Generative AI Bootcamp

Introductions

- Name
- Role (in one line – what you do)
- Technology background (Java / Data Science/ ML / front end etc.)
- Any experience with Python?
- Your interest in this course?

Program flow

- Day 1: Data Science, Machine Learning, NLP building blocks
- Day 2: Deep Learing, Transformers, Generative AI, Gen AI Apps, Prompt Engineering
- Day 3: RAG, Knowledge Graph, Fine Tuning
- Day 4: Agents, Agentic AI, Evaluating Gen AI Applications
- Day 5: LLMOps, Gen AI adoption, governance, risk, compliance

About me

Vijay Agrawal

- Over 30 years of experience in IT in senior positions with 14+ years in Silicon Valley, USA.
  Worked at several fortune 500 companies in software product development, design and consultancy, innovation leadership roles

- Expertise in GenerativeAI, Machine Learning, Data Science, Data Engineering, Automation.
  - Partner, Data & AI practice at Baker Tilly, India
  - VP Innovation & Business Enablement at Kroll
  - Sr Consultant, Cloud and Data Engineering at Red Hat, India

- Past 1.5 years focused on training and consulting in Data/AI, Gen AI
  - Delivered 1000+ hours of workshops on AI and Generative AI across the spectrum from leadership/strategy to execution and implementation of Gen AI solutions

https://linkedin.com/in/agrawalvijay

What is AI?

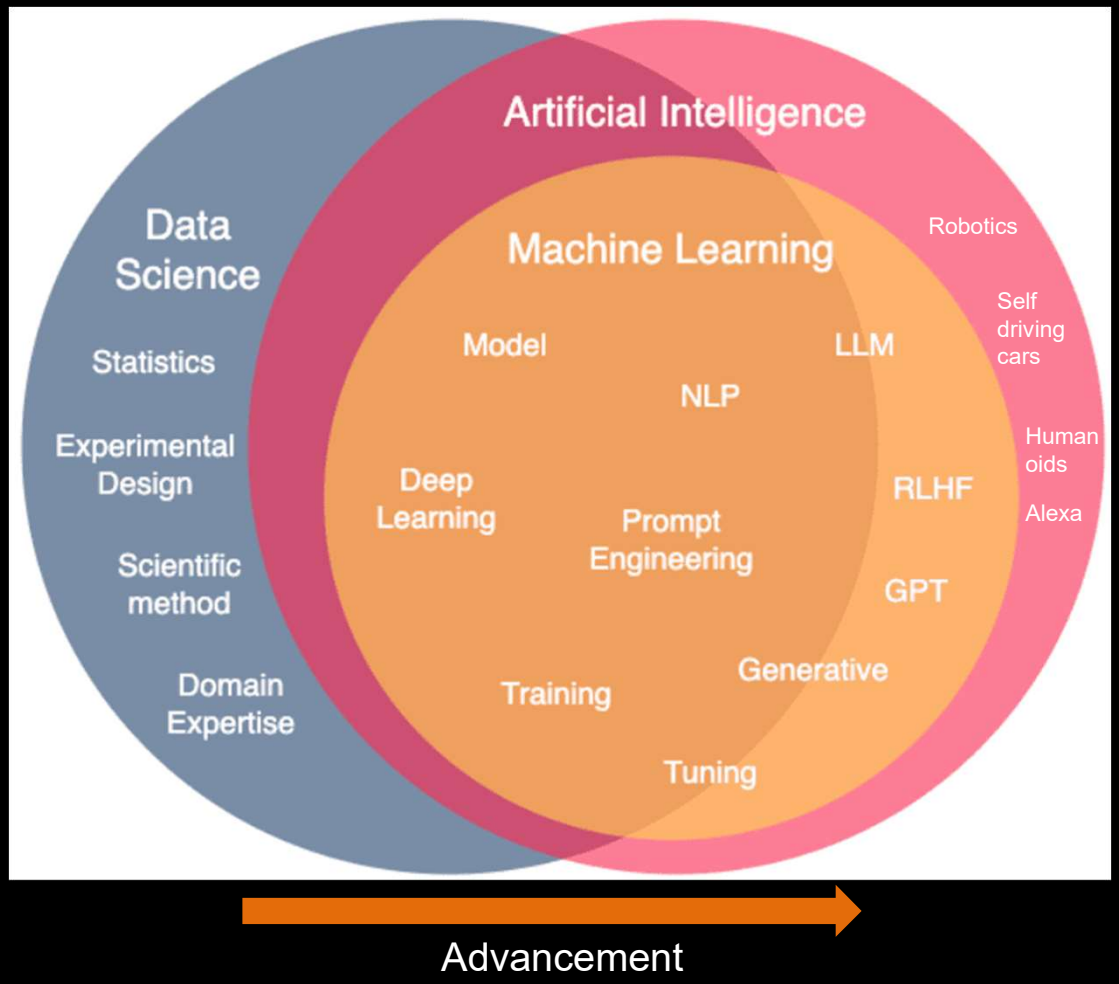
Artificial Intelligence (AI) is the field of computer science focused on creating machines or systems that can perform tasks that would typically require **human intelligence**.

These tasks often include:

- Reasoning
- Learning
- Problem-solving
- Understanding natural language
- Recognizing patterns
- Making decisions
- Generating new stuff based on it’s learning

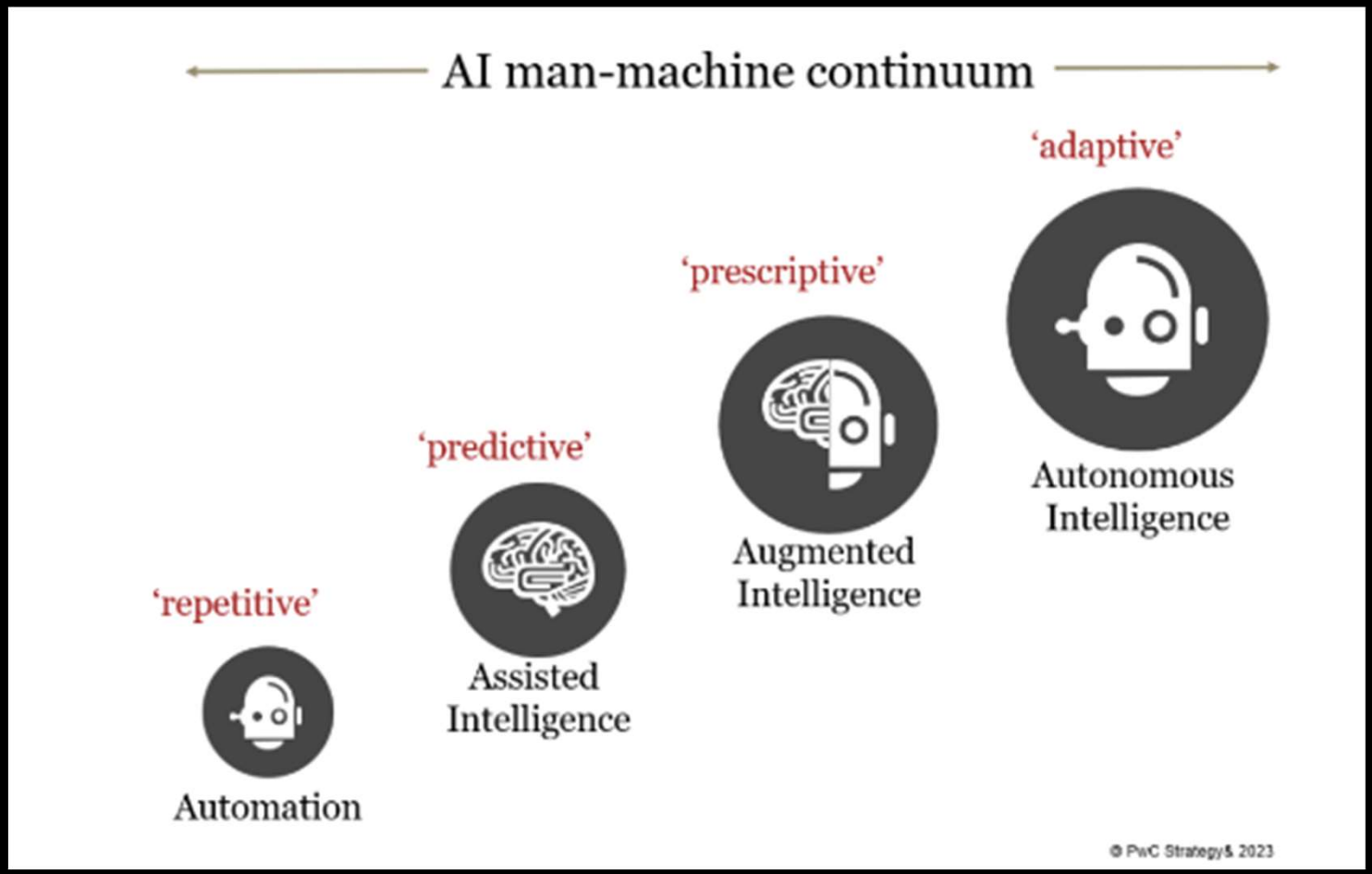
AI systems can adapt and improve over time, especially those utilizing advanced techniques like machine learning (ML) and deep learning.

