Learning Systems by Chip Huyen, 2022 O'Reilly Media, Inc. ISBN: 9781098107963
2. Fundamentals of Data Engineering by Joe Reis & Matt Housley, 2022

Designing ML Systems

Sashi@iisc.ac.in

Outcome of Module 5

By the end of this module, learners will:

- Be able to design ML systems from problem formulation to production.
- Understand data pipelines, feature engineering, and ML deployment.
- Learn how to evaluate, monitor, and optimize ML models.
- Gain exposure to cutting-edge topics like LLM architectures, ModelOps, and SecOps.

Practical MLOps (Module 6)

Course Plan

- Introduction to MLOps & Version Control
- CI/CD for ML Pipelines
- Data Version Control (DVC) & DataOps for ML
- Experiment Tracking & Model Versioning
- Automated Model Training Pipelines
- ML Deployment, Monitoring & AIOps

1. Designing Machine Learning Systems by Chip Huyen, 2022 O'Reilly Media, Inc. ISBN: 9781098107963

Practical MLOps

Outcome of Module 6

By the end of this module, learners will:

- Understand MLOps & Version Control – Learn MLOps principles, version control with Git, and set up CI/CD pipelines.
- Implement CI/CD for ML – Automate ML workflows using GitHub Actions, Jenkins, and model registries.
- Manage Data & Versioning – Use DVC, LakeFS, and Pachyderm for dataset tracking and automation.
- Track Experiments & Model Versions – Utilize MLflow, ClearML, and Weights & Biases for reproducible ML experiments.
- Automate Model Training – Build end-to-end ML pipelines with Kubeflow, Apache Airflow, and AutoML tools.
- Deploy & Monitor ML Models – Deploy models as APIs, implement AIOps for monitoring, and detect drift using Evidently AI, DeepChecks.
- Ensure Reliability & Scalability – Set up incident response strategies, automated alerting, and best practices for production ML.

Parallel Computer Architecture and Programming Models (Module 7)

Course Plan

- Introduction to Parallel Computing & Architectures
- Memory Models & Multi-threading
- GPU Programming for ML & LLMs
- Distributed Training Strategies for LLMs
- Performance Optimization & High-Performance Computing (HPC)
- Emerging Trends & Hands-on Parallel Computing

Outcome of Module 7

By the end of this module, learners will be able to

- design, implement, and optimize parallel computing strategies for machine learning (ML) workloads, including distributed training of large-scale models and LLMs, utilizing modern parallel programming frameworks and hardware accelerators.

Machine Learning at Scale (Module 8)

Course Plan

- Challenges & Fundamentals of Scaling ML
- Multi-GPU & Multi-Node Training
- Distributed Computing & ML Orchestration
- Federated Learning & Edge ML
- Large-Scale Deployment & Serving of LLMs
- End-to-End Scalable ML Pipeline Implementation

