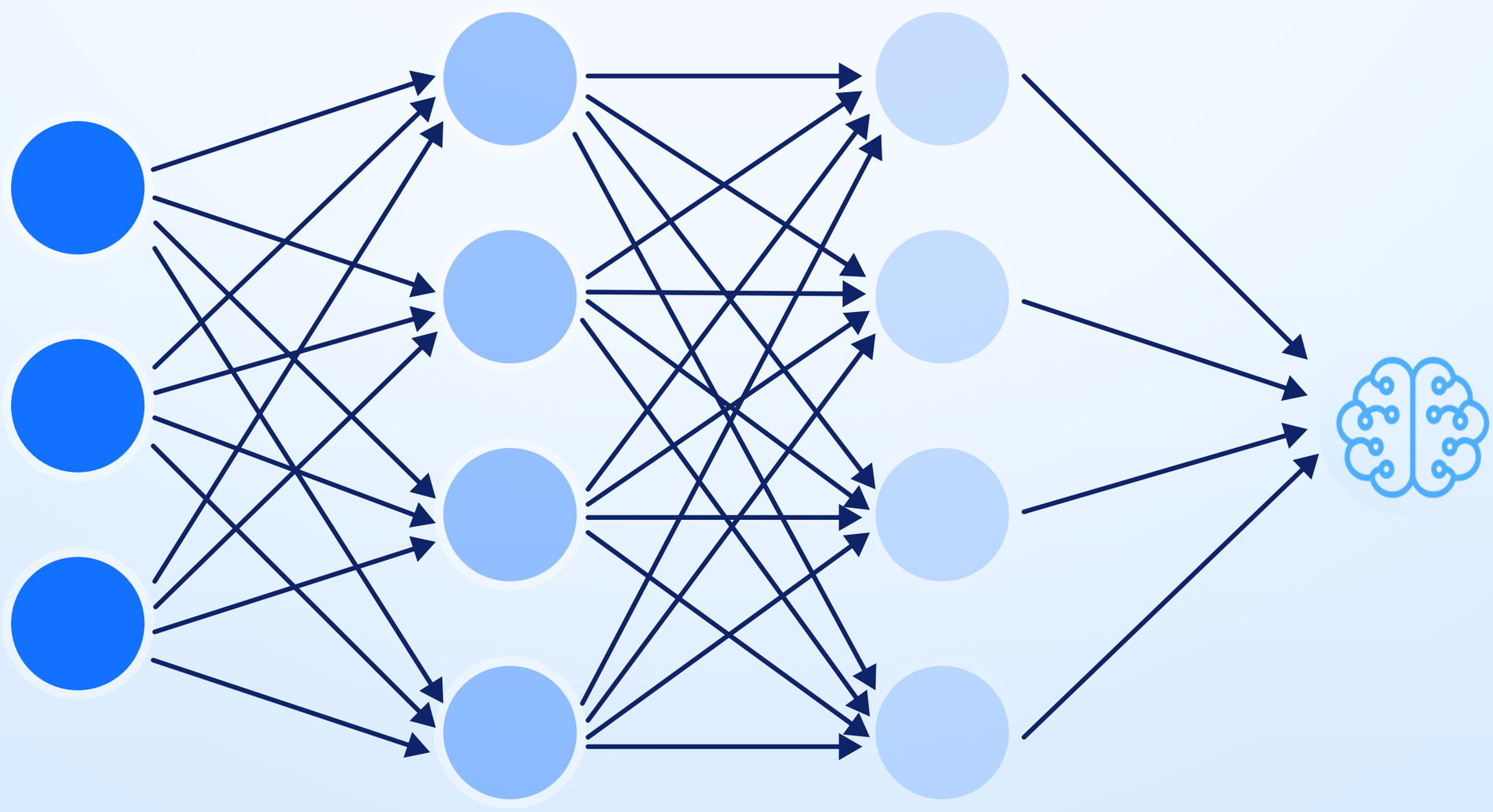
BOSSCODER ACADEMY

# MASTER MACHINE LEARNING



## In Just 30 Days

# *Disclaimer*

Everyone learns uniquely.

What matters is developing the problem solving ability to solve new problems.