Data science, Machine Learning, Deep Learning, AI

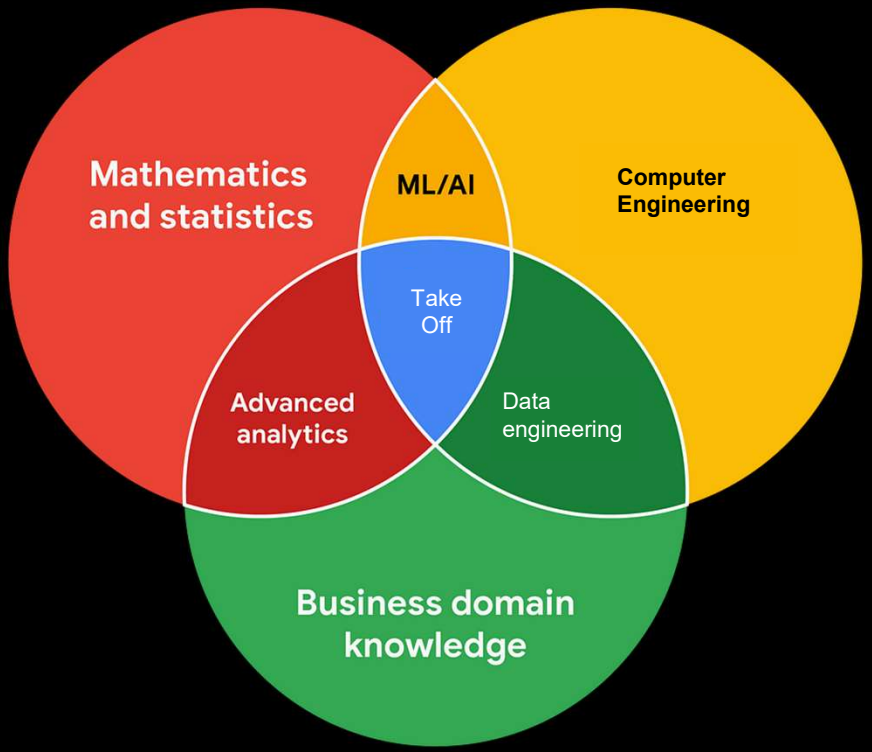


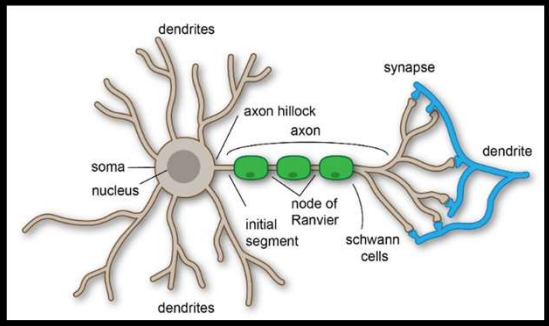
Data Science, Machine Learning, AI overlap and complement each other while still having their own distinct capabilities

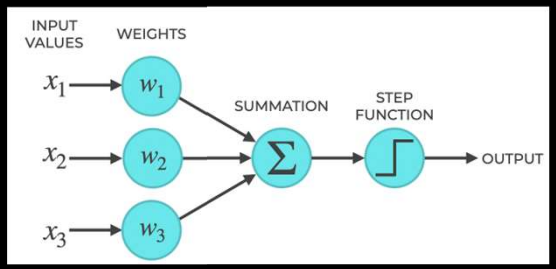
Evolution of AI

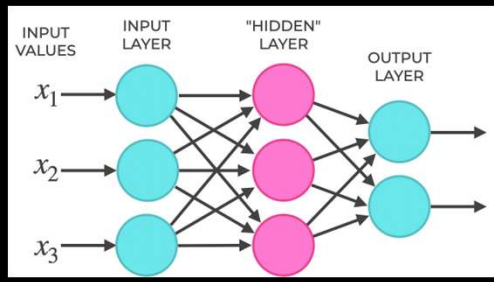


End goal: Mimic human brain

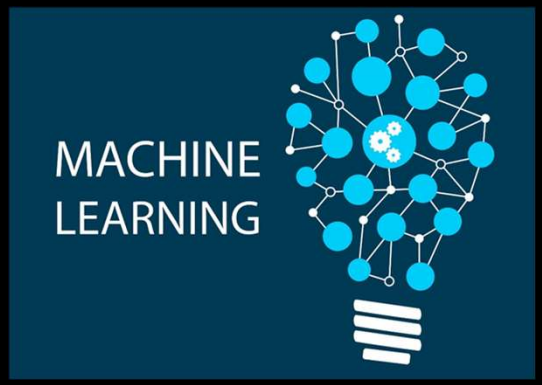






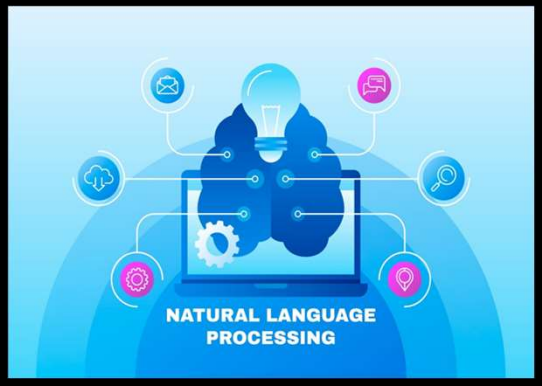


Key building blocks of modern AI



Machine Learning

Deep Learning



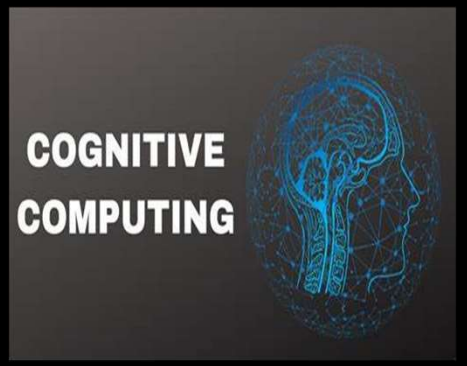
Natural Language Processing

Computer Vision

Cognitive computing

Cognitive computing represents a broader system that leverages both machine learning and deep learning to achieve human-like intelligence and decision-making capabilities.

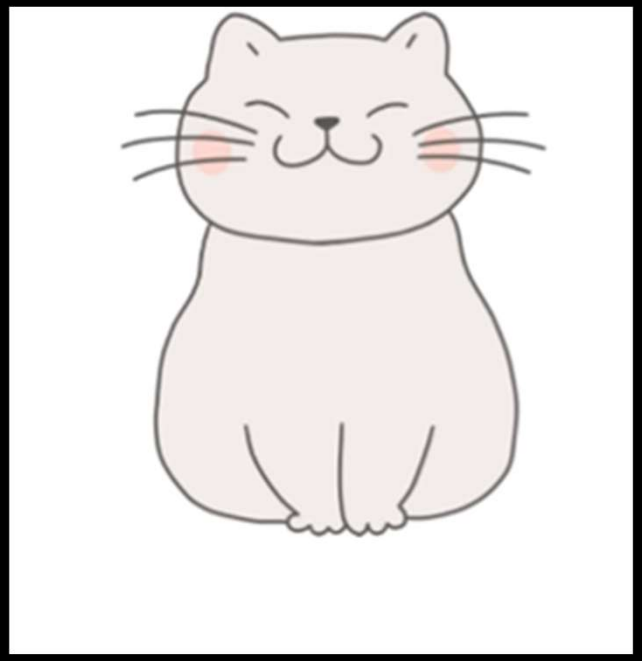
Feature	Machine Learning	Deep Learning	Cognitive Computing
Goal	Identify patterns, predict outcomes	Automate feature extraction, handle unstructured data	Simulate human thought, assist decision-making
Data	Structured	Structured and unstructured	Structured and unstructured
Scope	Predictive tasks	Complex recognition tasks	Decision support, reasoning
Technology	Algorithms (SVM, trees, etc.)	Neural networks	AI, ML, NLP, DL
Human Interaction	Limited	Limited	High



Cognitive Computing

Let’s take another problem

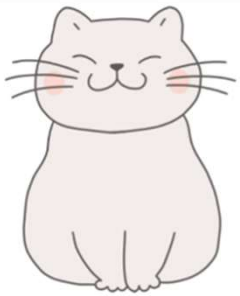
Is this a cat or a dog?