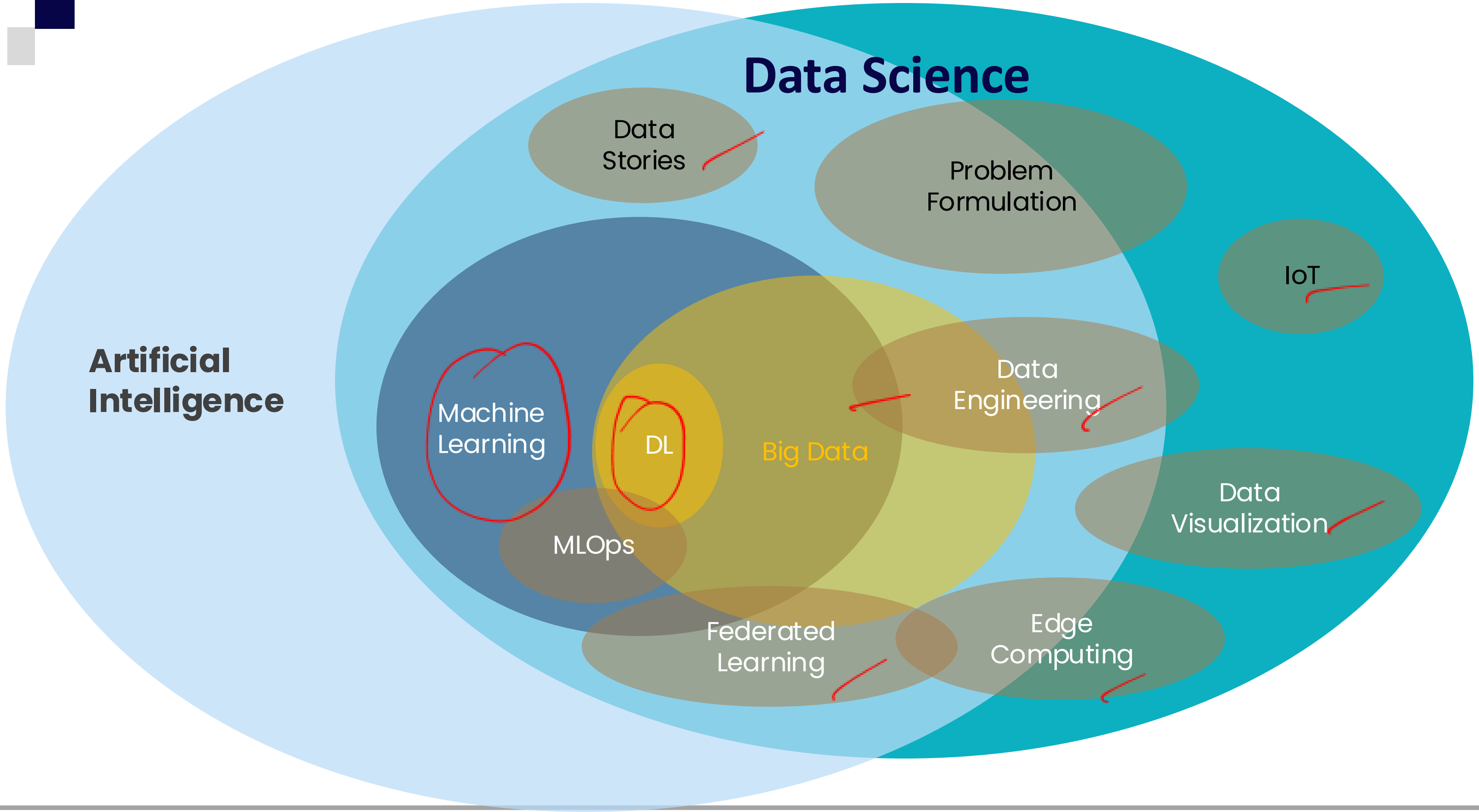
Outcome of Module 8

By the end of this module, learners will be able to

- design, train, deploy, and scale large-scale ML systems, including LLMs, using distributed computing, cloud infrastructure, and advanced model optimization techniques. They will gain hands-on experience with tools like Apache Spark, Ray, Kubernetes, and TensorRT for large-scale deployment.

ML Systems

Why Now and Challenges


Data Science
Data Stories
Problem Formulation
IoT
Artificial Intelligence
Machine Learning
DL
Big Data
Data Engineering
Data Visualization
MLOps
Federated Learning
Edge Computing

ML Systems: Why now and Challenges

Challenges

- MLOps is a methodology to standardize and streamline ML lifecycle management
- For traditional entities, deployment of (several) ML models in a production environment are relatively new
- Needs an understanding ML models and their dependencies at an entity-wide level
- With decision automation, ML lifecycle becomes critical
- The reality of ML lifecycle in an entity is much more complex, in terms of needs and tooling

ML Systems: Why now and Challenges

Challenges

- Not only is data constantly changing, but business needs shift as well
- Predictions need to be continuously monitored to ensure that the results align with expectations and meet the original goal
- ML lifecycle involves several people (Business, data engineer, data scientist, ML engineer, customer, etc.), and not everyone speaks the same language
- Data scientist are not necessarily ML engineers

ML Systems: Why now and Challenges

DevOps Vs. MLOps

- If MLOps sounds familiar, that's because it is derived from DevOps methodology
  - Robust automation and coordination between teams
  - Collaboration and increased communication between teams
  - The end-to-end life cycle (build, test, release)
- Deploying a software is fundamentally different than deploying ML models into production
  - ML modes are not just code but also data and ML methods
  - Data changes continuously, ML models are non-static

ML Systems: Why now and Challenges

Risks associated with ML models in production

- ML models' risks vary widely
  - Unavailable for a given period of time
  - Returns a bad prediction for a given sample
  - Accuracy and fairness decrease over time
  - Skills (workforce) necessary to maintain the model are lost
- Risks are usually larger for models deployed and used outside of the entity
- Implications of wrong predictions
  - For E.g., minimal on OTT platform, whereas sever on Defense
- Accountability and certification

ML Systems: Why now and Challenges

Summary

- MLOps is a critical part of transparent strategies for ML model lifecycle
- Teams that attempt to deploy ML systems without proper MLOps practices in place will face issues:
  - Model quality, negative impact on business, accountability, reproducibility, responsible AI, etc.