**This Doc will help you with the same.**

DAY 1

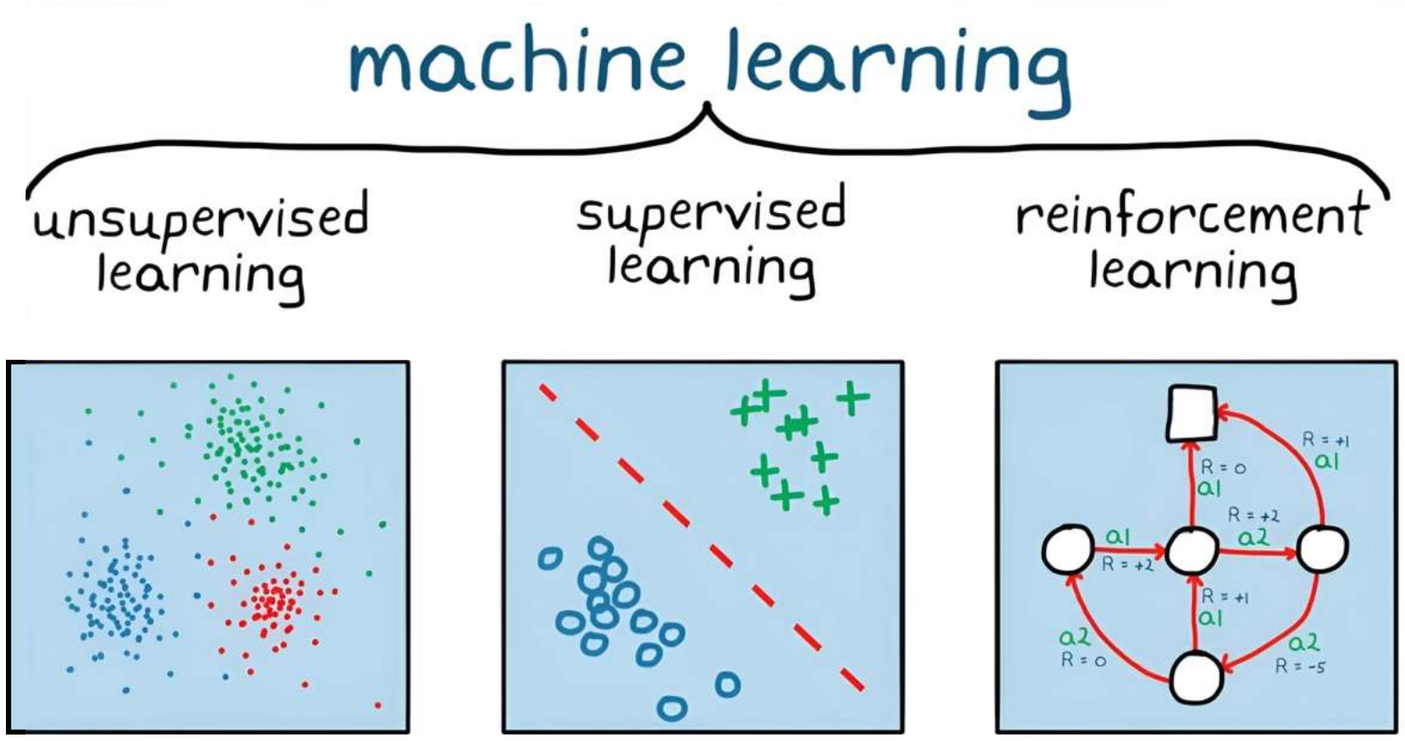
# Basics of Machine Learning

## Concepts:

1. Introduction to Machine Learning
2. Types of problems (classification, regression, etc.)
3. Supervised vs. Unsupervised learning vs Reinforcement learning
4. Basic concepts like features, labels, models, etc.

## Questions:

1. What are the real-world applications of Machine Learning?
2. Can you differentiate between supervised and unsupervised learning?



DAY 2

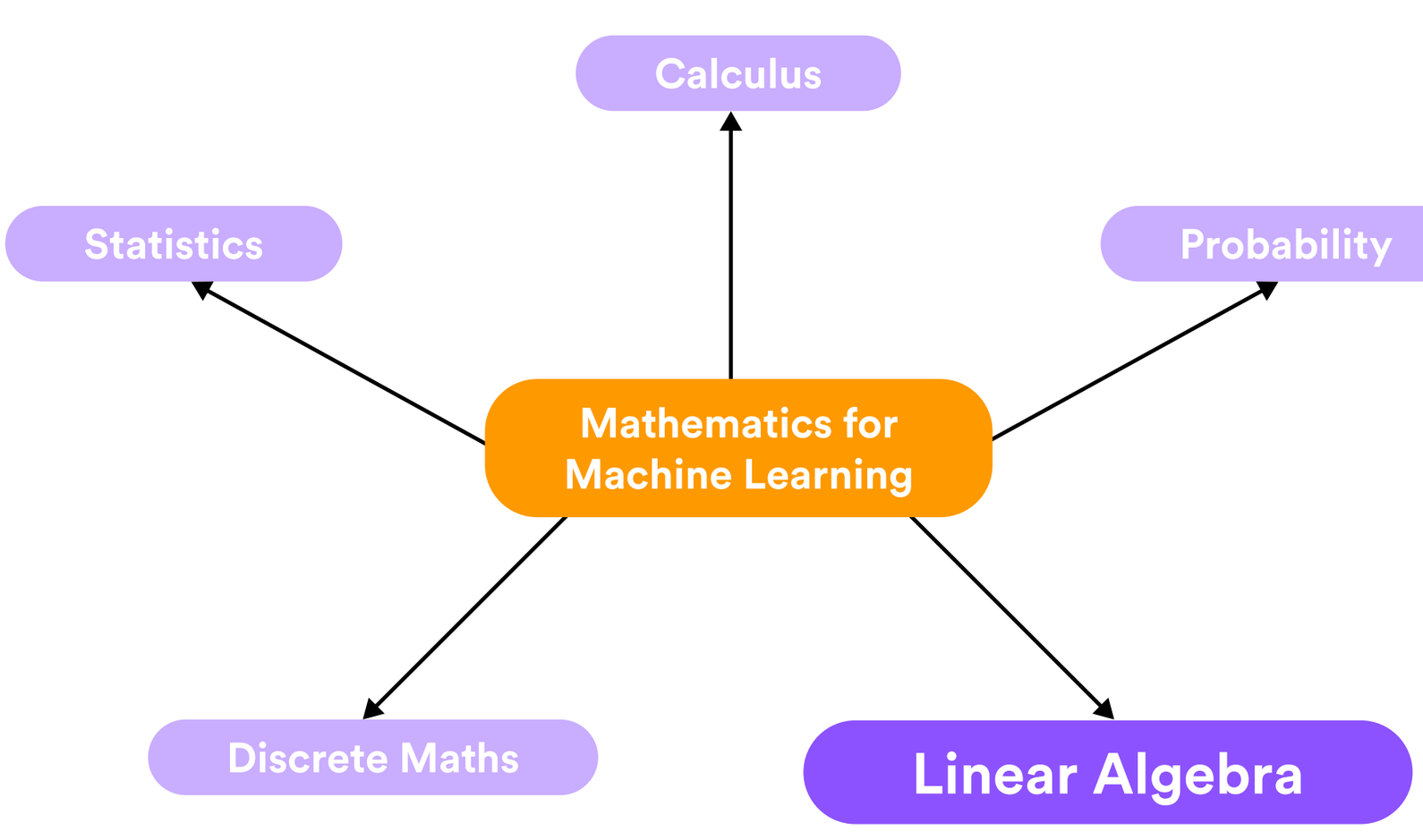
# Linear Algebra Fundamentals

## Concepts:

1. Basic linear algebra concepts (vectors, matrices, operations)
2. Matrix multiplication, transpose, and inverse
3. Vector operations such as dot product and cross product