How does it learn to predict?

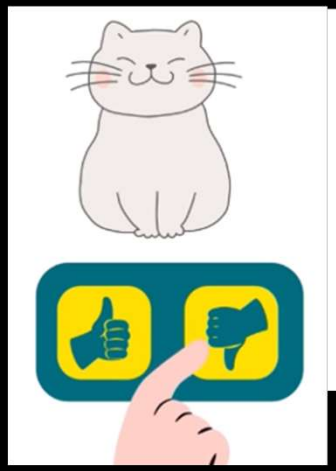
To Answer, "Is this a cat?"

Traditional Programming



Must hard code rules for what is a "cat"

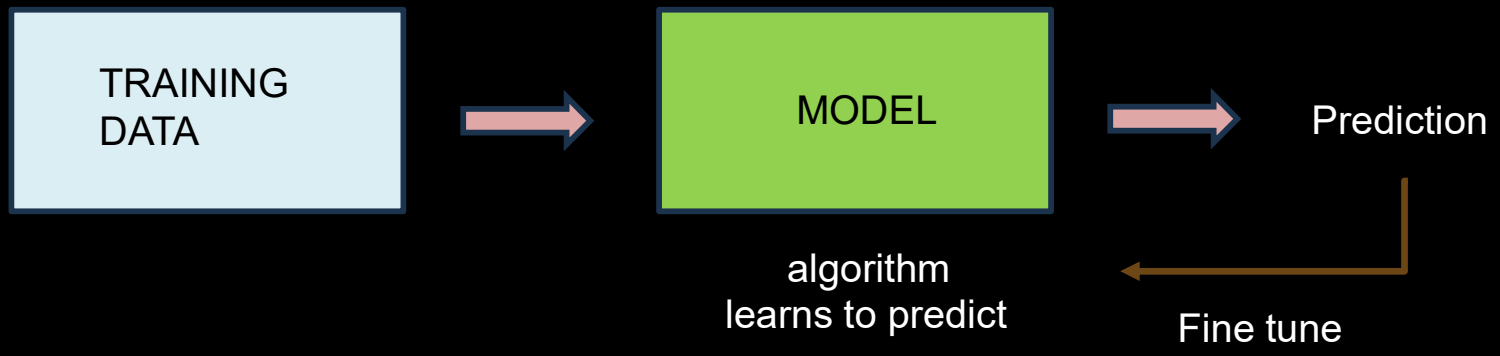
With Machine Learning



- Must give the network pictures of cats and dogs
- Makes a prediction based on training

Machine learning (ML) is defined as a discipline of artificial intelligence (AI) that provides machines the **ability to automatically learn** from data and past experiences to identify patterns and make predictions with minimal human intervention ,operating **without explicit programming**

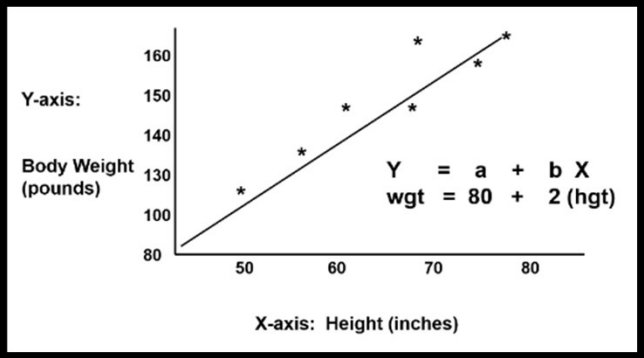
It all starts with the data



End to end – predict weight of a person based on height

1. Start with training dataset

2. Determine the model to use
   Linear regression in this case



y = mx + c

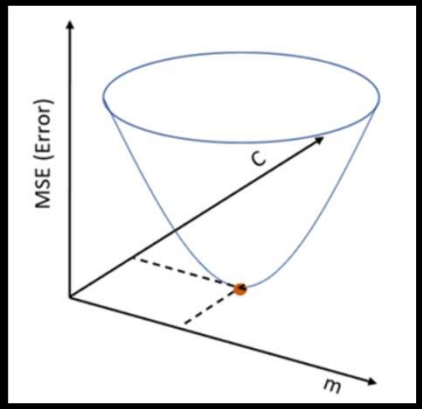
3. Measure error / accuracy

$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

Cost function (or loss function) measures the error.
Our goal is to minimize the cost function to find the best fit line

4. Tune the model - Minimise error (improve accuracy)

Gradient Descent is a mathematical function that can help compute the model parameters (m, c) that will minimize the loss function in minimum number of iterations.



Relies on derivatives and calculus

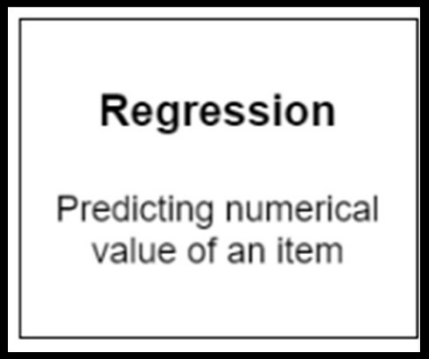
$$D_m = \frac{\partial(Cost\ Function)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$D_c = \frac{\partial(Cost\ Function)}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$m = m - LD_m$$

$$c = c - LDc$$

Types of ML problems



- Is it a cat?
- Is this a fraud transaction?
- Will the customer default on the loan?

- Binary Classification
- Multiclass Classification

- home price prediction
- stock price prediction
- weight prediction based on height

Regression is a method for understanding the relationship between independent variables or features and a dependent variable or outcome. Outcomes can then be predicted once the relationship between independent and dependent variables has been estimated.

Property price prediction

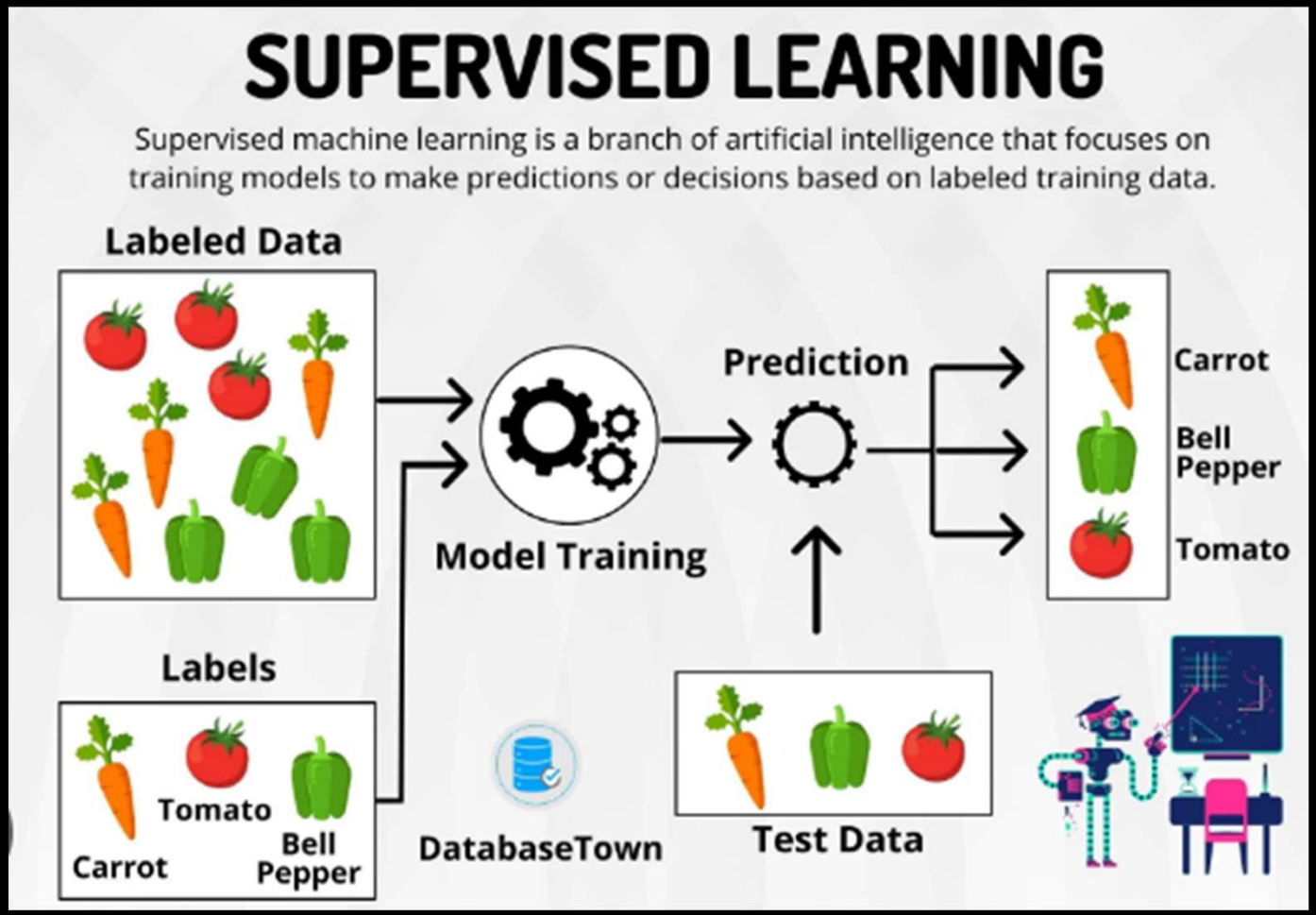
	tx_price	beds	baths	sqft	year_built	lot_size	property_type	exterior_walls	roo
0	295850	1	1	584	2013	0	Apartment / Condo / Townhouse	Wood Siding	Na
1	216500	1	1	612	1965	0	Apartment / Condo / Townhouse	Brick	Composition Shing
2	279900	1	1	615	1963	0	Apartment / Condo / Townhouse	Wood Siding	Na
3	379900	1	1	618	2000	33541	Apartment / Condo / Townhouse	Wood Siding	Na
4	340000	1	1	634	1992	0	Apartment / Condo / Townhouse	Brick	Na
5	265000	1	1	641	1947	0	Apartment / Condo / Townhouse	Brick	Na
6	240000	1	1	642	1944	0	Single-Family	Brick	Na
7	388100	1	1	650	2000	33541	Apartment / Condo / Townhouse	Wood Siding	Na
8	240000	1	1	660	1983	0	Apartment / Condo / Townhouse	Brick	Na
9	250000	1	1	664	1965	0	Apartment / Condo / Townhouse	Brick	Na