ML Systems

The Team

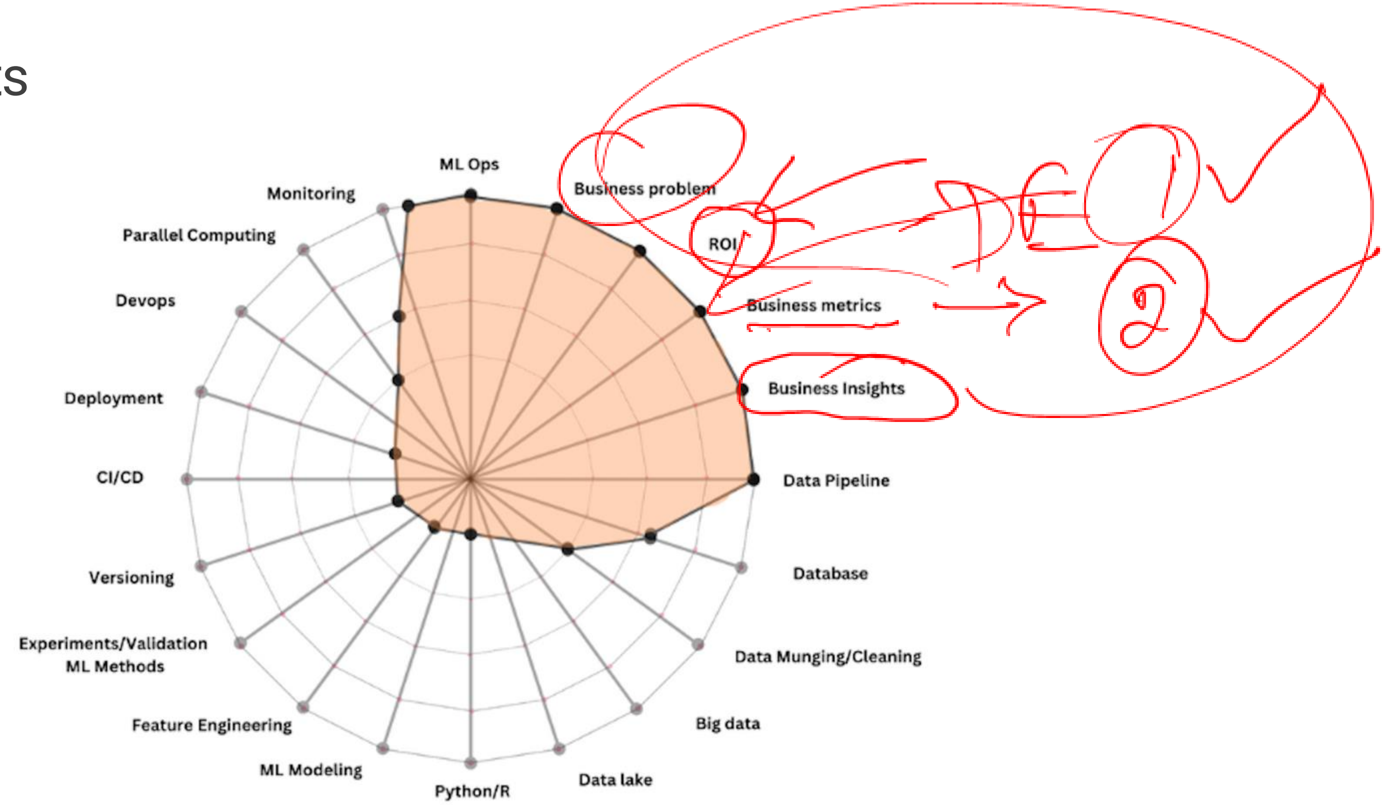
ML Systems: The Team

Domain/Subject Matter Experts

- ML system lifecycle starts and ends with SMEs
- Needs deep understanding of business problems that needs to be addressed using ML system
- Should come with clearly defined goals, business questions and KPI that they want to achieve
- Rather than starting with well-defined ML problems, start with less defined goals and make it well-defined iteratively in collaboration with data scientists
- SMEs are needed for business outcome, to gain traction and budget
- Business Decision Modeling: Create a business blueprint integrating ML models with business rules

ML Systems: The Team

Doamin Experts



ML Systems: The Team

Data Engineer

$$\boxed{Y}(x) = w\underline{x}^{c} + b$$

Target

Input feature

Data Capture

transforms data into a useful format for analysis

Collection, cleaning

Software solutions

advanced programming and system creation

creating software solutions around big data

Big Data

creating software solutions around big data

create data pipelines

understand the right tool

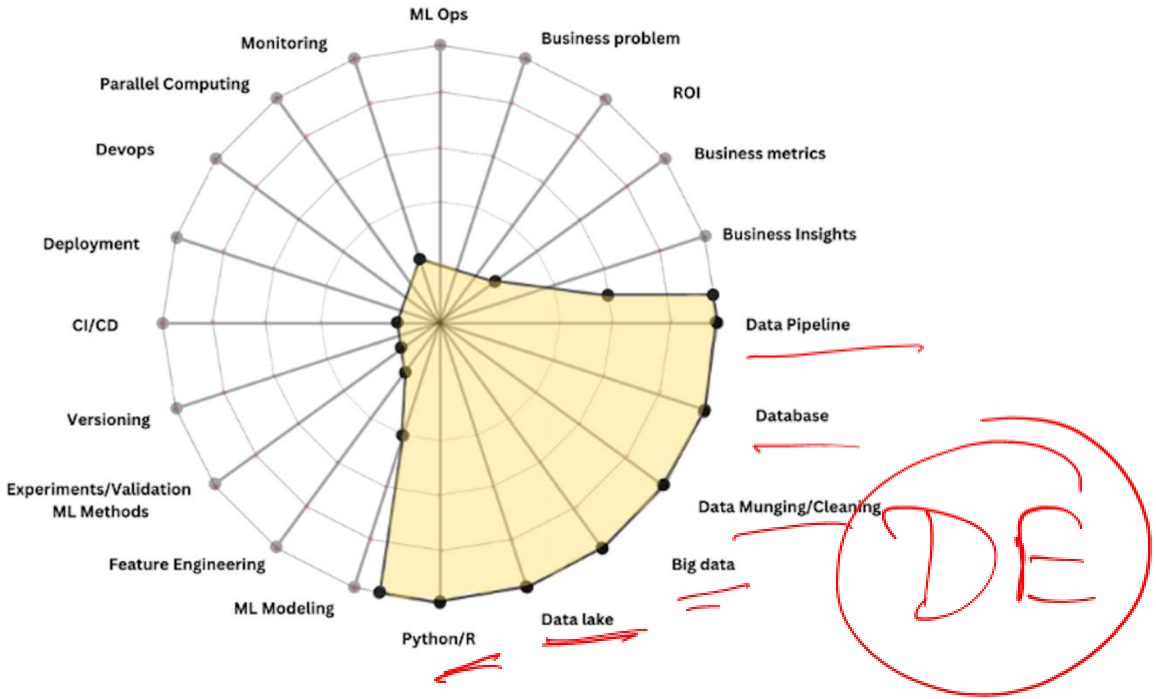
Skills

MySQL, MongoDB, Scala, Dask, Apache Spark, Hadoop

Hadoop, MapReduce, Hive, Pig, Data streaming, NoSQL, SQL,

ML Systems: The Team

Data Engineer



ML Systems: The Team

Data Scientist

ML Expert

Data analyst who applies ML/AI

picked up programming out of necessity

Domain Expert

Interact with business side

understand the domain enough to make insights

Data Story

Verbally and visually communicate complex results in a way that the business can understand and act on them