## Questions:

1. Implement matrix operations using Python and NumPy.
2. Explore how vectors are represented in NumPy and implement their operations



DAY 3-4

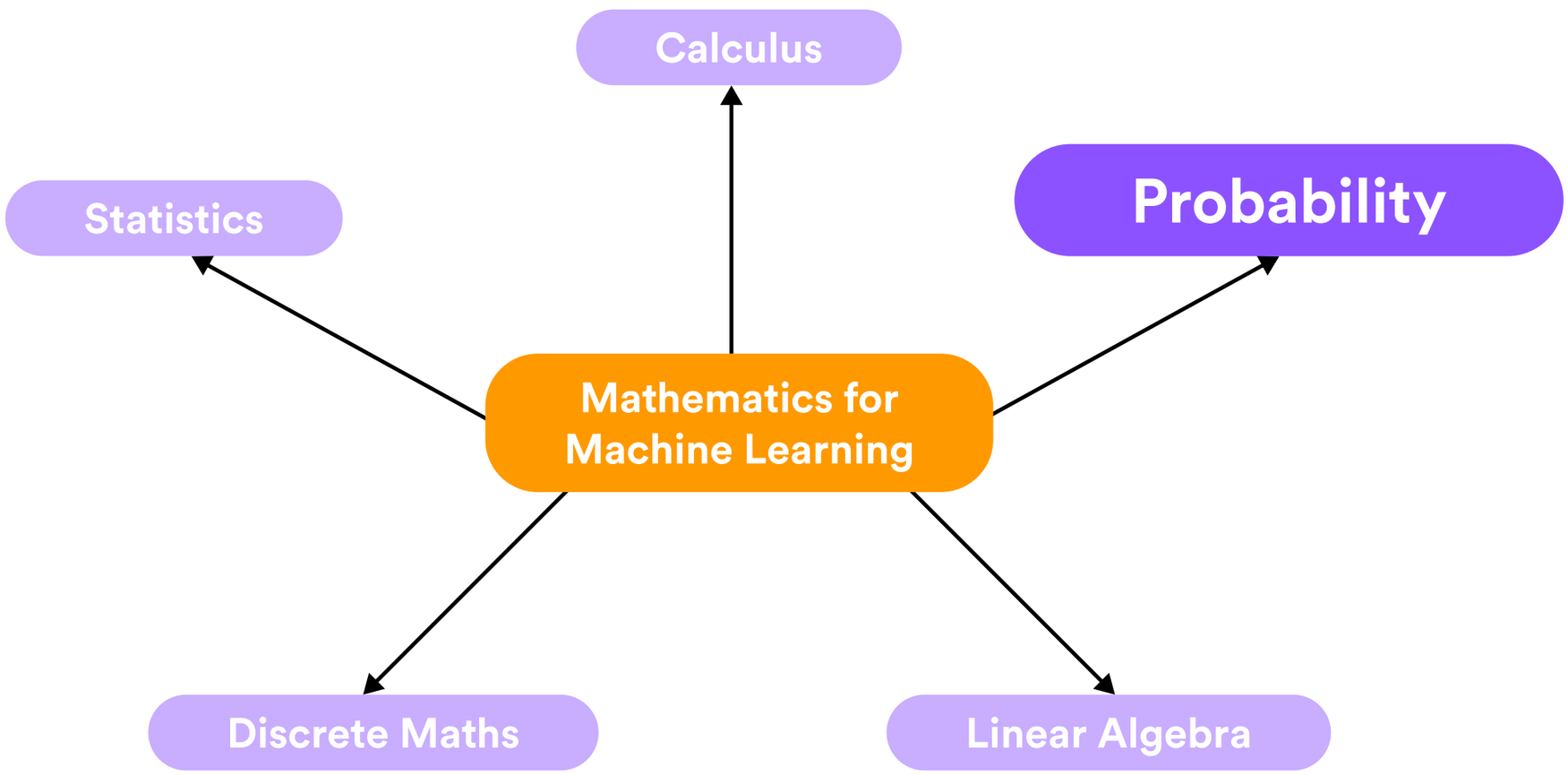
# Probability Fundamentals

## Concepts:

1. Fundamental probability concepts (random variables, distributions, Bayes' theorem).
2. Basic statistics (mean, median, mode, variance, standard deviation).
3. Probability distributions (Gaussian, Bernoulli, Binomial).