Target Variable
/ Dependent Variable

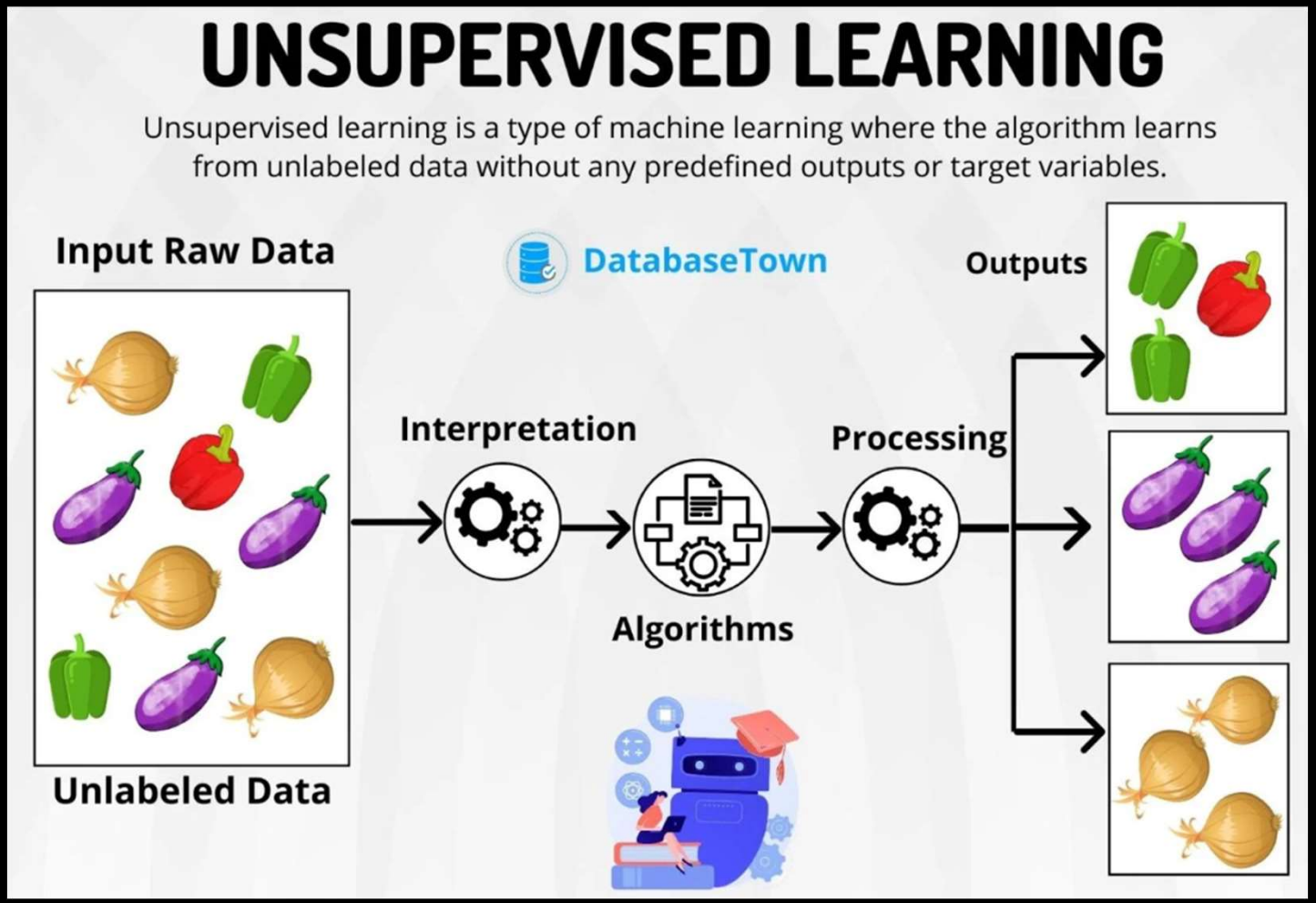
Features /
Independent variables

ML Concept: Features / Independent Variables
Target/Dependent Variable

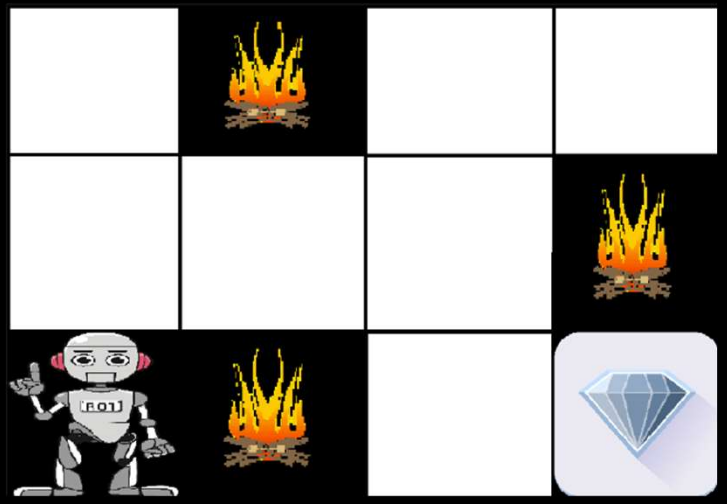
How do we train the model? – supervised learning