Skills

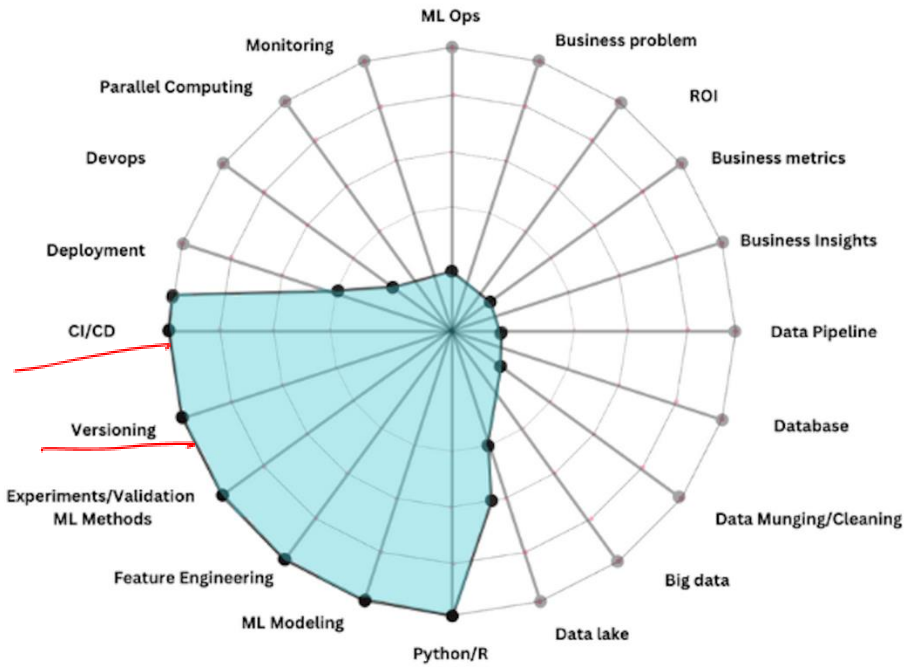
Math, statistics and basic programming to analyse data

R, create ML models

Jupyter, TensorFlow,

ML Systems: The Team

Data Scientist



DS

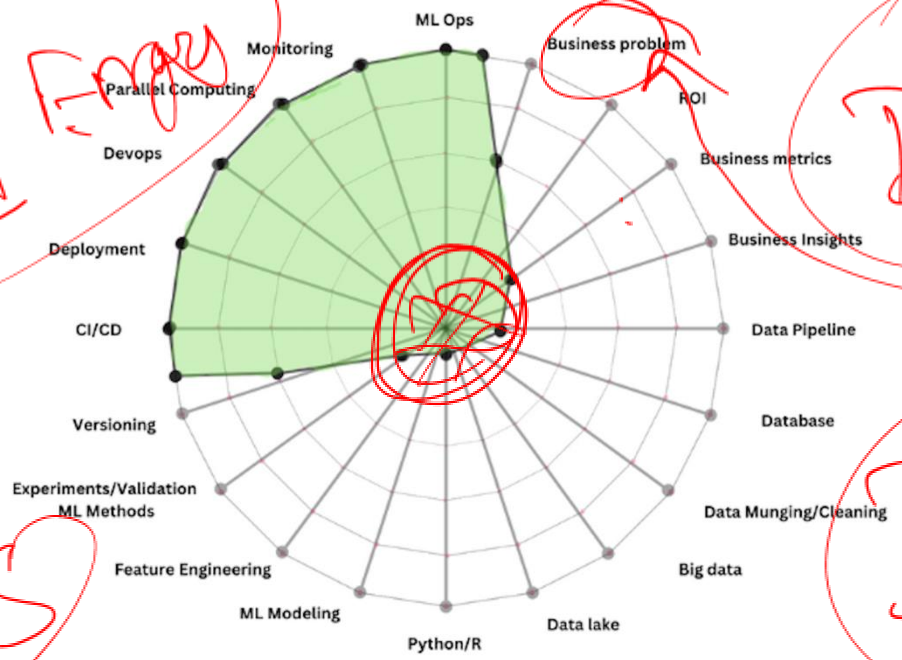
ML Systems: The Team

ML Engineer

- Need a systematic & efficient approach to be build ML models
- Needs to apply DevOps' best-practices to the emerging ML technologies
- Widespread adoption of ML models globally necessitates a sudden rise in demand for ML Engineers
- cross-trained enough to become proficient at both data engineering and data science
- primarily come from data engineering backgrounds
- sits at the crossroads of data science and data engineering, and has proficiency in both data engineering and data science

ML Systems: The Team

ML Engineer



AI Engg

DE

Users

DS

DE

ML Systems: The Team

Summary

MLOps isn't just for Data Scientists

- A diverse group of experts, Domain experts, Data engineers, Data scientists, AI engineers, across the entity need to work together to build deployable AI systems