## Questions:

1. Calculate probabilities and perform statistical analysis on sample datasets.



DAY 5

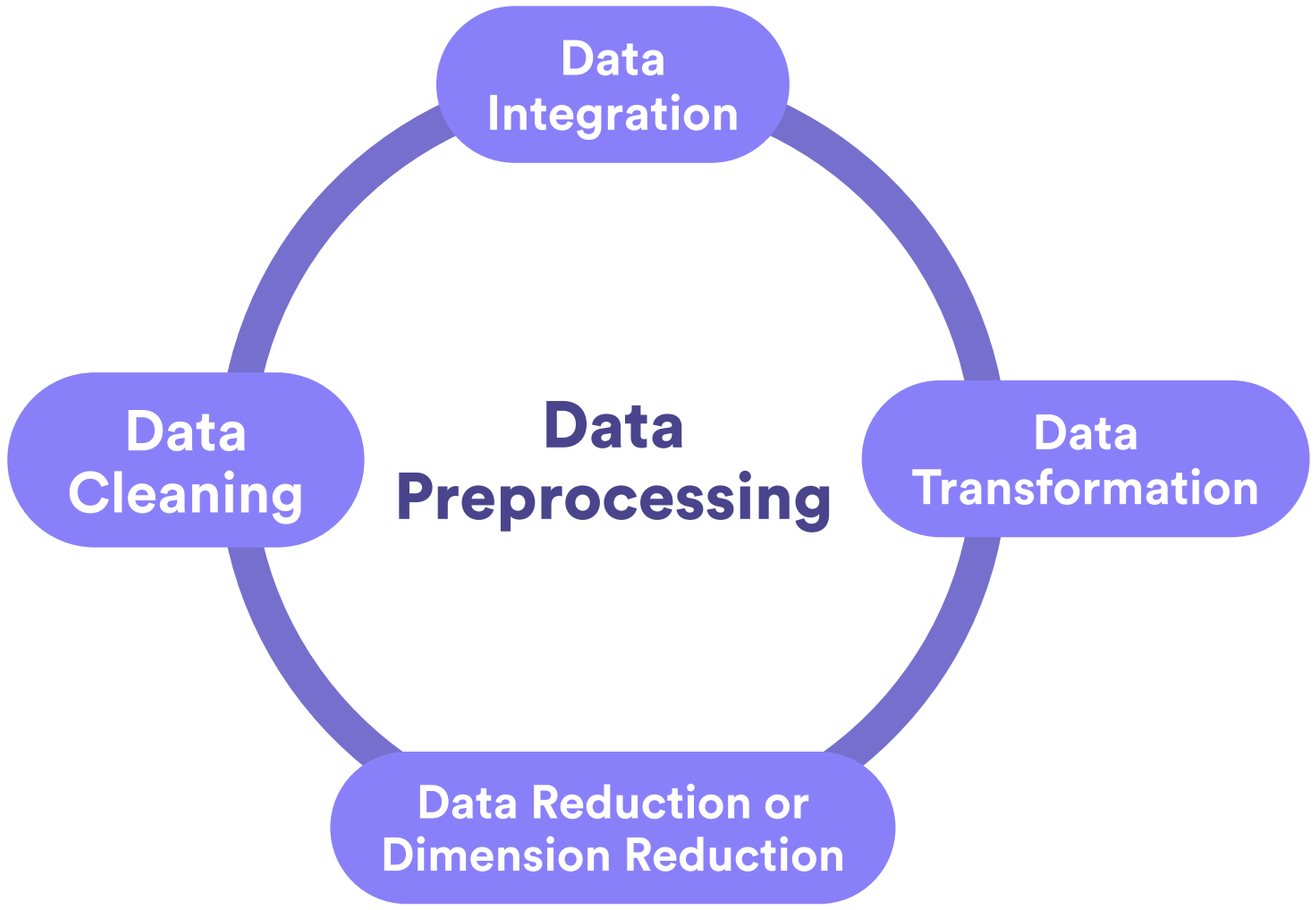
# Data Preprocessing

## Concepts:

1. Understanding different data types
2. Data pre-processing (cleaning, scaling, missing values, handling missing data, encoding categorical variables).
3. Data normalisation and standardisation

## Questions:

1. What are different data pre-processing techniques? How do they impact model performance?
2. Clean a small dataset (e.g., from Kaggle) using techniques learned.



DAY 6

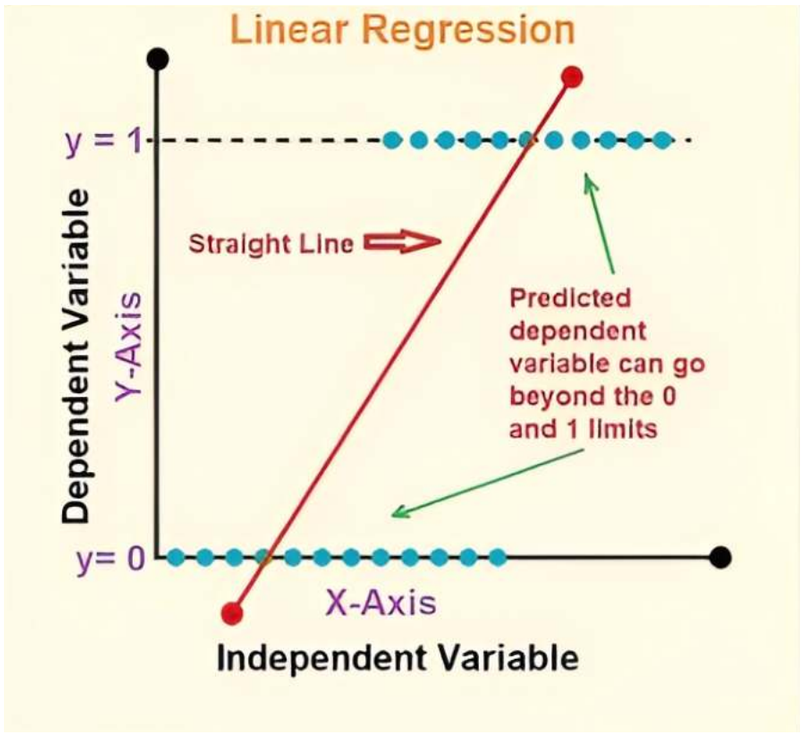
# Supervised Learning - Linear Regression

## Concepts:

1. Linear Regression, model fitting, classification algorithms
2. Multiple Linear Regression
3. Evaluation metrics (mean squared error, R-squared).

## Questions:

1. How does Linear Regression work? What are the limitations of Linear Regression?
2. Implement Linear Regression in Python (e.g., using scikit-learn) to predict a value in your chosen dataset.