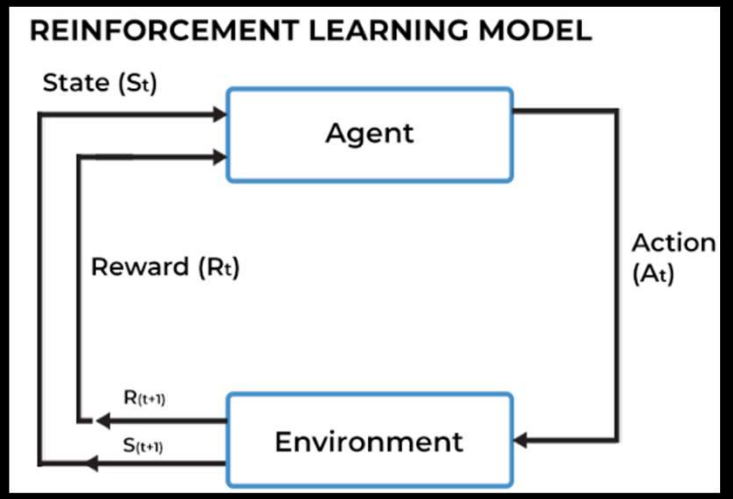


How do we train the model? – unsupervised learning



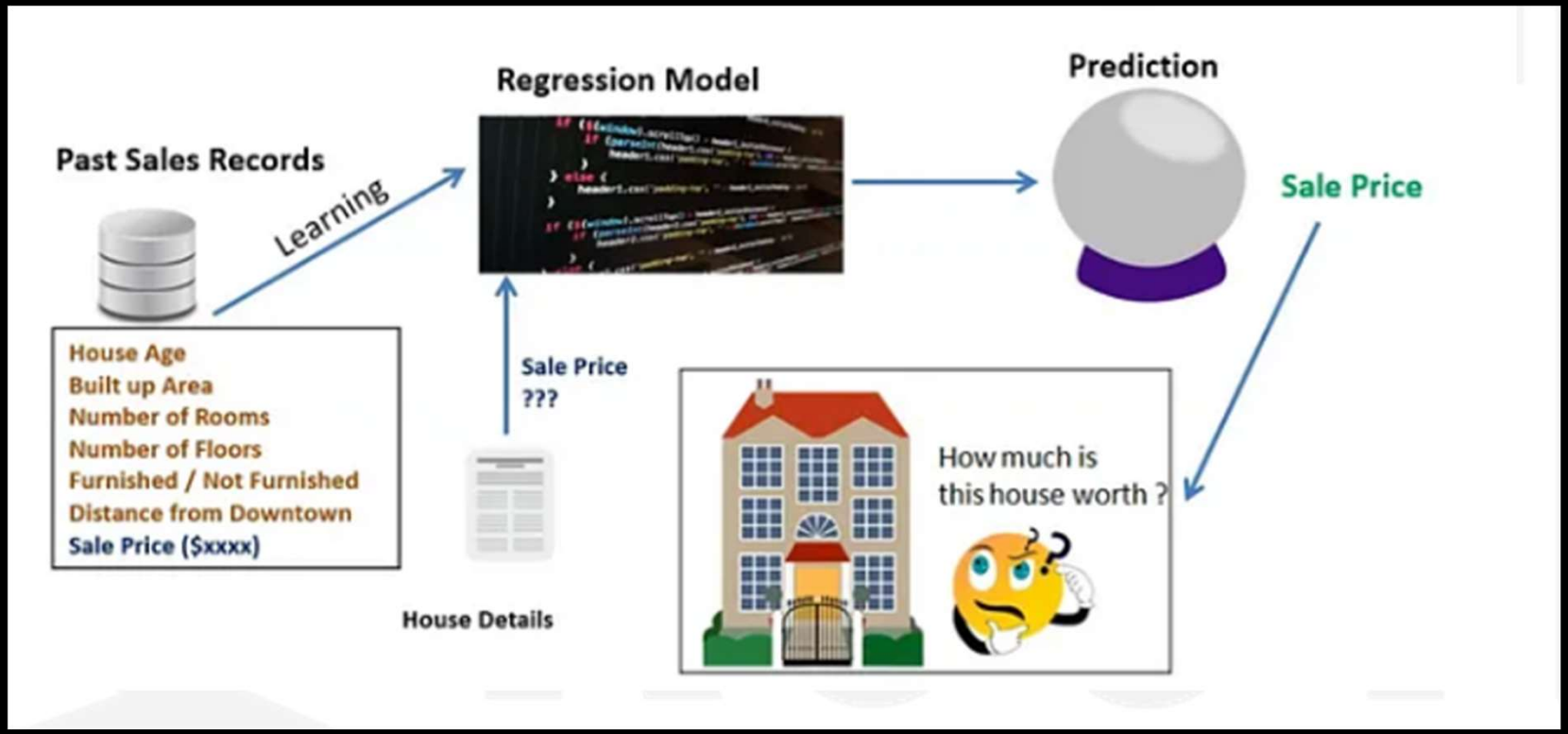
Reinforcement learning



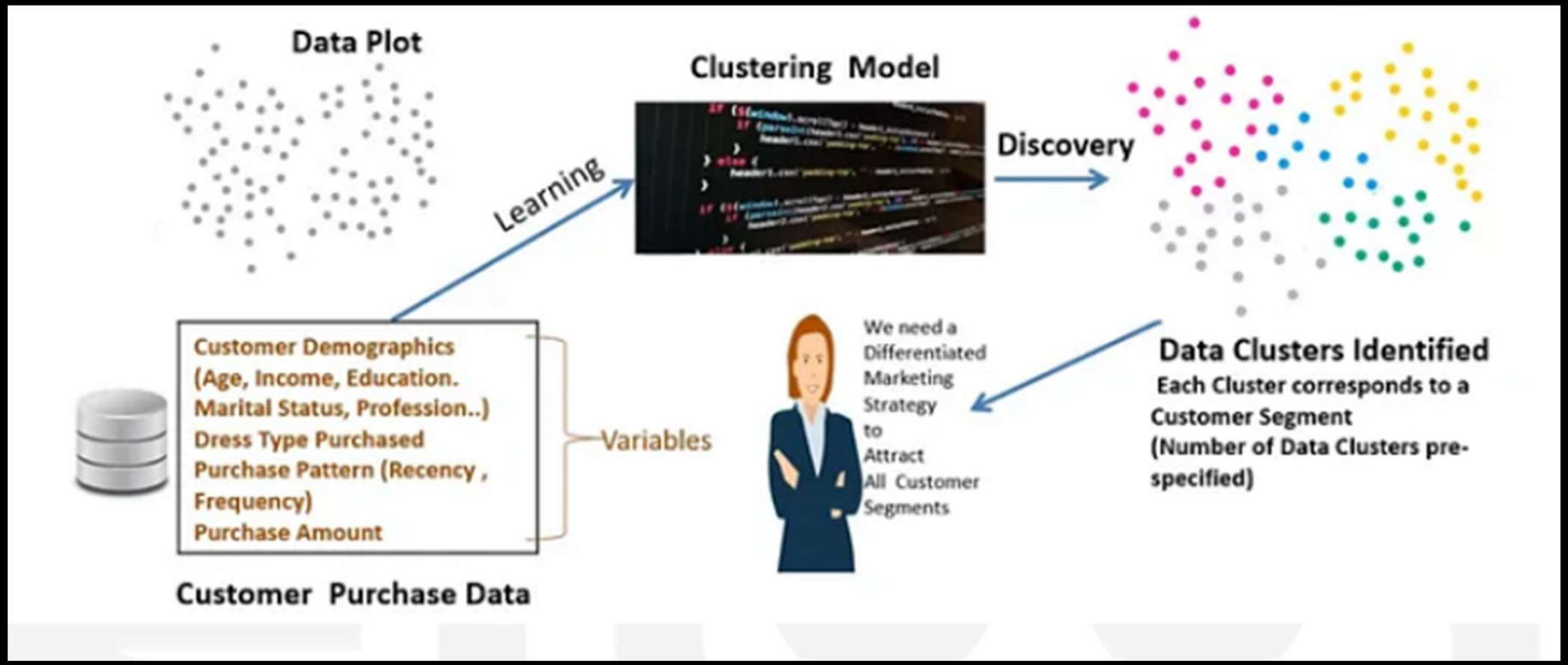


- The goal of the robot is to reach the diamond without hitting the fire squares.
- For each correct move, it is rewarded
- There is no input data
- State, Action, Reward are the key steps
- Robot is the Agent in our example
- Game Board is the Environment

Regression example: property price prediction



Clustering example: customer segmentation



Data science problem types summary

Problem	Description	Examples
Classification	Predicting whether a data point belongs to one of the predefined classes	Customer Churn Prediction, Spam Detection
Regression	Predicting the numerical value of the target variable	Agricultural Yield, Inflation Rate Prediction
Clustering	Identifying natural groups within the dataset based on similar inherent property of the data points	Social Network Analysis, Crime Incidence Analysis, Search Result Grouping
Anomaly Detection	Predicting whether a data point is an outlier in comparison with other points in the data set	Detecting Network Intrusions, Predicting Machine Failures, Detecting Gene Mutations
Association	Discovering rules that govern frequent simultaneous occurrence of certain items or phenomena	Medical Diagnosis, Protein Sequencing, Building Intelligent Transportation Systems
Recommendation	Suggesting items for users based on the past preferences of theirs and similar users	Recommendation of Movies, Books, Restaurants, Holiday Destinations

