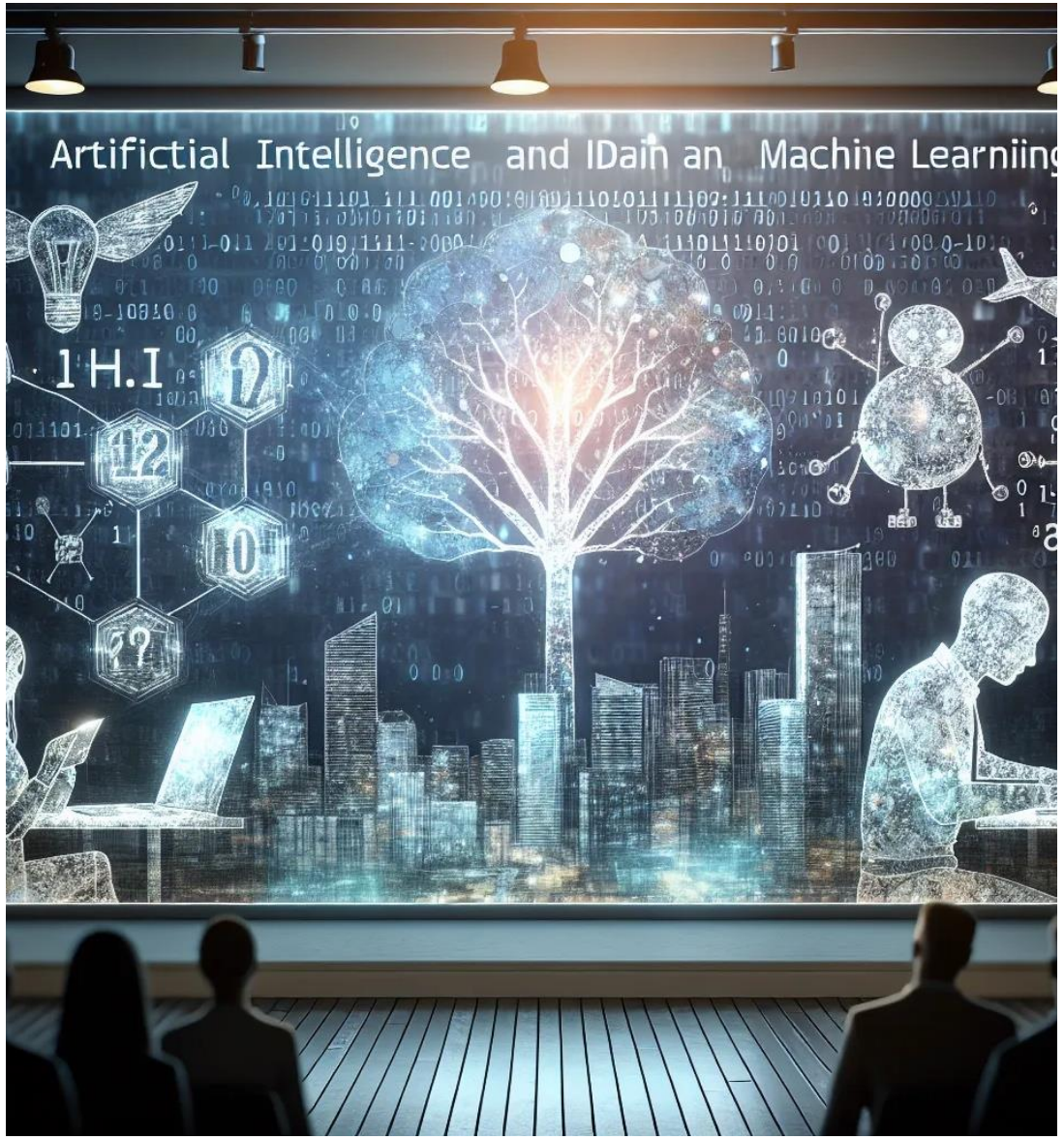
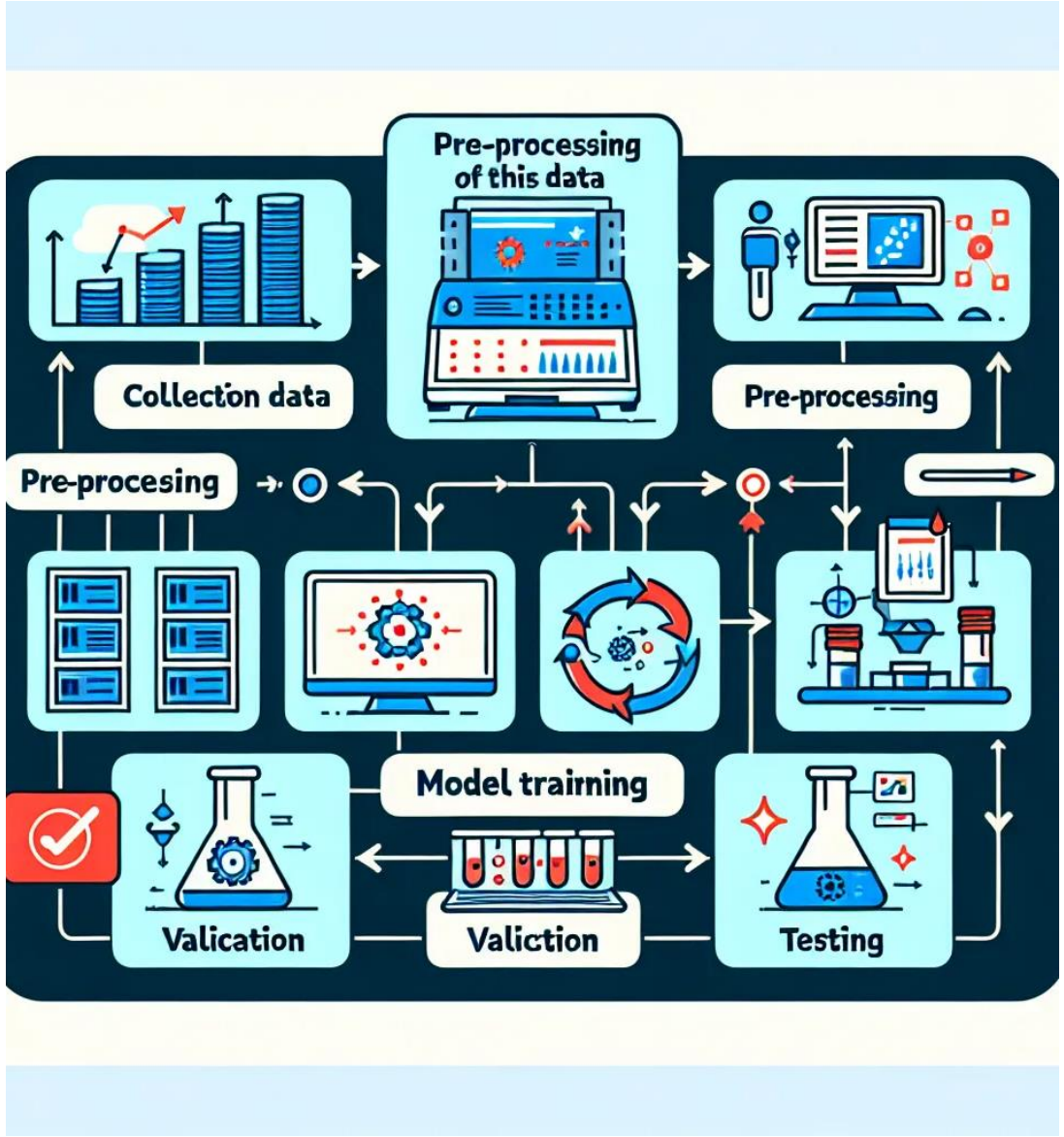
Artificial Intelligence and Machine Learning