DAY 7

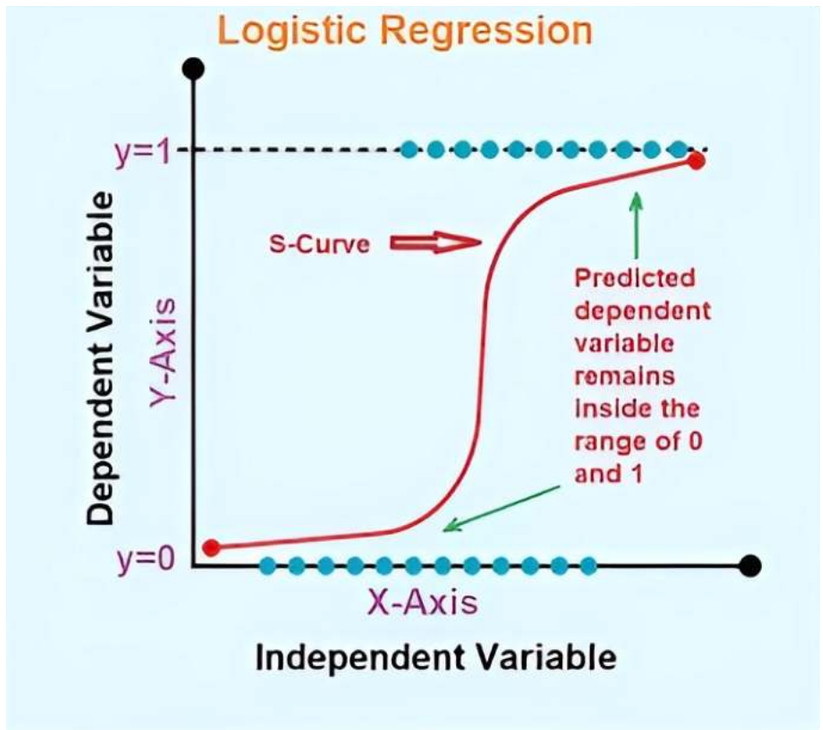
# Supervised Learning - Logistic Regression

## Concepts:

1. Logistic Regression, classification problems, decision boundaries.
2. Evaluation metrics (mean squared error, R-squared).

## Questions:

1. How does Logistic Regression work? How can we evaluate the performance of a classification model?
2. Implement Logistic Regression in Python to classify data points in your dataset. Visualise the decision boundary.



DAY 8

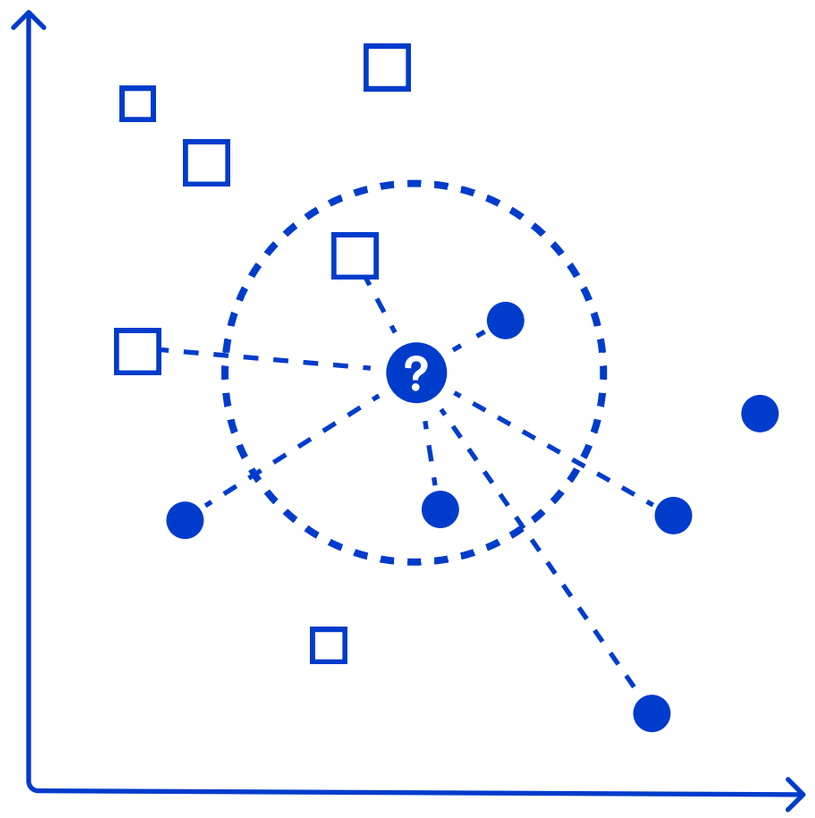
# Supervised Learning - K Nearest Neighbors

## Concepts:

1. K-Nearest Neighbors (KNN) algorithm
2. k-value selection, similarity measures.

## Questions:

1. How does KNN work? How does the choice of k influence the model?
2. Implement KNN in Python to predict a category in your dataset. Experiment with different k values.



DAY 9

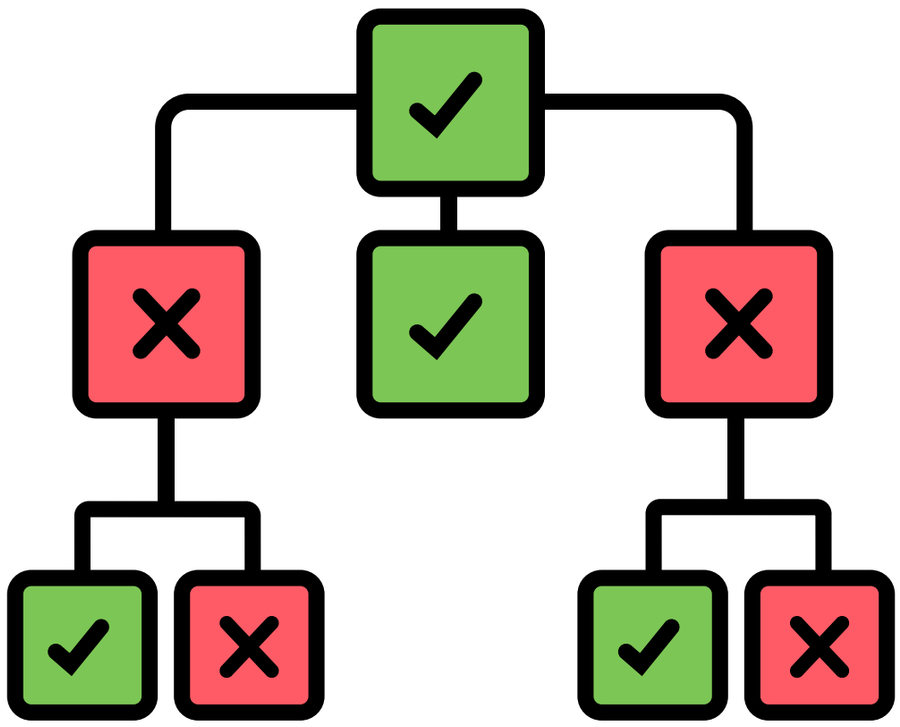
# Supervised Learning - Decision Trees

## Concepts:

1. Decision Trees
2. Information gain, entropy, tree pruning
3. Feature Importance

## Questions:

1. How do Decision Trees make predictions? What are the benefits and drawbacks of Decision Trees?
2. Build a Decision Tree in Python to classify data points in your dataset. Analyze the feature importance.