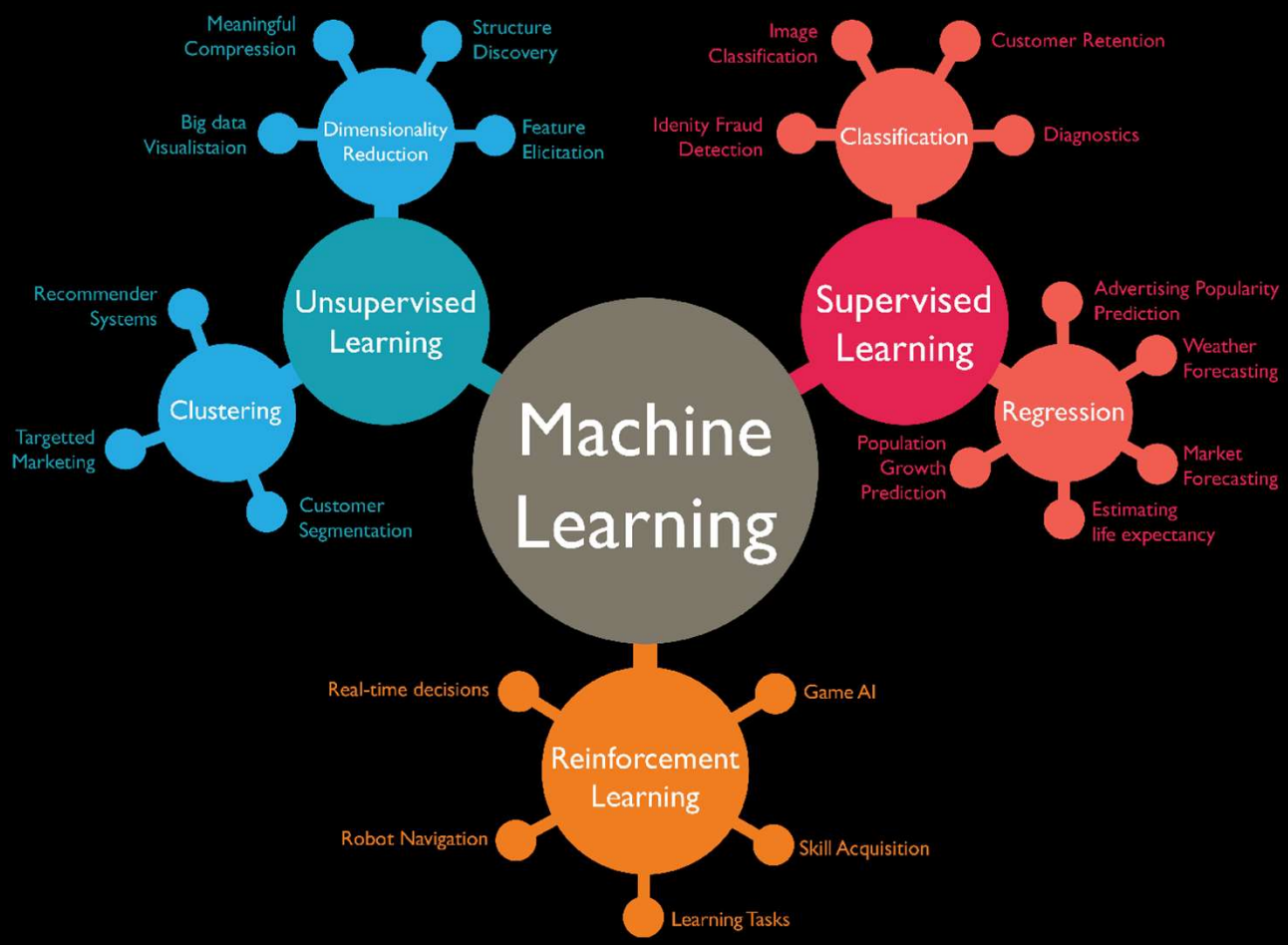
ML Problem types and training methods

Problem Type	Training Method	Common Algorithms
Classification	Supervised	Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Neural Networks
Regression	Supervised	Linear Regression, Ridge/Lasso Regression, Polynomial Regression, Random Forest, Gradient Boosting Machines (GBMs), Neural Networks
Clustering	Unsupervised	K-Means, Hierarchical Clustering, DBSCAN, Gaussian Mixture Models (GMMs), Spectral Clustering
Anomaly Detection	Semi-supervised, Unsupervised, or Supervised	Isolation Forest, One-Class SVM, Autoencoders, DBSCAN, Gaussian Mixture Models, LOF (Local Outlier Factor)
Association	Unsupervised	Apriori, Eclat, FP-Growth (Frequent Pattern Growth)
Recommendation Engine	Supervised, Unsupervised, or Reinforcement Learning	Collaborative Filtering (Matrix Factorization, Singular Value Decomposition), Content-Based Filtering, Neural Collaborative Filtering, Reinforcement Learning (Bandit algorithms)

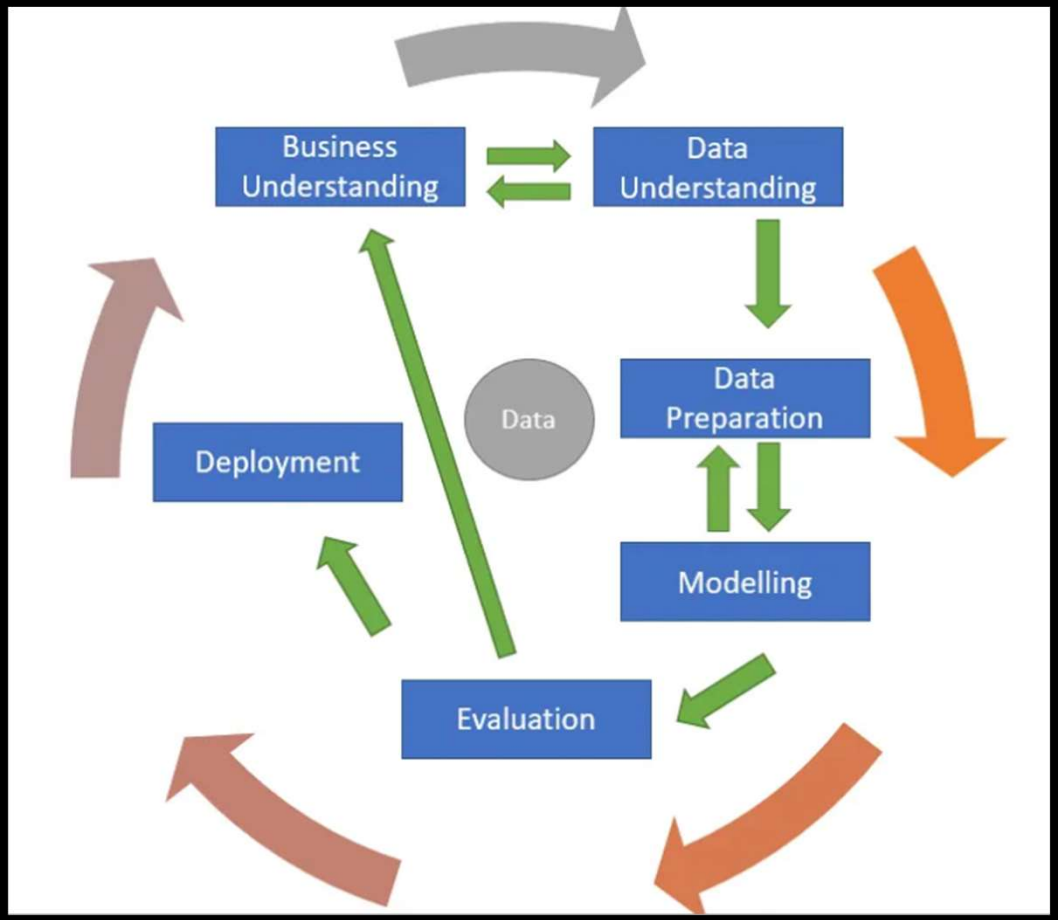
ML Problem types and training methods

Problem Type	Training the model	Model examples
Classification	Supervised	Logistic regression, K-nn, Decision trees
	Unsupervised (Not commonly used)	K-means clustering
Regression	Supervised	Linear regression
	Unsupervised (Not commonly used)	Neural Networks
Clustering	Unsupervised	Neural Networks
	Semi-supervised	Uses a small amount of labeled data to guide the clustering process