- **Hardware**:
- CPU, GPU, Storage, Memory, Cloud Computing, Edge Computing
- **Software**:
- Data Handling & Preprocessing (Pandas, Numpy, Scikit-learn, Tensorflow, Pytorch)
- Machine Learning Algorithms (Scikit-learn, Tensorflow, PyTorch, XGBoost)
- Deep Learning Tools (PyTorch, Tensorflow)
- Visualization: (Matplotlib, Seaborn, Plotly)



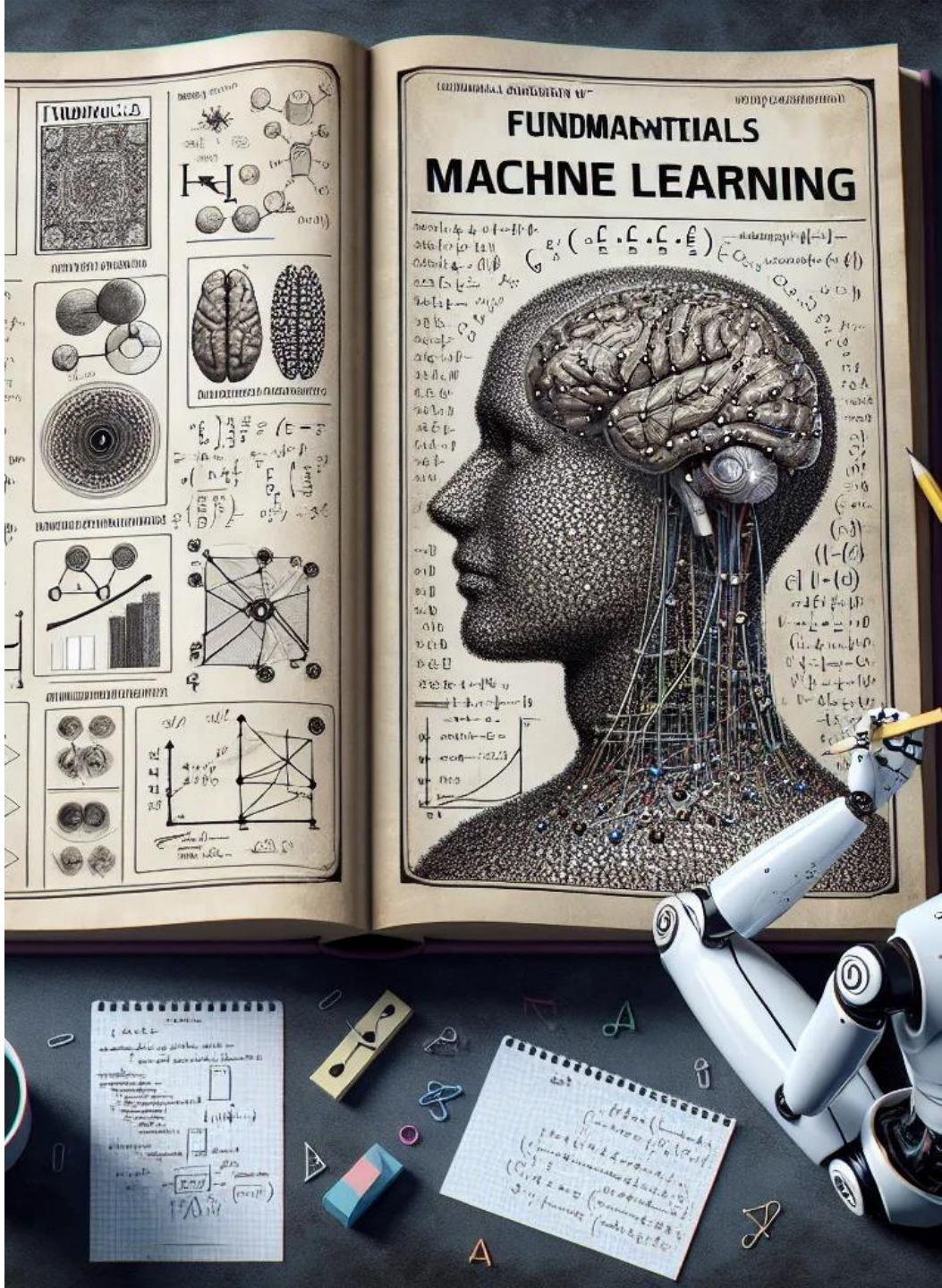
Machine Learning Workflow

1. **Raw Data Collection**: Gather data from various sources.
2. **Data Management**: Store, organize, and secure data.
3. **Data Preprocessing**: Clean, transform, and scale data for modeling.
4. **Feature Selection**: Choose relevant features for model training.
5. **Model Selection**: Choose between supervised, unsupervised, or reinforcement learning.
6. **Model Training**: Train the model using the data.
7. **Model Evaluation**: Evaluate the model using appropriate metrics.
8. **Model Optimization**: Tune hyperparameters and improve the model.
9. **Model Deployment**: Deploy the model into production.
10. **Monitoring & Maintenance**: Continuously monitor and retrain the model as needed.