Introduction to ML System Design

Framing ML Problems & Choosing the Right ML Approach

ML System Design: Introduction

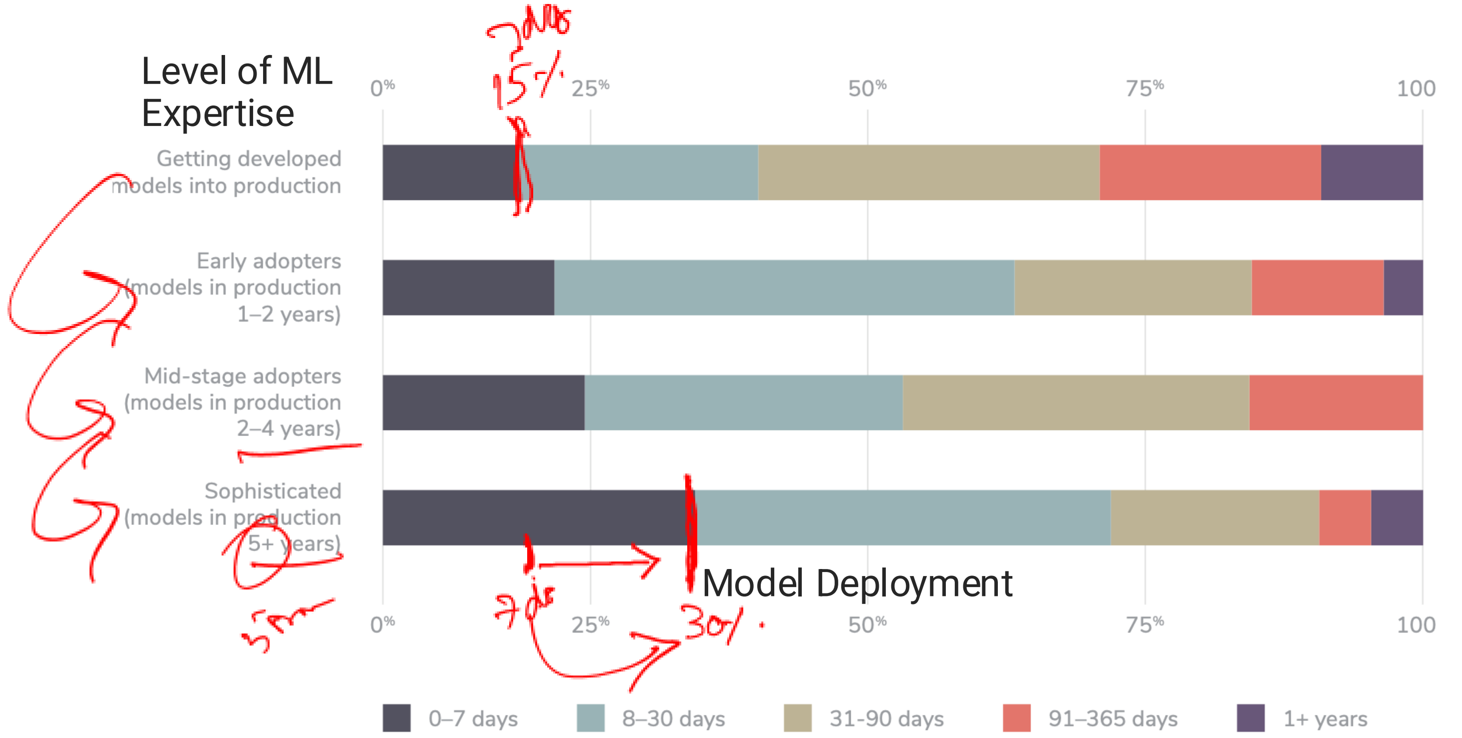
Objectives

- Learn how to frame ML problems correctly
- Understand how to align business & technical goals
- Identify the right ML approach for a given problem

/method

ML System Design: Introduction

How long it takes for an entity to bring ML systems in production?