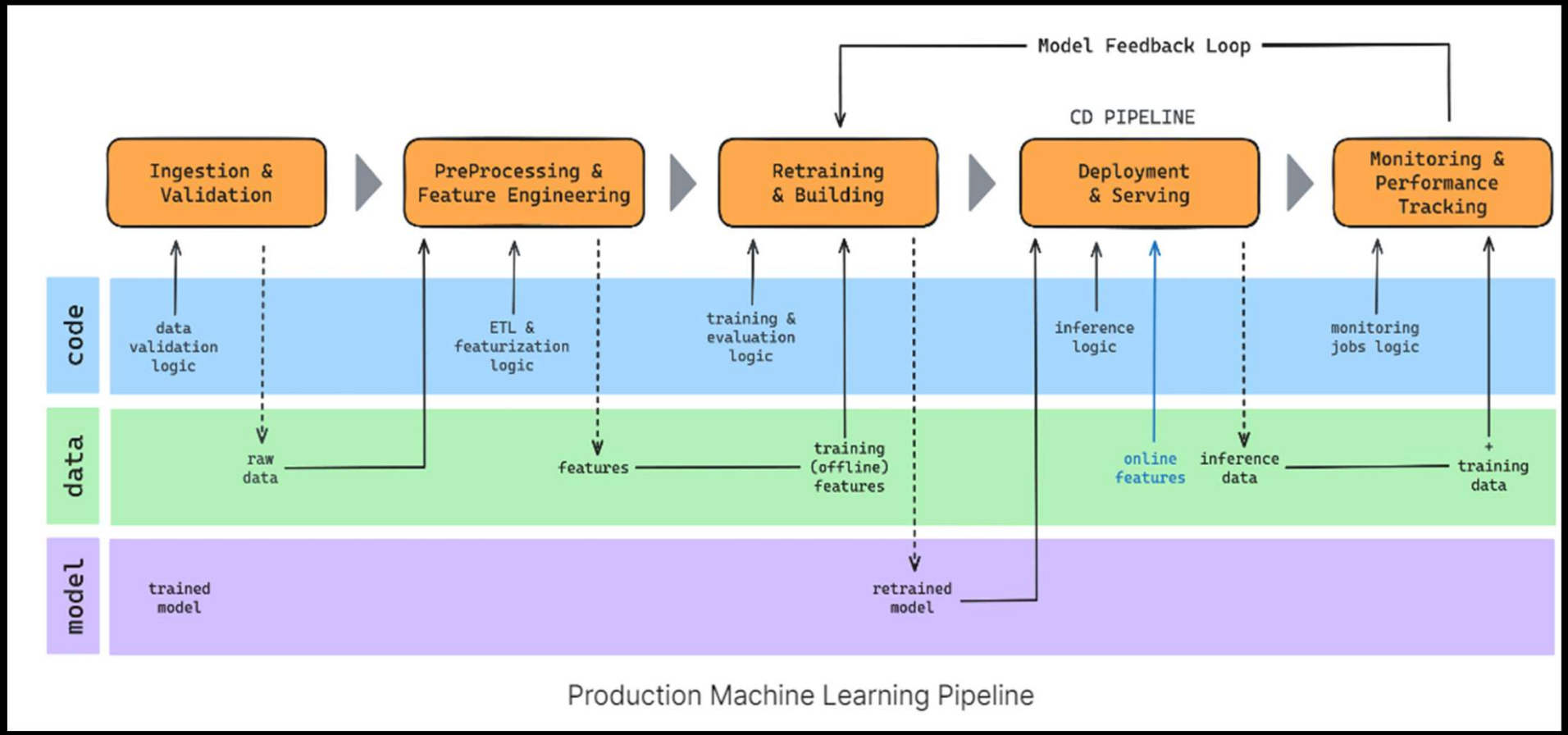
Machine learning map



Machine learning problem solving steps

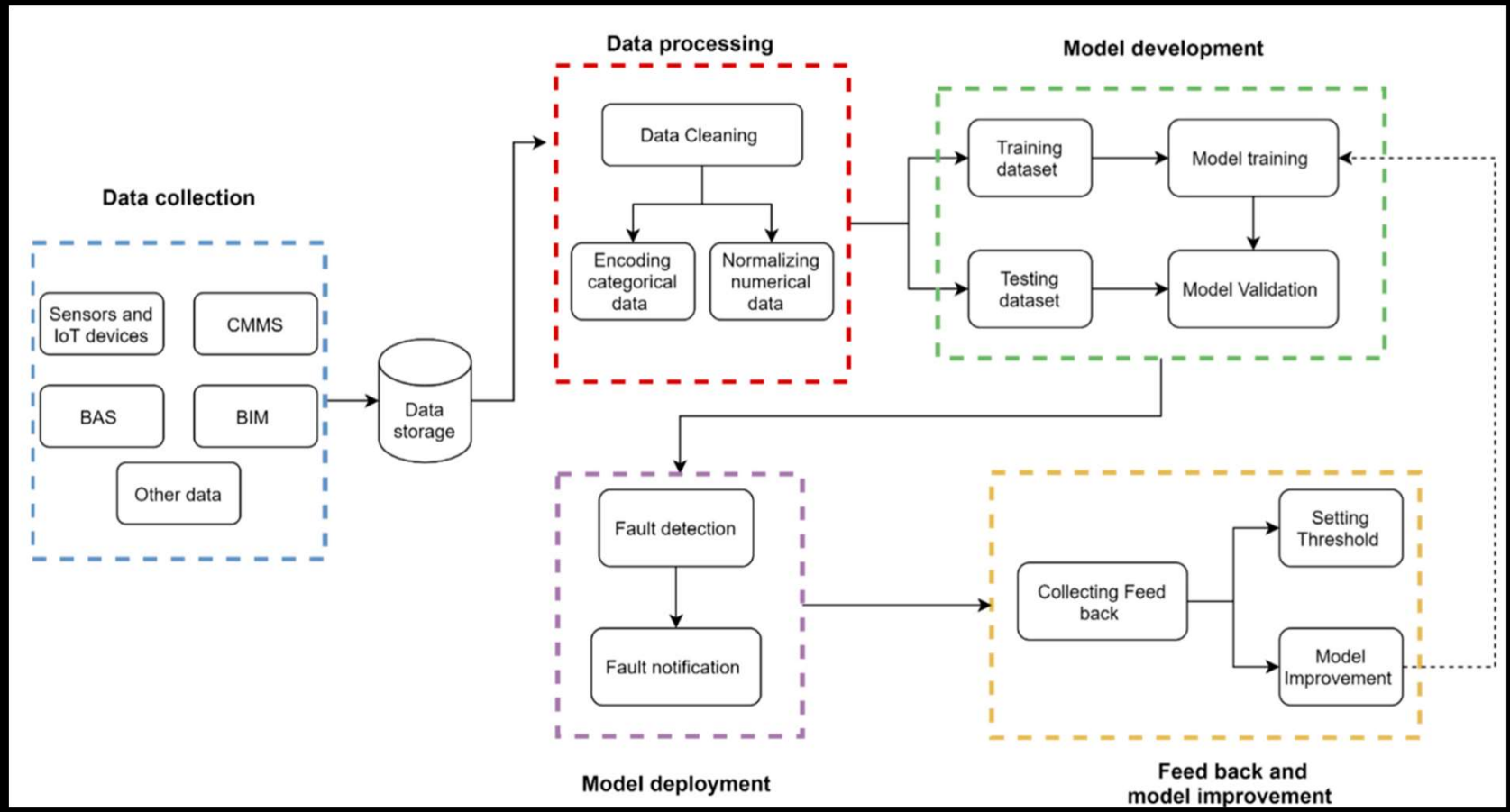


Machine learning pipeline



Production Machine Learning Pipeline

Detailed flow and steps to execute a data science project



Let’s get ready for hands-on

- Python and pip installed on your laptop
- VS Code or similar IDE installed on your laptop
- Logged into colab.research.google.com

Python, pandas refresher

```
ṣʧụđêŋʧṣ□□□□
□□□□ŋắṇê□□□□□Λîkắỳ□□□□ắĝê□□□□_̥□□
□□□□ŋắṇê□□□□□Ṣụŋîʧắ□□□□ắĝê□□□□̦́□□
□□□□ŋắṇê□□□□□N̂êĝḥắ□□□□ắĝê□□□□̦̦□
□
```

```
ṣộsʧêđ□ṣʧụđêŋʧṣ□□lêỳ□ľắṇčđắ□îʧêṇ□□îʧêṇ□□ắĝê□□□□sêŵêsṣê□Ţsụê□
```

```
îṇřộsʧ□řắŋđắṣ□ắṣ□řđ

đğ□□□řđ□Dắʧắ Gsắṇê□ṣʧụđêŋʧṣ□

đğ□ḥêắđ□□
```

numpy refresher

```
# Importing Numpy package
import numpy as np

# Creating a 3-D numpy array using np.array()
org_array = np.array(
        [[23, 46, 85],
         [43, 56, 99],
         [11, 34, 55]])

# Printing the Numpy array
(org_array)

# Printing the Numpy array
np.sort(org_array)
```