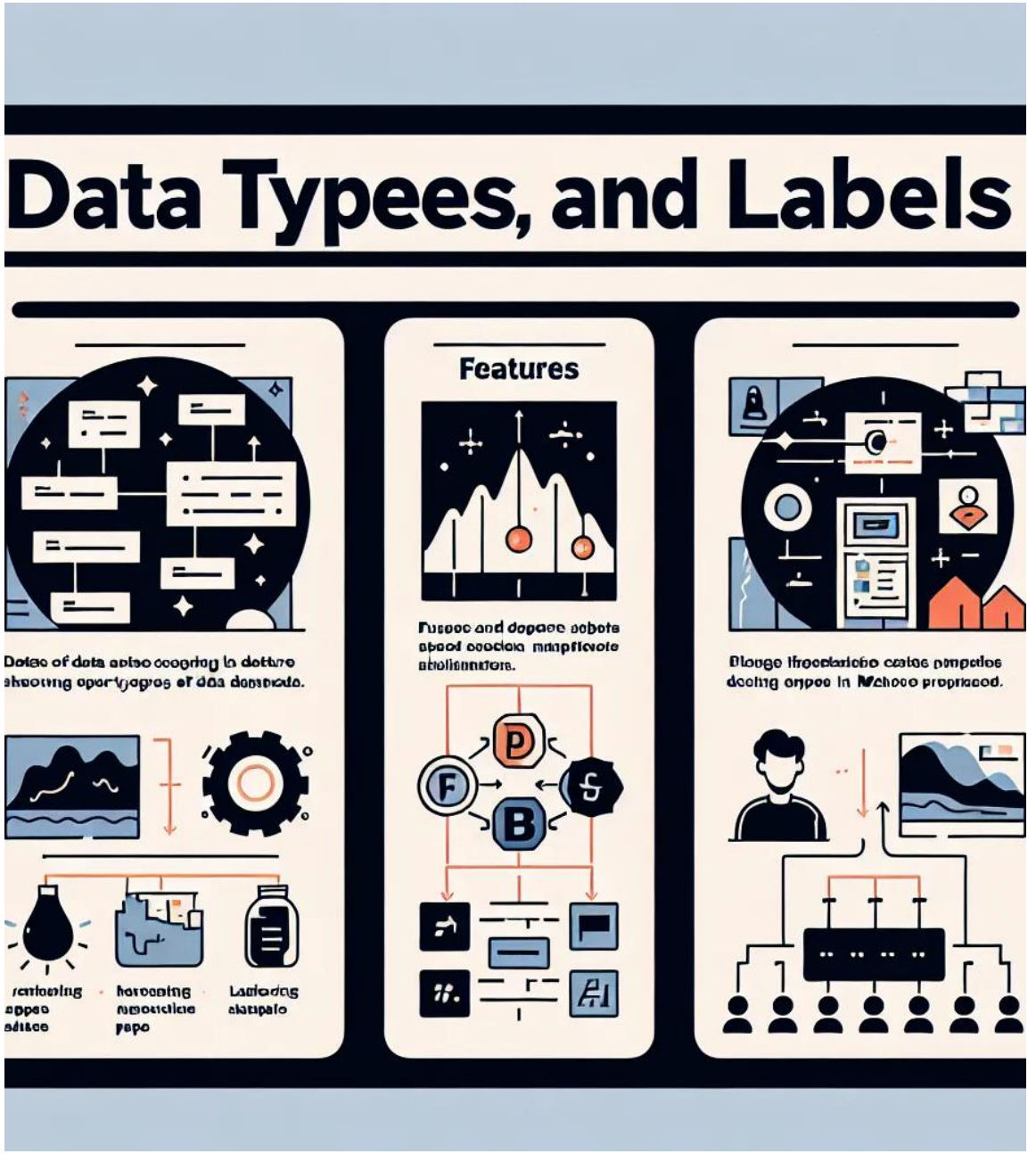
Machine Learning Cheat sheet

- **Label:** Label is the property we are trying to predict. In a linear regression, we use ‘y’ to denote label
- **Features:** Input variables, in linear regression we use ‘x’ to denote features. x_1, x_2, x_3, …. X_n.
- **Regression vs Classification:** Regression model is used to predict continuous values (Eg:- house prices) whereas classification model predicts discrete values (cat or dog). Logistic regression model uses sigmoid function to predict a value between 0 and 1.
- **Training:** Determining the optimum values for the bias and weights to predict the result.
- **Loss:** It is a number indicating how bad the prediction was.
- **Accuracy:** Fraction of predictions that were right in a classification model
- **Clustering**: Grouping of related examples during unsupervised learning (Eg:- K-means)
- **Hyperparameter:** Values that are tweaked during the training of a model (Eg:- Learning rate)
- **Supervised Learning:** Training a model with input data with its corresponding labels
- **Unsupervised Learning:** Training a model to find patterns in data set that are unlabeled.
- **Validation:** A process used in machine learning to evaluate the quality of the trained model.
- **Gradient descent Algorithm**: Calculate the gradient of the loss curve at the starting point. Gradient is the derivative (slope) of the curve. Takes the direction of negative gradient to reduce the loss as quick as possible.
- **Learning rate**: step size in gradient descent to multiply the gradient with a scalar to determine the next point.
- **Training set**: A subset to train the model
- **Test Set**: A subset to test the model.



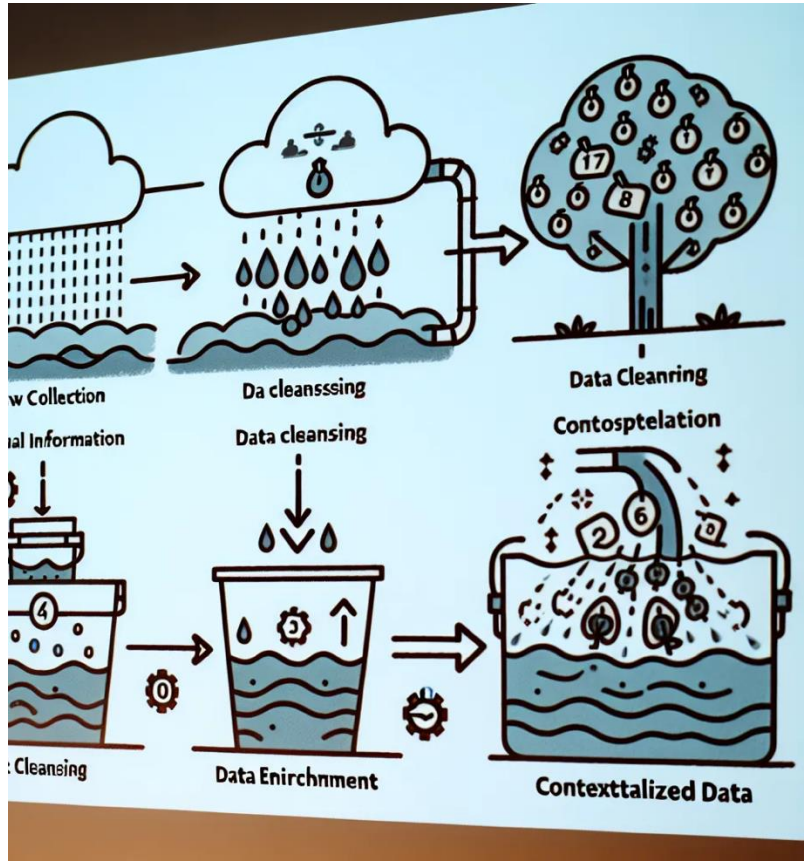
Data Types, Features and Labels

1. **Numeric Data (Quantitative Data, Integers or Floats)**:
    - **Definition**: Data that represents quantities and can be measured and expressed as whole numbers like number of customers or cars or products called **Integers** (50, 10, 5) or numbers with decimals and fractions to express height, weight, temperature (5.5, 120.9, 30.12) called **Floating point** numbers .
2. **Categorical Data (Qualitative Data)**:
    - **Definition**: Data that represents categories or labels. It can take a limited number of distinct values and is typically used for labeling or classifying observations.
    - **Examples**:
        - Gender (e.g., Male, Female).
        - Type of car (e.g., Sedan, SUV, Coupe).
        - Country of residence (e.g., USA, Canada, UK).
    - **Subtypes**:
        - **Nominal**: Categories that don't have a specific order (e.g., colors like red, blue, green; or animal species like dog, cat, rabbit).
        - **Ordinal**: Categories with a defined order or ranking, but without a measurable difference between the ranks (e.g., education level: High School, Bachelor's, Master's, PhD; customer satisfaction: Poor, Fair, Good, Excellent, first, second, third).
3. **Features** are the input variables that are used by the model to make predictions.
4. **Labels** are the output variable that the model is trying to predict based on the features