Source: Algorithmia

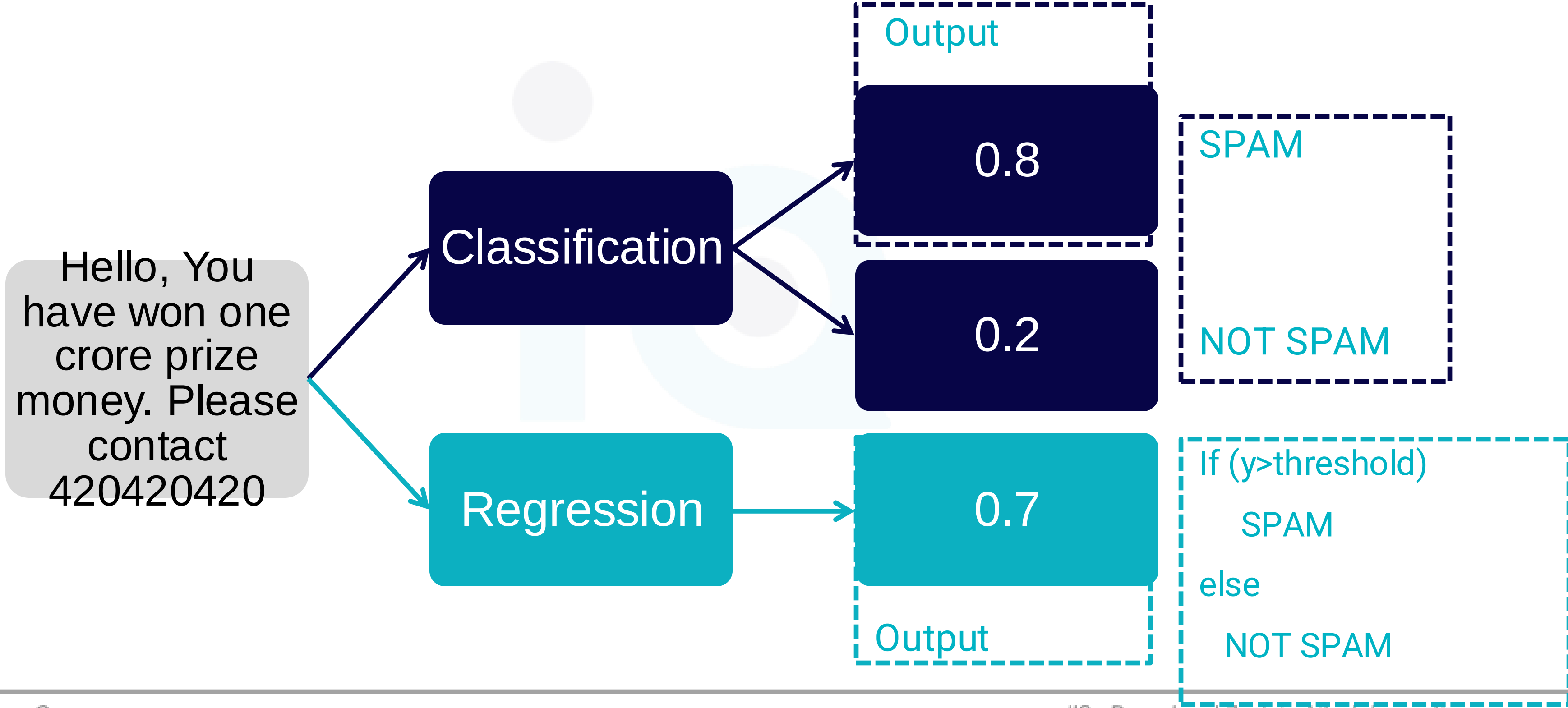
ML System Design: Introduction

Why ML System Design Matters?

- ML models fail not just because of bad algorithms but due to poor problem framing
- Example: Predicting
  - Customer churn – What are we predicting? Binary classification? Regression?
  - Anomaly detection – Should we use supervised learning or unsupervised learning?
- 💡 Key Question: Are we asking the right question for ML to solve?