DAY 10

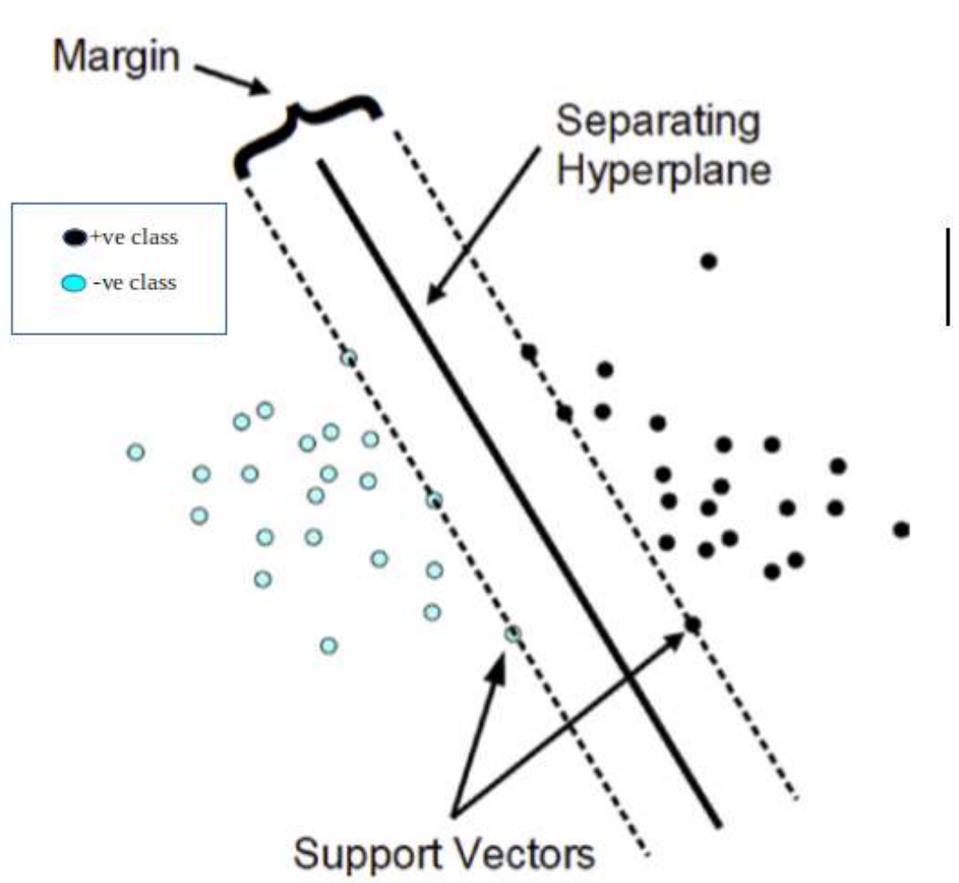
# Supervised Learning - Support Vector Machines

## Concepts:

1. Support Vector Machines (SVM) Algorithm
2. Hyperplanes, kernels.

## Questions:

1. How do SVMs work? What are different types of kernels used in SVMs?
2. Implement SVM in Python to classify data points in your dataset. Explore the impact of different kernels.



DAY 11

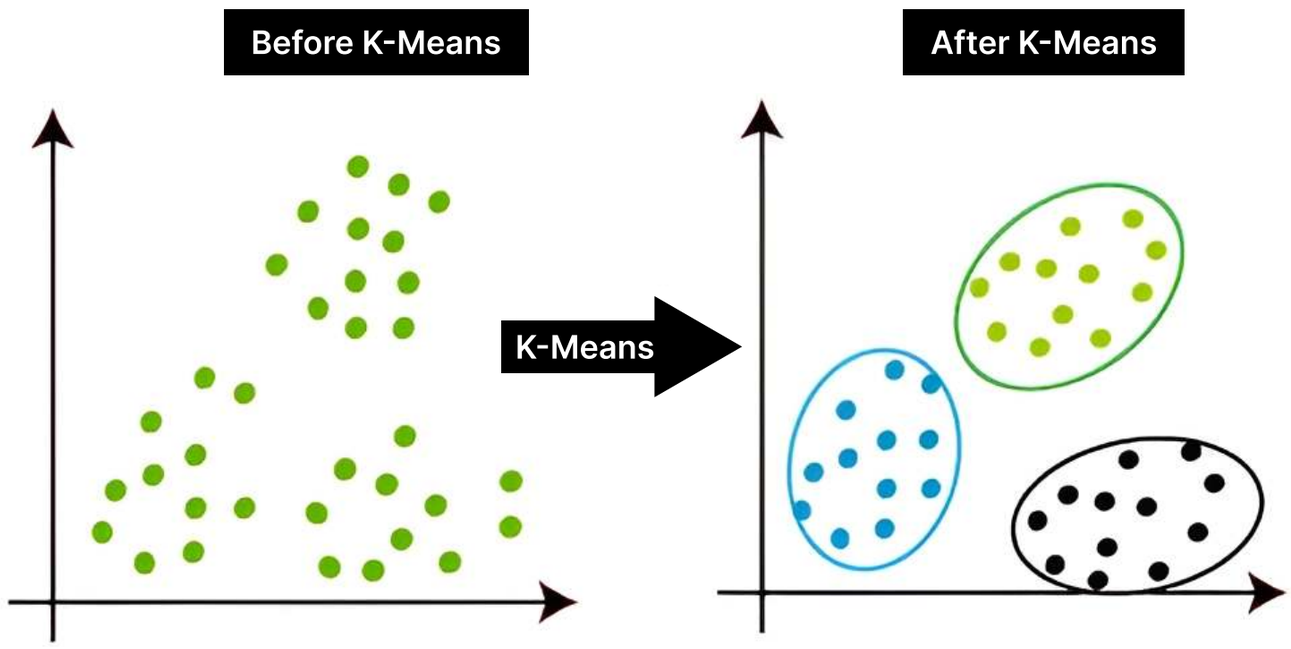
# Unsupervised Learning - K Means Clustering