Data Preprocessing and Contextualization

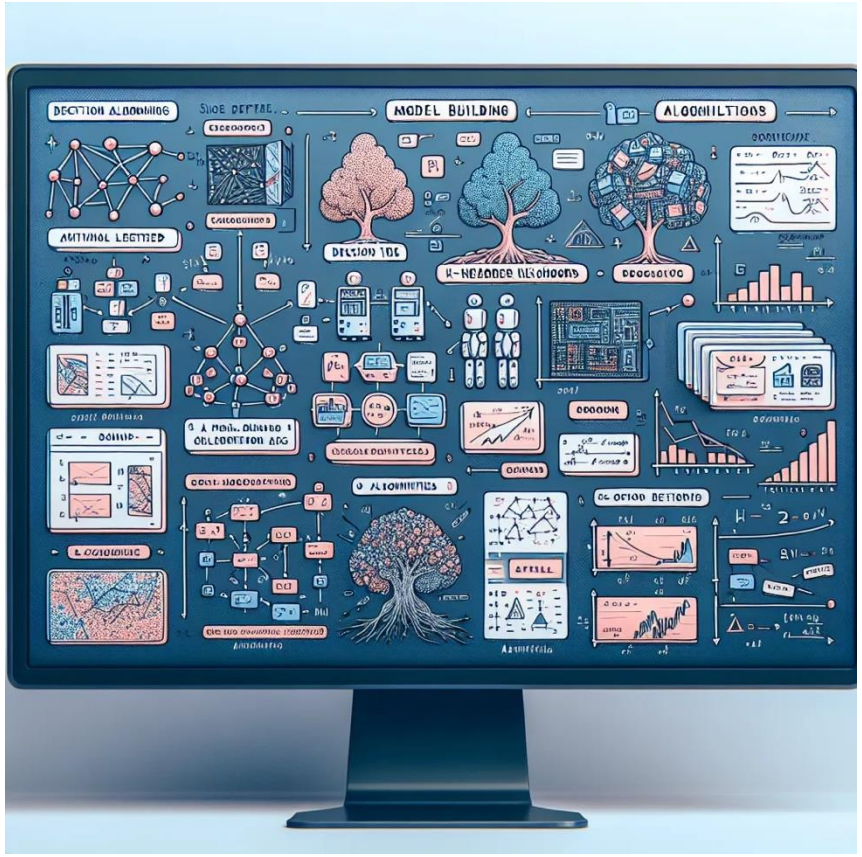
- **Data Cleaning:**
  - **Goal**: To ensure that the data is accurate, consistent, and free from errors. Fix typos, inconsistencies such as ”NY” vs “New York” and remove duplicates. Remove rows or columns with missing values.
- **Normalization / Scaling of Data:**
  - **Goal**: Ensure that numerical features have the same scale, (e.g.: between 0 and 1) so that the machine learning algorithm doesn't bias one feature over another due to differences in magnitude. Having data on the same scale is critical to the performance.
- **Encoding Target Variables**:
  - **Goal**: For classification tasks, ensure that the target (label) variable is encoded correctly to be used in models. **Label Encoding**: For binary or ordinal target variables, map categories to numbers (e.g., "Spam" = 1, "Not Spam" = 0) For a target with three classes, the encoding might be [1, 0, 0], [0, 1, 0], or [0, 0, 1]) or "Red" = 0, "Green" = 1, "Blue" = 2.
- **Handling Imbalanced Data**:
  - **Goal**: Ensure that the dataset is not biased towards the majority class. Over or under sampled.
- **Data Augmentation:**
  - **Goal:** Fill the missing value with mean, median, mode interpolation etc. For images apply transformations like rotation, flipping, cropping, or changing brightness to generate new images
- **Dimensionality Reduction**:
  - **Goal**: Reduce the number of features while retaining as much information as possible. e.g., **Principal Component Analysis (PCA),** A linear method that transforms the original features into a smaller set of orthogonal (uncorrelated) components, which capture the most variance in the data. Reduces computational complexity and helps avoid overfitting.
- **Feature Selection**:
  - **Goal**: Identify and keep only the most important features for model training, removing irrelevant or redundant features. **Correlation-based feature selection**: This involves selecting features that have a strong linear relationship with the target variable, and discarding features that are highly correlated with each other (to avoid redundancy). Reduces overfitting, improves model performance, and speeds up training.



Model Building Algorithms

1. Supervised Learning

- Supervised learning involves training a model on a dataset where the input data (features) and the correct output (label or target) are provided. The goal is for the model to learn the mapping from inputs to outputs so it can make predictions on new, unseen data.
- **Common Algorithms in Supervised Learning:**
  - **Linear Regression**: Used for predicting continuous values (regression tasks). It assumes a linear relationship between the input variables and the output.
  - **Logistic Regression**: Used for binary classification tasks, where the output is a probability score that is then mapped to a class label (0 or 1).
  - **Decision Trees**: A model that splits the data into branches based on feature values, ultimately assigning a predicted class label or continuous value at the leaf nodes.
  - **k-Nearest Neighbors (k-NN)**: A simple algorithm that classifies data points based on the majority class among the k-nearest neighbors.
  - **Neural Networks**: Used for both classification and regression tasks, neural networks consist of interconnected layers of nodes that learn complex patterns in data.



2. Unsupervised Learning

- The model tries to identify patterns, groupings, or structure in the data without being told what the correct output is. The goal is for the model to find hidden patterns or intrinsic structures in the data. In this case, there are no explicit labels or targets, so the model must learn from the data itself.
- Common Algorithms in Unsupervised Learning:
  - **Clustering** is a type of unsupervised learning where the goal is to group data points into clusters or groups based on similarity.
  - **Anomaly Detection**: Identifying rare or abnormal data points in the dataset that do not conform to the expected pattern. Algorithms such as **Isolation Forest** is commonly used for this task.

3. Reinforcement Learning

- The model (agent) takes actions in an environment, receives feedback (in the form of rewards or penalties), and adjusts its strategy to maximize the cumulative reward over time.
- Common Algorithms in Reinforcement Learning:
  - **Q-learning**: A model-free algorithm where the agent learns the value of actions in a given state, updating its knowledge based on rewards received.
  - Application in autonomous vehicles to train self-driving cars to navigate and make decisions based on their environment.

Model Evaluation

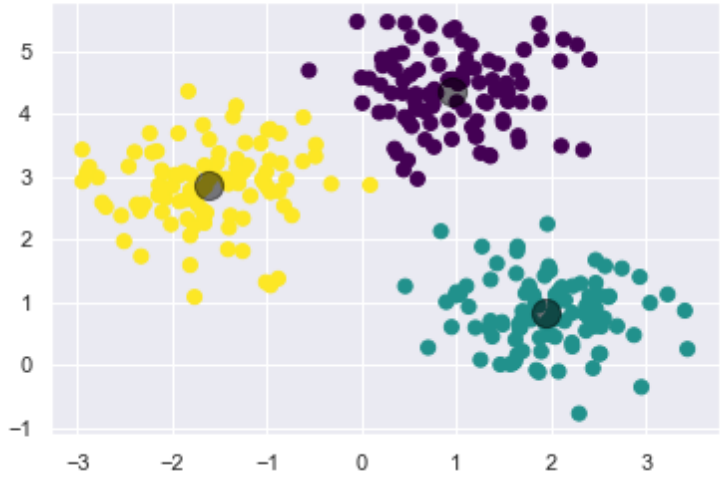
Model evaluation helps assess the performance of a trained model and ensures that it generalizes well to unseen data. During training and testing, several evaluation metrics are used to quantify the model's effectiveness, and these can be related to both the **loss** (or error) during training and **performance metrics** that evaluate the final model.