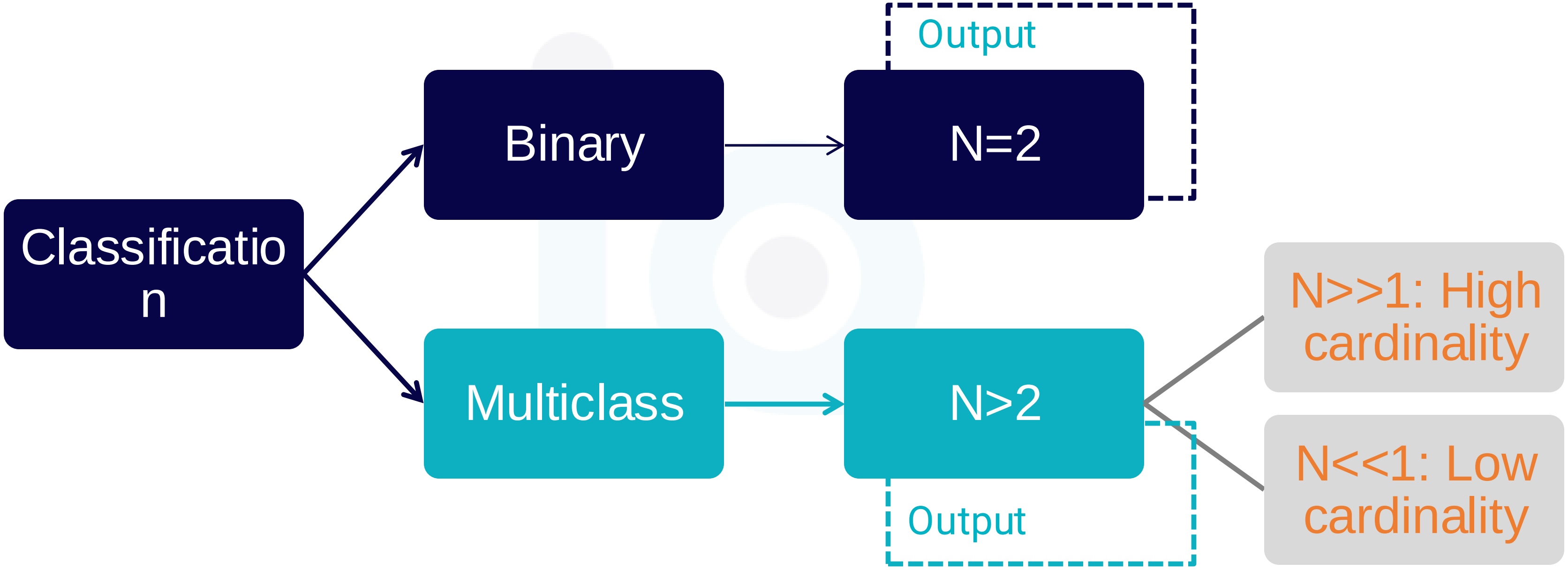
ML System Design: Introduction

Types of ML Problems: Classification Vs. Regression



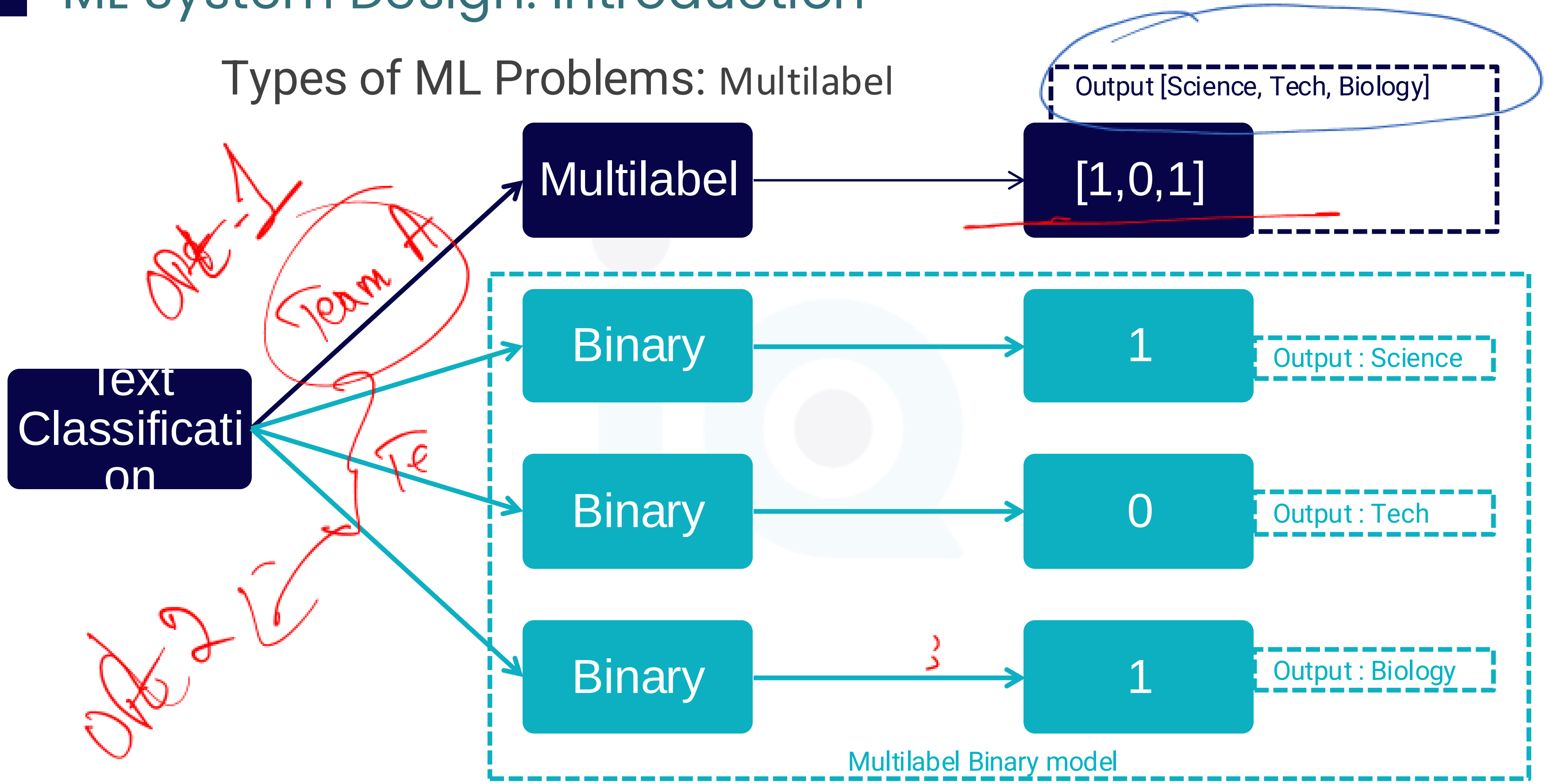
ML System Design: Introduction

Types of ML Problems: Binary Vs. Multiclass



ML System Design: Introduction

Types of ML Problems: Multilabel



ML System Design: Introduction

How to frame a ML Problem

Suppose your AI team needs to predict what product an E-Commerce customer will buy next among the list of products:

100 Product

- Available Dataset
  - User's features (Age, Gender, previous purchase history, frequency, etc.)
  - Environment (Time, location, region, etc.)
  - Features of the products
  - Sales history of the products

150

What ML system do you recommend as a team lead?

Regression or Classification?

ML System Design: Introduction

Framing ML Problems - Key Steps

- Define the Objective (Business vs. Technical)
- Identify the Input & Output Variables
- Select the ML Approach (Supervised, Unsupervised, RL)
- Evaluate Success Metrics & Constraints
- Plan for Model Deployment & Monitoring

ML System Design: Introduction

Defining Business & Technical Objectives

Business Objectives → ML Problem → ML Model Type

Business Goal	ML Problem Type	Example
Increase sales	Recommendation System	Amazon's "People Also Bought"
Reduce fraud	Anomaly Detection	Credit Card Fraud Detection
Automate processes	NLP & Computer Vision	Chatbots, Image Classification
Improve customer retention	Predictive Analytics	Customer Churn Prediction

Example:

📌 Business Goal: Increase Customer Engagement

✅ ML Solution: Build a personalized recommendation engine

ML System Design: Introduction

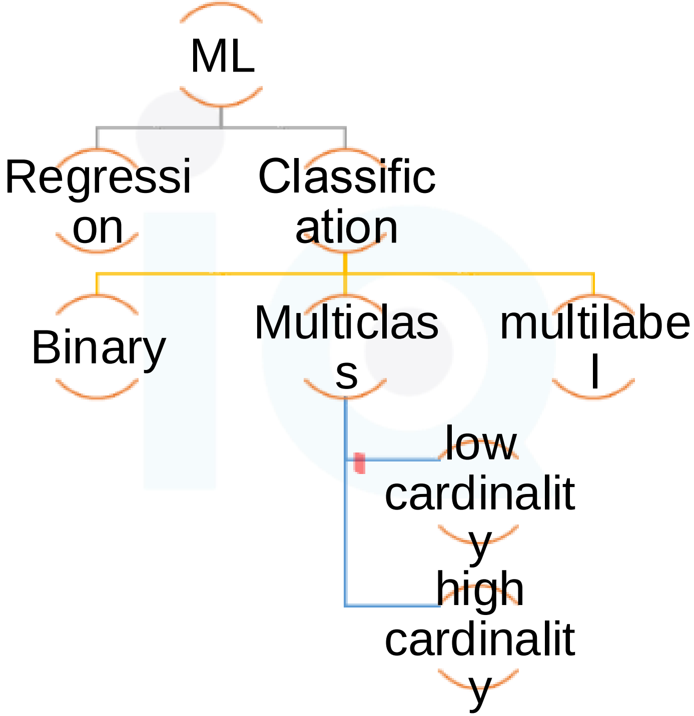
Identifying Input & Output Variables

- Example 1:
  - Customer Churn Prediction
    - Input: Customer history (age, purchase history, engagement level)
    - Output: Will they churn? (Yes/No)
- Example 2:
  - Price Prediction for Houses
    - Input: Size, Location, Number of Rooms
    - Output: House Price (continuous value)

```python
import pandas as pd

# Load dataset
data = pd.read_csv("customer_data.csv")

# Define input (X) and output (y)
X = data[['age', 'purchase_history', 'engagement_score']]
y = data['churn']  # Binary classification (0 = No churn, 1 = Churn)

print(X.head(), y.head())
```