Data collection

We will learn following techniques:

- Data collection via API
- Data collection via file upload
- Data collection via web scraping

Loading data from file

```
from google.colab import files
uploaded = files.upload()
```

```
import pandas as pd

# Specify the file path to your CSV file
file_path = 'path_to_your_file.csv'

# Load the data into a pandas DataFrame
df = pd.read_csv(file_path)

# Display the first few rows of the data
print(df.head())
```

Gathering data by web scraping

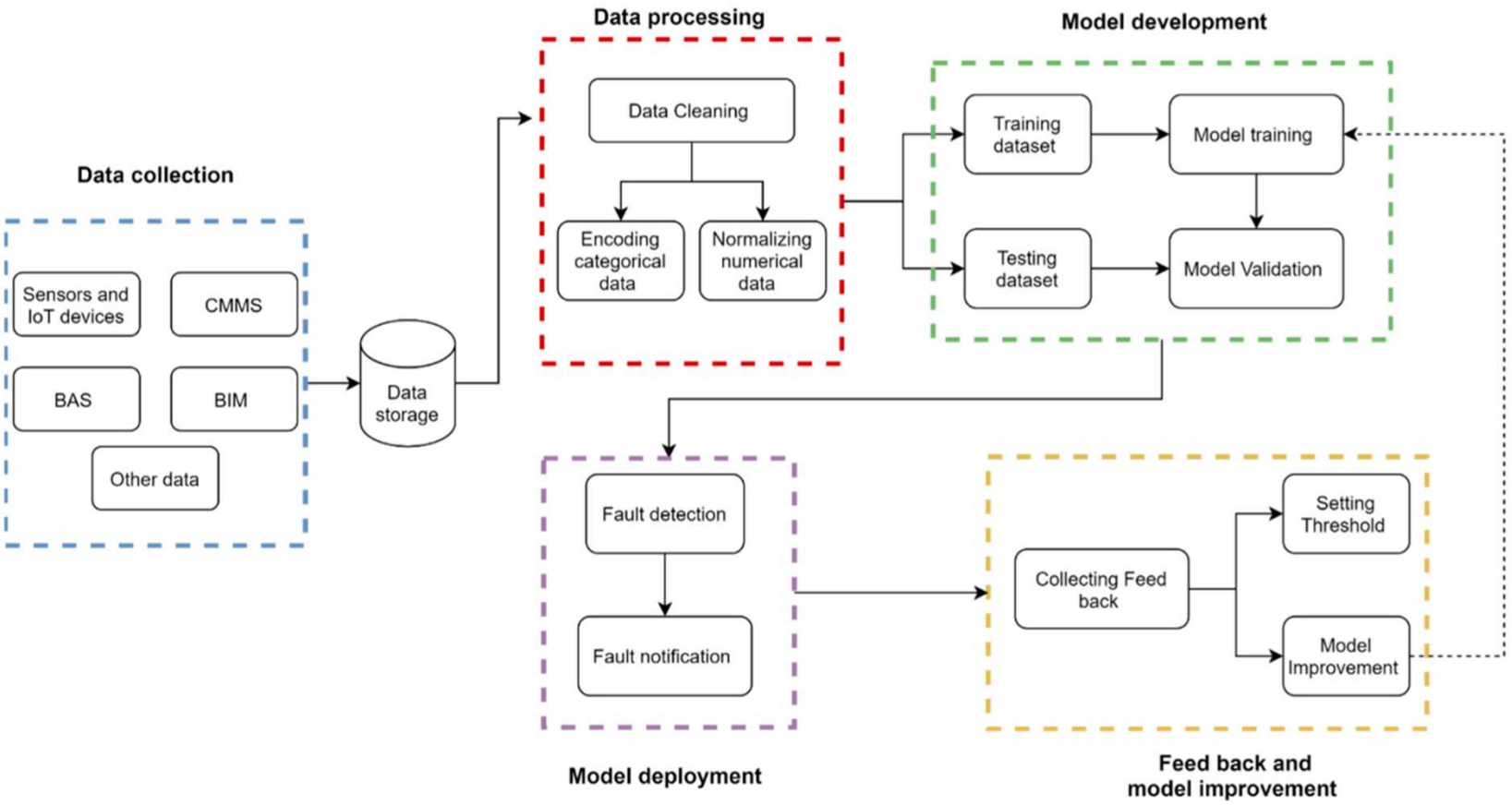
Walkthrough and demo of amazon web scraping

Gathering data by using APIs

Let us fetch stock data by using APIs

https://rapidapi.com/linuz/api/indian-stock-exchange-api2

What are some other types / modes of data sources for data collection you have used / can think of?


Data collection
Sensors and IoT devices
CMMS
BAS
BIM
Other data
Data storage
Data processing
Data Cleaning
Encoding categorical data
Normalizing numerical data
Model development
Training dataset
Model training
Testing dataset
Model Validation
Fault detection
Fault notification
Model deployment
Collecting Feed back
Setting Threshold
Model Improvement
Feed back and model improvement

Exploratory Data Analytics (EDA)

Exploratory Data Analysis (EDA) is the process of **investigating** and **summarizing** a dataset to understand its underlying patterns, relationships, and structures before applying formal modeling or hypothesis testing.

EDA helps in identifying anomalies, testing assumptions, and determining the most important variables.

Why is EDA important?

Informed Decisions

Helps financial institutions make informed decisions about risk, investments, and creditworthiness.

Data Quality

Identifies potential data issues before building models, reducing the risk of faulty predictions.

Hypothesis Testing

Helps generate hypotheses about financial behavior, which can be tested and validated in later stages.

Key steps in EDA

Data Cleaning
Handling missing values, outliers, duplicates, and inconsistencies.

Descriptive Statistics
Summarizing data using mean, median, mode, standard deviation, etc.

Data Visualization:
Using plots like histograms, scatter plots, box plots, and correlation matrices to visualize relationships between variables.

Feature Engineering
Creating new features or transforming existing ones to make data more meaningful.

Key outcomes of EDA

Insights from EDA:

- **Outliers**: You might discover that loans issued to applicants with very low incomes or credit scores have higher default rates.
- **Trends**: You could find a trend where applicants with a higher debt-to-income ratio are more likely to default on their loans.
- **Relationships**: A correlation matrix might reveal that Credit Score is the most important factor in determining loan default.

Cleaning the data

- Removing duplicates
- Remove irrelevant data
- Standardize capitalization
- Convert data type
- Handling outliers
- Fix errors
- Language Translation
- Handle missing values
- Handling null or n/a values

  Imputation: Replacing missing values with a specific value (mean, median, mode, etc.).
  Dropping: Removing rows or columns with null/N/A values.
  Default Values: Assigning a default value to handle missing data.

Cleaning the data

Demo and hands-on

Descriptive Statistics and data visualization

Size of data

Mean and Median of some key features

Calculate the mean and median of applicants' income and credit scores to understand the central tendency.

Distribution of the features and target variable

Loan Status, for example: Check how many loans have defaulted vs. not defaulted, which gives a baseline for understanding risk.

Descriptive Statistics and data visualization

Data Visualization

- **Histogram**: Plot the distribution of credit scores to see if they are normally distributed or skewed.
    - Example: A histogram might reveal that most applicants have a credit score between 600-800, but there are a few with scores below 500, indicating high-risk profiles.
- **Scatter Plot**: Plot Income vs. Loan Amount to see if there is a relationship between income and the loan size applicants seek.
    - Example: A scatter plot could show that higher-income applicants tend to apply for larger loans, but there are exceptions.
- **Correlation Matrix**: Create a correlation heatmap between variables like Credit Score, Income, and Loan Status to identify strong correlations.
    - Example: If you find a strong negative correlation between Credit Score and Loan Default, you can hypothesize that lower credit scores are associated with higher loan defaults.

Descriptive analysis and visualization

Demo and hands-on

Feature Engineering

- Dimensionality Reduction
- One hot encoding (convert categorical features into numerical)
- Normalization (Min-max scaling, Log transformation)
- Binning
- Adding new columns

Feature Engineering

Demo and hands-on

Case study – loan defaulters prediction EDA

https://www.kaggle.com/code/abhishek14398/loan-defaulters-prediction-eda-lending-club-study