- Key Concepts in Model Evaluation:
  - **Training loss** (also known as **cost** or **error**) is a measure of how well the model is performing on the training data. It quantifies the difference between the predicted outputs and the true values (targets) for the training dataset.
    - **Mean Squared Error (MSE)**: Often used for regression tasks, calculates the average squared difference between predicted and actual values.
    - **Cross-Entropy Loss**: Often used for classification tasks, measures the difference between the predicted probability distribution and the true class label.
  - **Validation loss** is calculated using a separate dataset, called the **validation set**, which is not used during training. It helps to evaluate how well the model is performing on data it has never seen before. It helps detect **overfitting**. The goal is to minimize both training and validation loss.
  - **Testing Loss**: It is calculated using the test set, which is a dataset that the model has never seen during training or validation. The testing loss gives an indication of how well the model will generalize to unseen real-world data.
  - **Accuracy** is the percentage of correct predictions (both true positives and true negatives) out of all predictions made.
  - **Precision** is the proportion of true positive predictions (correctly predicted positives) out of all positive predictions (both true positives and false positives).
  - **Recall** is the proportion of true positive predictions out of all actual positive cases (true positives and false negatives).
  - **Visualizing Model Evaluation**
    - Learning Curves, Confusion Matrix, Receiver Operating Characteristic - Area Under Curve

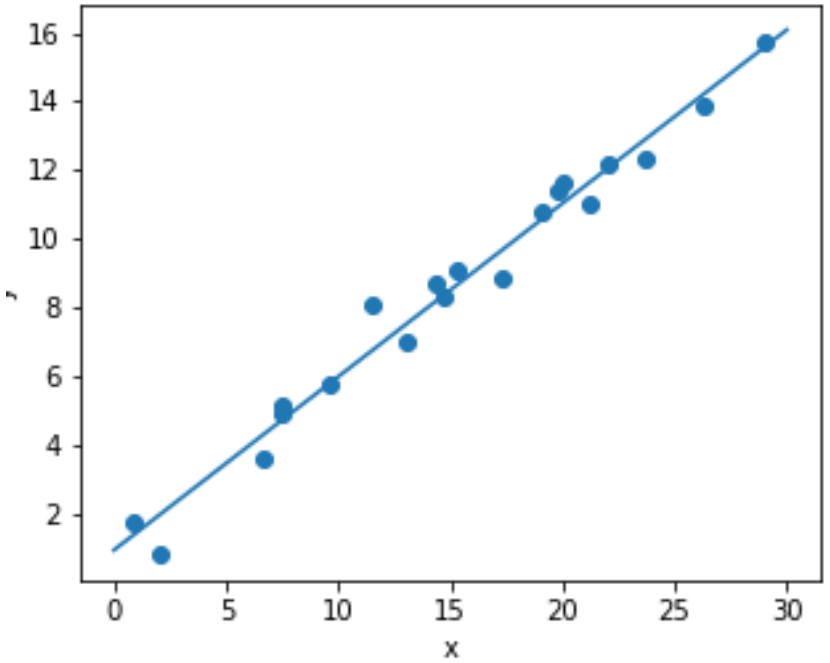
K-means Clustering

- The cluster center is the arithmetic mean of all the points belonging to the cluster.
- Each point is closer to its own cluster center than to other cluster centers.
- Euclidean distance between two points a and b with k dimension is calculated as follows

$$\sqrt{\sum_{j=1}^{k}(a_j - b_j)^2}$$



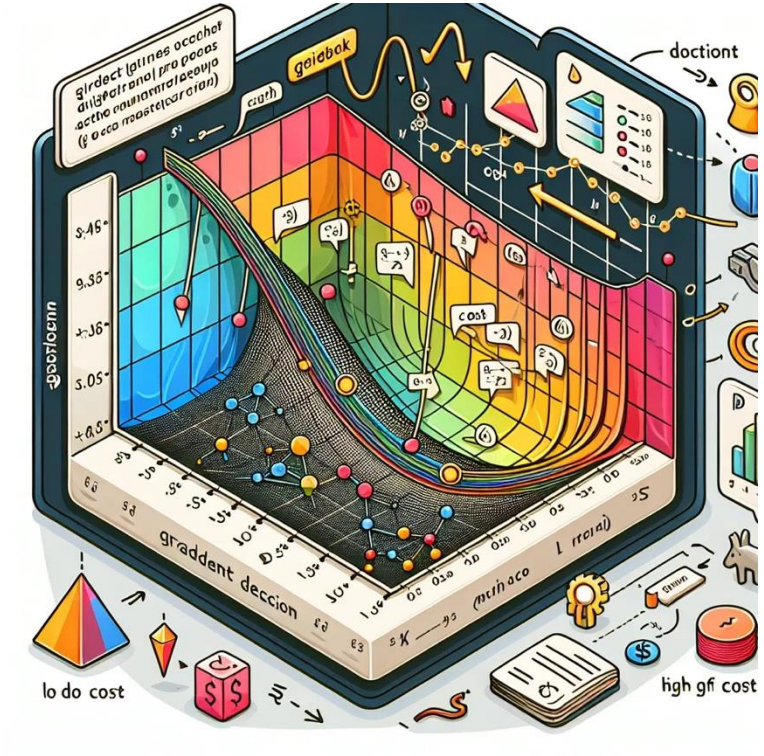
Linear Regression

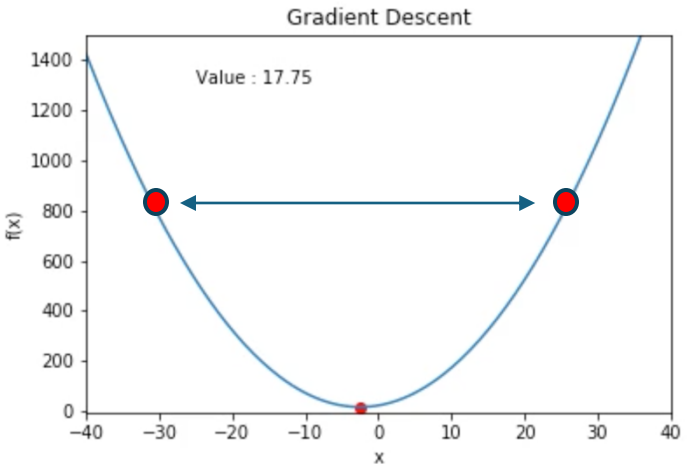


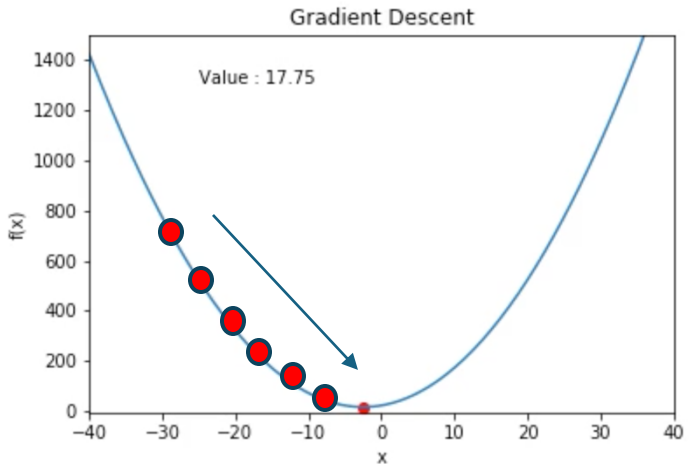
- Mean Square Error
- $MSE = \frac{1}{N}\sum_{(x,y)\in D}(y - y'(x))^2$