ML System Design: Introduction

Choosing the Right ML Approach



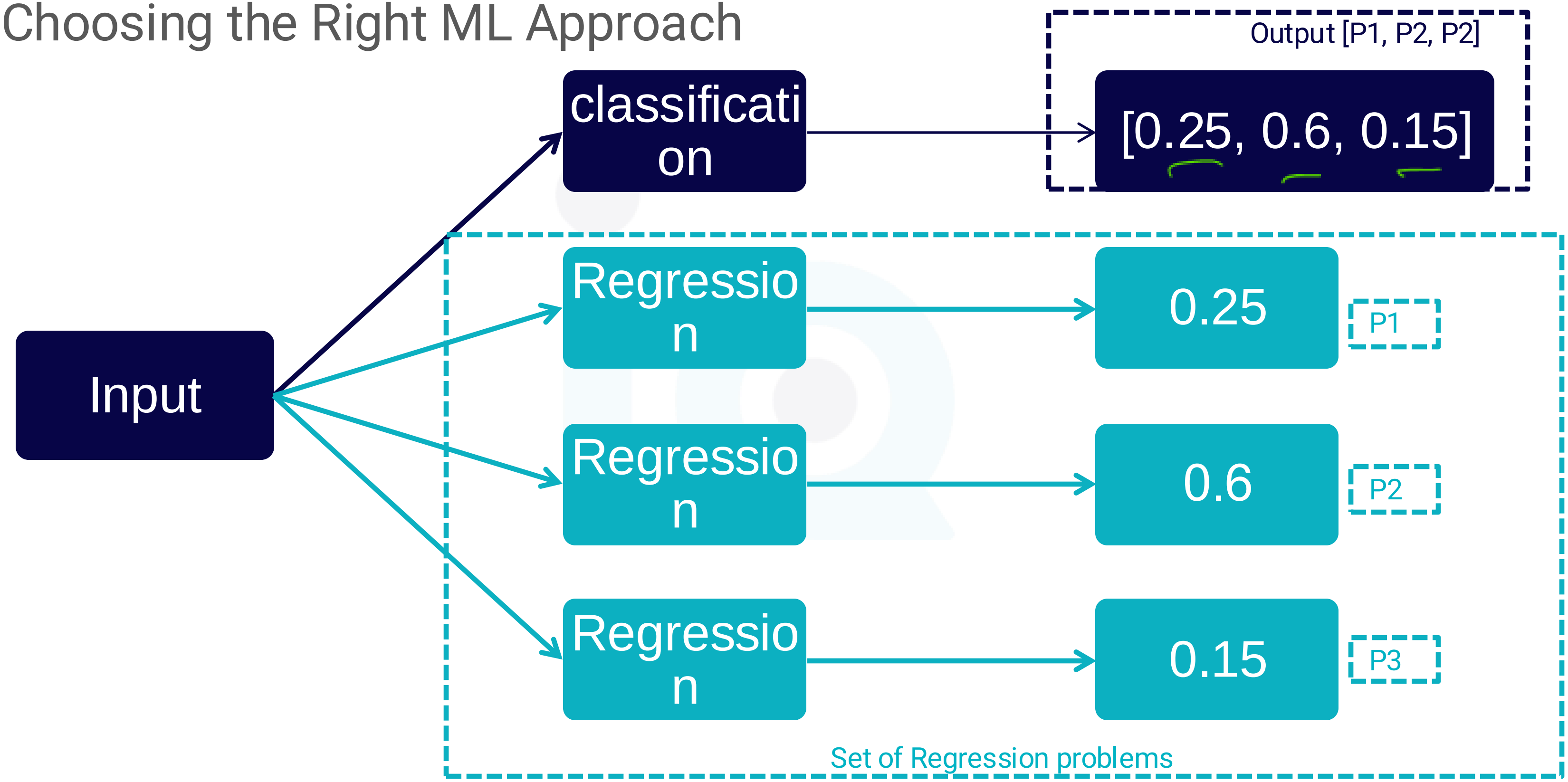
ML System Design: Introduction

Supervised vs. Unsupervised vs. Reinforcement Learning

Learning Type	Definition	Example Use Case
Supervised Learning	Learns from labeled data (X → y)	Spam classification, fraud detection
Unsupervised Learning	Finds patterns in unlabeled data	Customer segmentation, anomaly detection
Reinforcement Learning	Learns by interacting with the environment	Self-driving cars, game-playing AI

ML System Design: Introduction

Choosing the Right ML Approach



ML System Design: Introduction

Summary

- ML System is NOT a magic solution to all problems
- Every project starts with why this projects needs ML
- Build ML system with business metrics, accuracy of ML models is not enough for the success
- Be clear on what is a good ML system, and understand its requirements
- Building ML system isn't a one-off task but an iterative process
- Balance the Mind and Data