## Concepts:

1. Clustering Algorithm
2. K-Means algorithm, k-value selection
3. Distance metrics: Euclidean, Manhattan, etc.

## Questions:

1. What are the different types of clustering algorithms? How does k-means clustering work?
2. Perform k-means clustering on your dataset to identify groups of similar data points. Visualise the clusters.



DAY 12

# Unsupervised Learning - Hierarchical Clustering

## Concepts:

1. Hierarchical Clustering: Agglomerative and Divisive approaches.
2. Distance metrics: Euclidean, Manhattan, etc.
3. Linkage methods: Single, Complete, Average, Ward.
4. Dendrograms: Interpreting and visualizing cluster hierarchies.

## Questions:

1. How does Hierarchical Clustering differ from other clustering methods?
2. When would you choose Hierarchical Clustering over other methods?
3. Perform Both agglomerative and divisive clustering on your dataset to identify groups of similar data points.

DAY 13

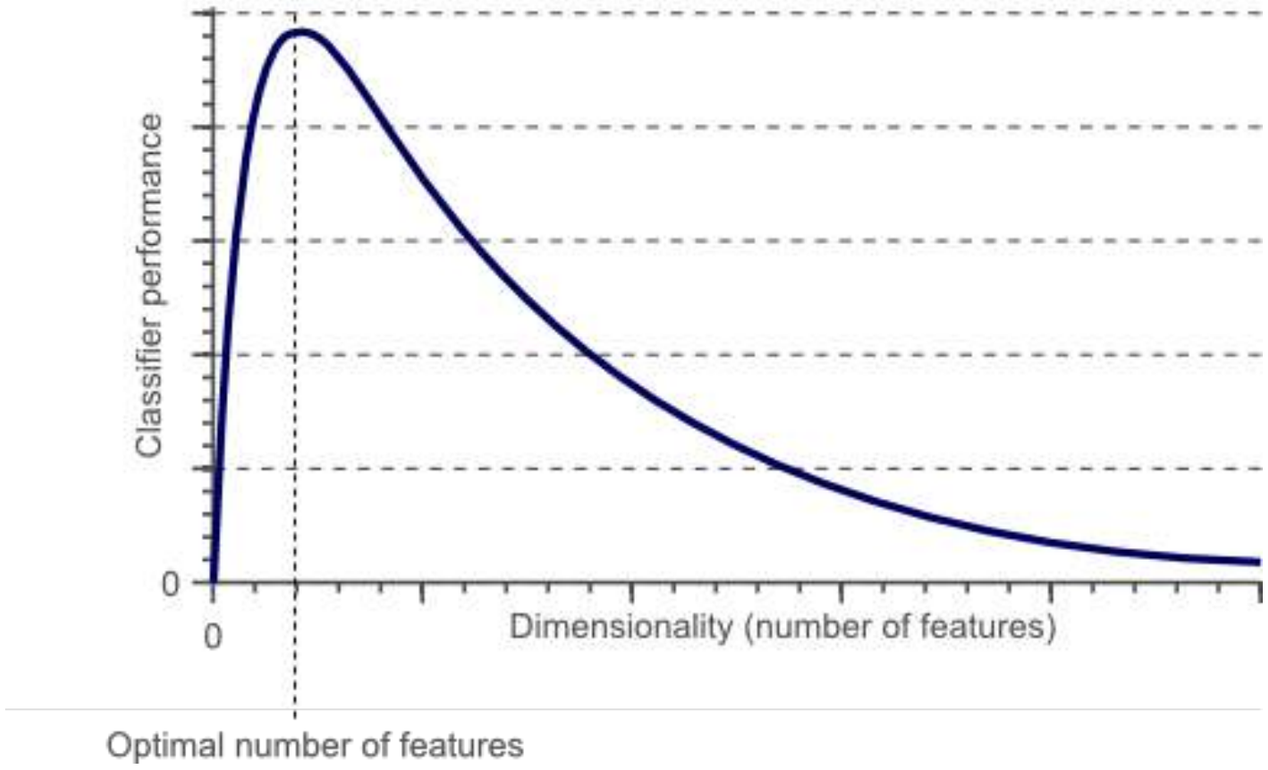
# Dimensionality Reduction

## Concepts:

1. Dimensionality Reduction
2. Principal Component Analysis (PCA),
3. Feature selection.

## Questions:

1. Why is dimensionality reduction important? How does PCA work?
2. Apply PCA to your dataset to reduce dimensionality. Analyze the impact on model performance.