- x is set of features
- y is the actual label,
- y'(x) predicted value
- D is the dataset,
- N is the number of examples in the data set

- Linear Regression Model
  - Single feature
- $y' = w_1\, x + b$
  - Multiple features
- $y' = b + w_1x_1 + w_2x_2 + w_3x_3 + \cdots$







Gradient Descent (Hyerparameters – Learning rate, Epochs, Batch size)

- Gradient descent is a mathematical technique that iteratively finds the weights and bias that produce the model with the lowest loss
- Calculate the loss with the current weight and bias.
- Determine the direction to move the weights and bias that reduce loss.
- Alter the weight and bias values a small amount in the direction that reduces loss.
- Return to step one and repeat the process until the model can't reduce the loss any further.

Single Instruction Multiple Threads (SIMT) Algorithm, GPUs and Deep Learning Data Parallelism

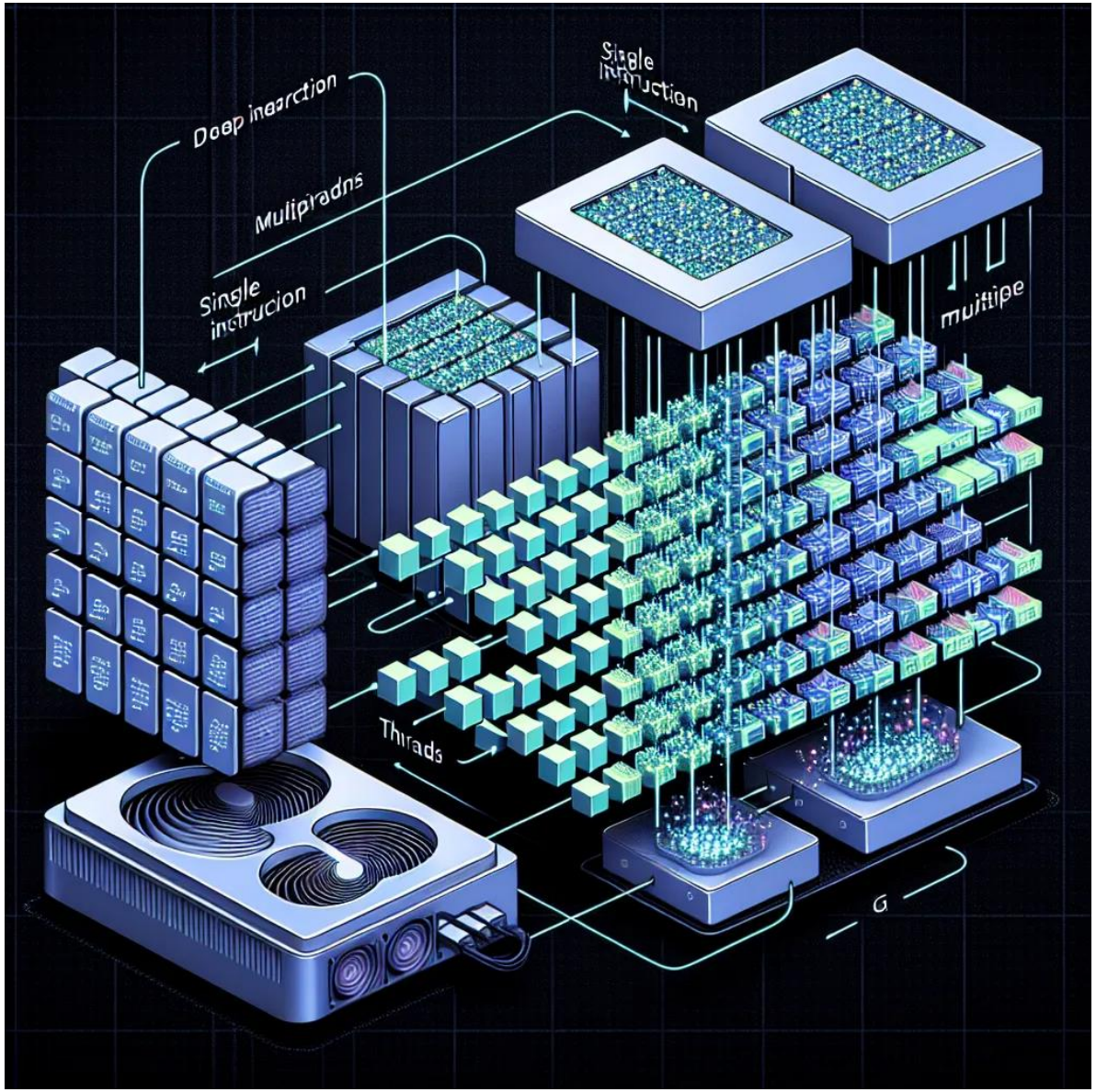
Matrix Vector Multiplications (multiply and add operation)

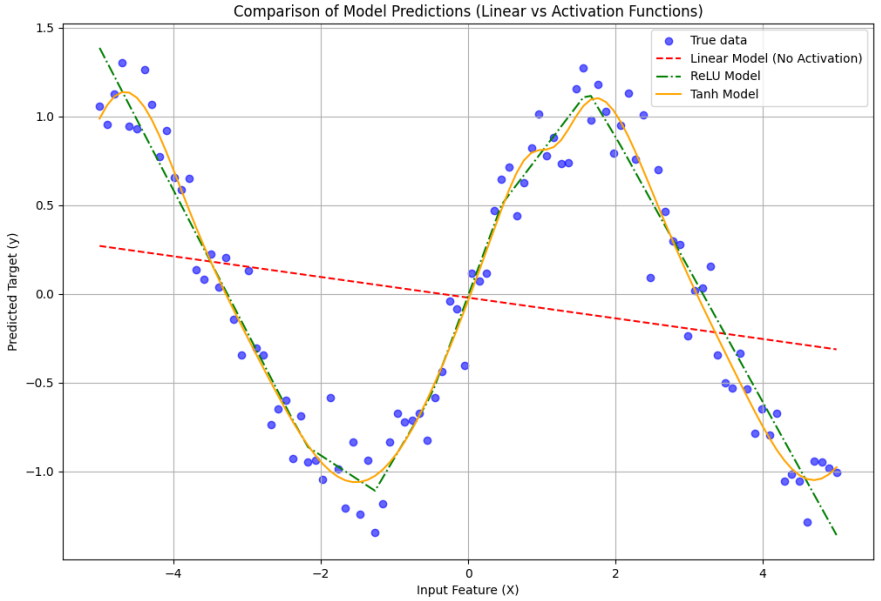
$$\begin{bmatrix} w_{11}, w_{12}, w_{13} \\ w_{21}, w_{22}, w_{23} \\ w_{31}, w_{32}, w_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{11} * x_1 + w_{12} * x_2 + w_{13} * x_3 \\ w_{21} * x_1 + w_{22} * x_2 + w_{23} * x_3 \\ w_{31} * x_1 + w_{32} * x_2 + w_{33} * x_3 \end{bmatrix}$$

Weights Inputs

GPUs Vs CPUs

Tens of thousands of concurrent threads vs Complex Operation





Linearity vs Nonlinearity

Linear Function:

- Imagine a bunch of uncooked spaghetti.
- The strands are like a straight line on graph, like in a linear function.

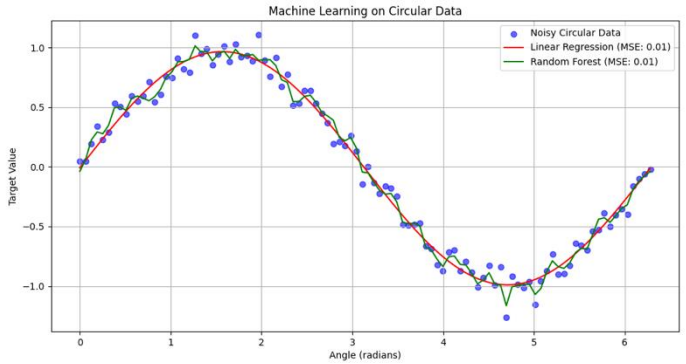
Non-linear Function:

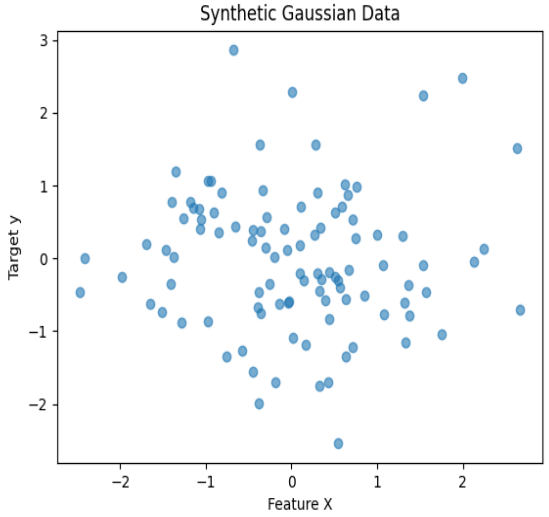
- Now, picture cooked spaghetti.
- The strands are curved, bent, and intertwined, it no longer follows a straight line and represents a non-linear function.

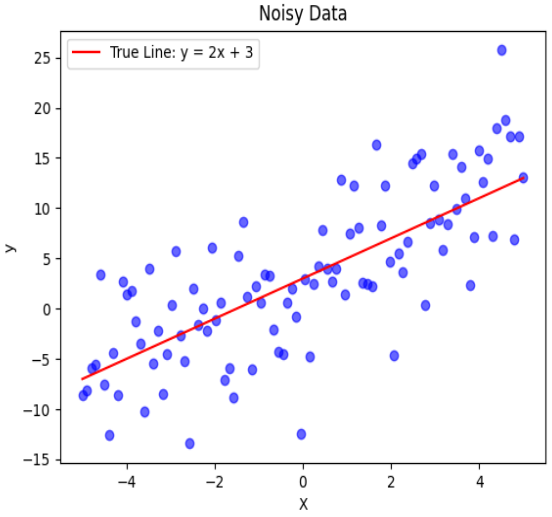
Activation function:

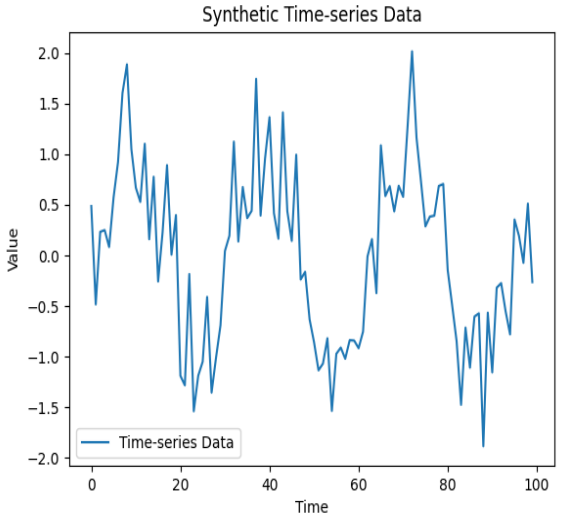
Cooking process is like applying an activation function in a neural network, transforming the "uncooked" (linear) input into a "cooked" (non-linear) output. Sigmoid and ReLU $f(x) = \max(0, x)$ functions are two examples for **activation function**

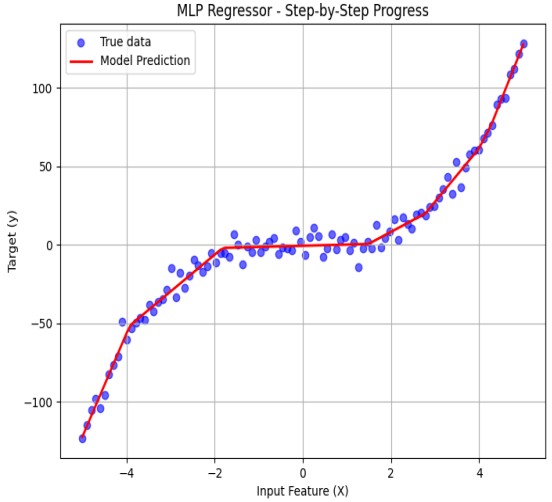
Data Distribution

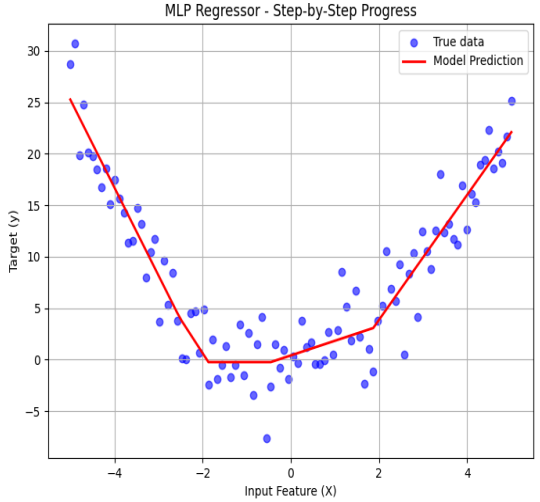




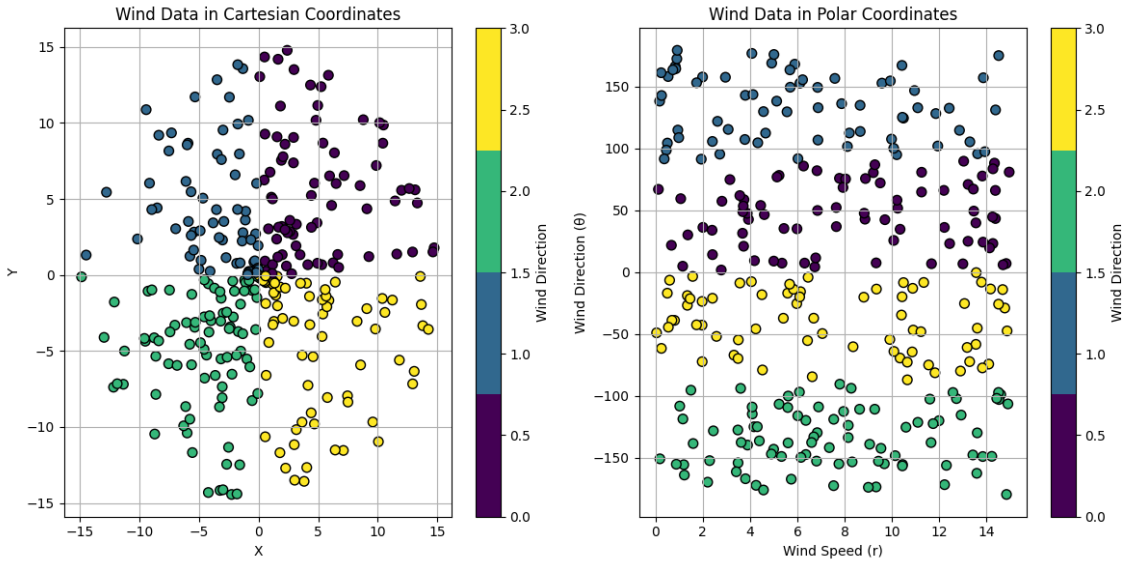


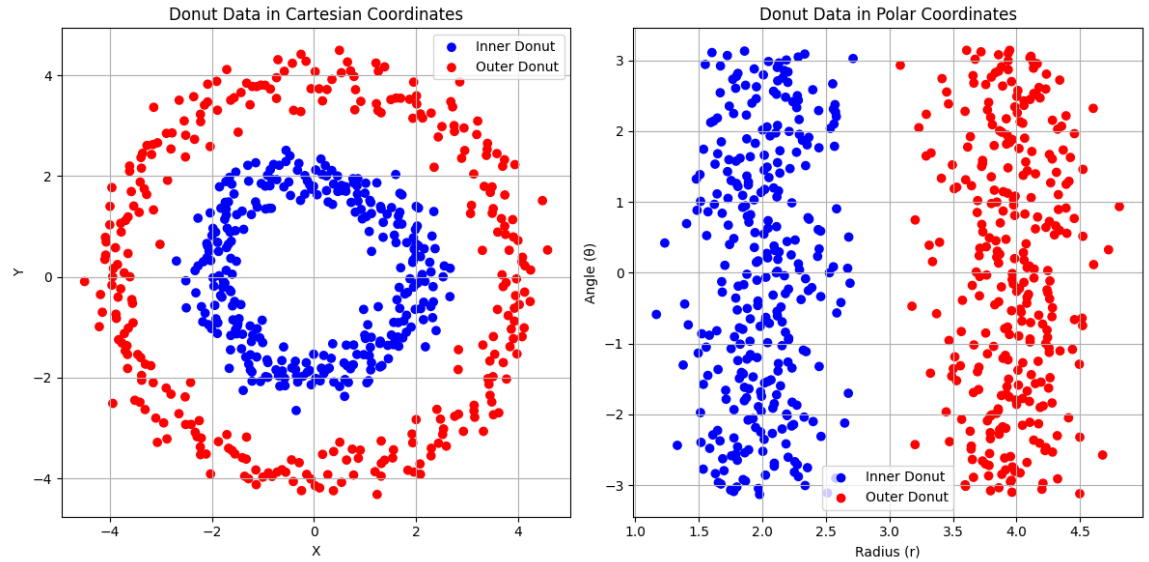


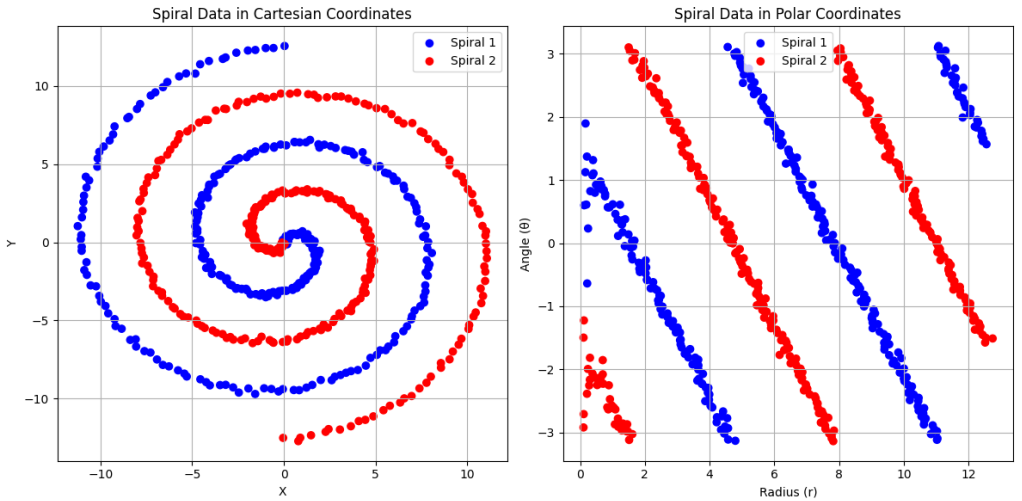


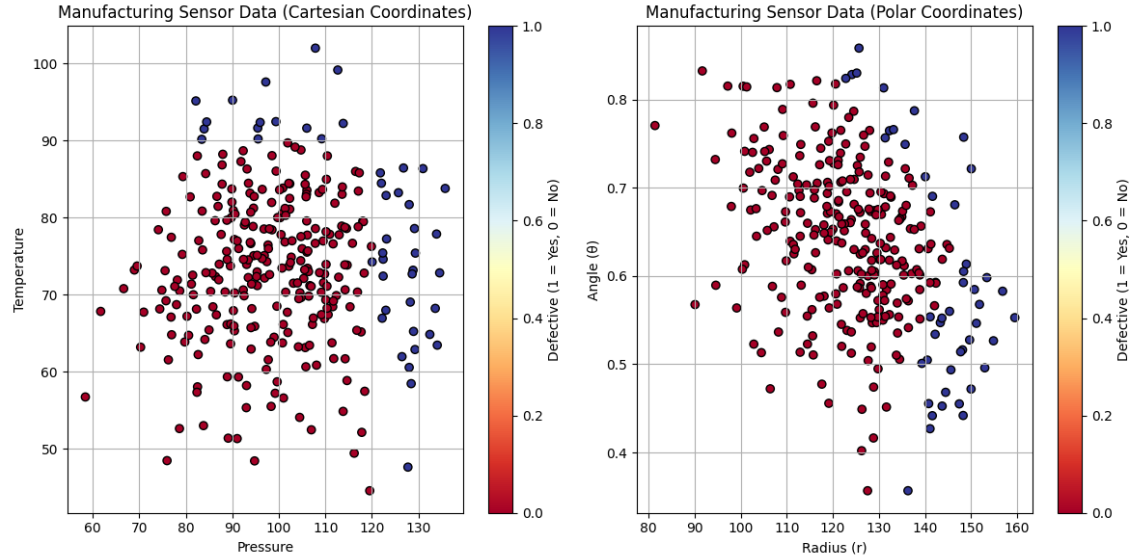


Coordinate Transformation









Dimensionality Reduction and Principal Component Analysis (PCA)