DAY 14

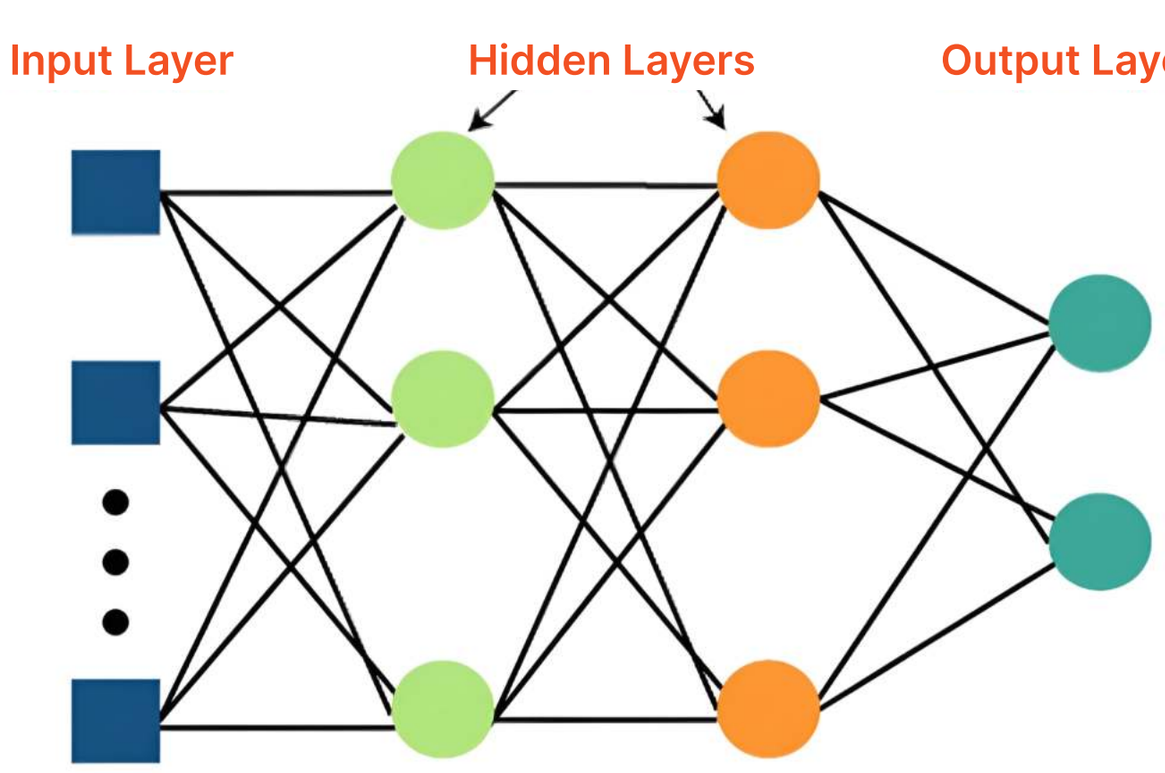
# Introduction to Neural Networks

## Concepts:

1. Basic building blocks of Neural Networks- neurons, activation functions
2. Layers and Architectures of Neural Networks

## Questions:

1. Implement a simple Perceptron or Multi-Layer Perceptron (MLP) in Python to solve a linear classification problem.
2. Explore different activation functions and their impact on network performance.



DAY 15-16

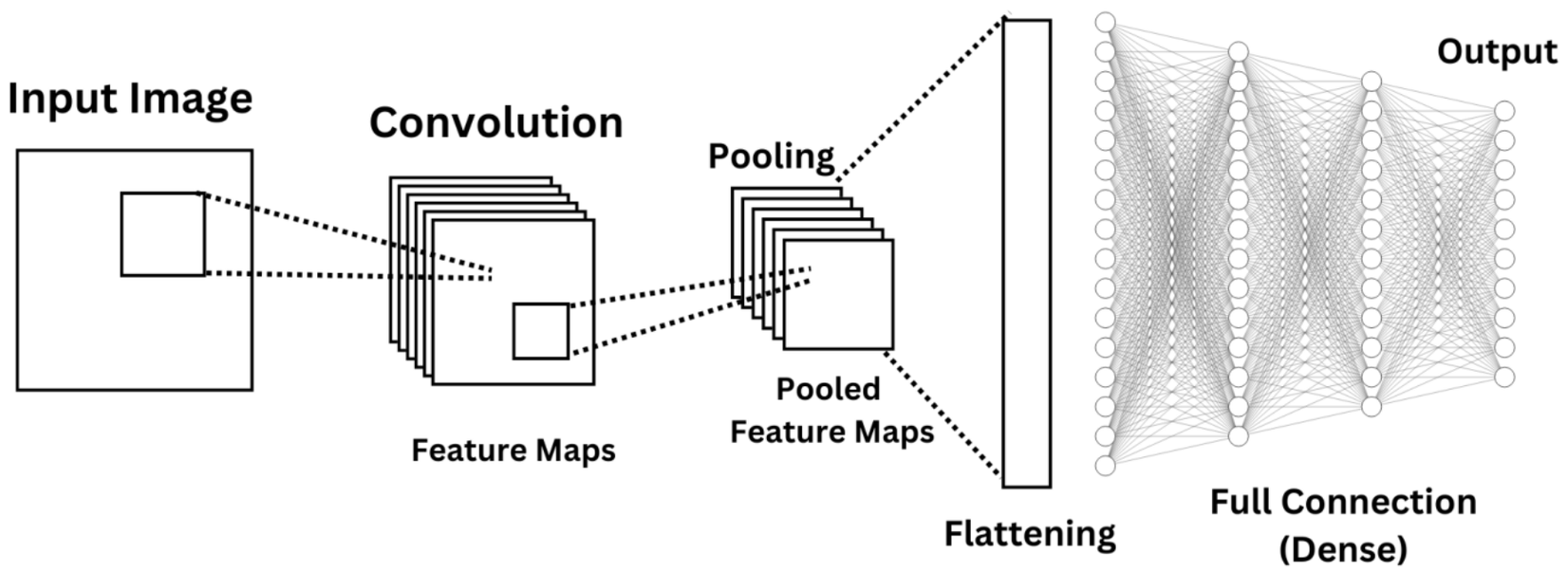
# Convolutional Neural Networks (CNNs)

## Concepts:

1. Key components of CNNs- convolutional layers, pooling layers, and activation functions.

## Questions:

1. Build a basic CNN in Python (e.g., using TensorFlow or PyTorch) to classify handwritten digits from the MNIST dataset.
2. Apply data augmentation techniques to improve model performance.



DAY 17-18

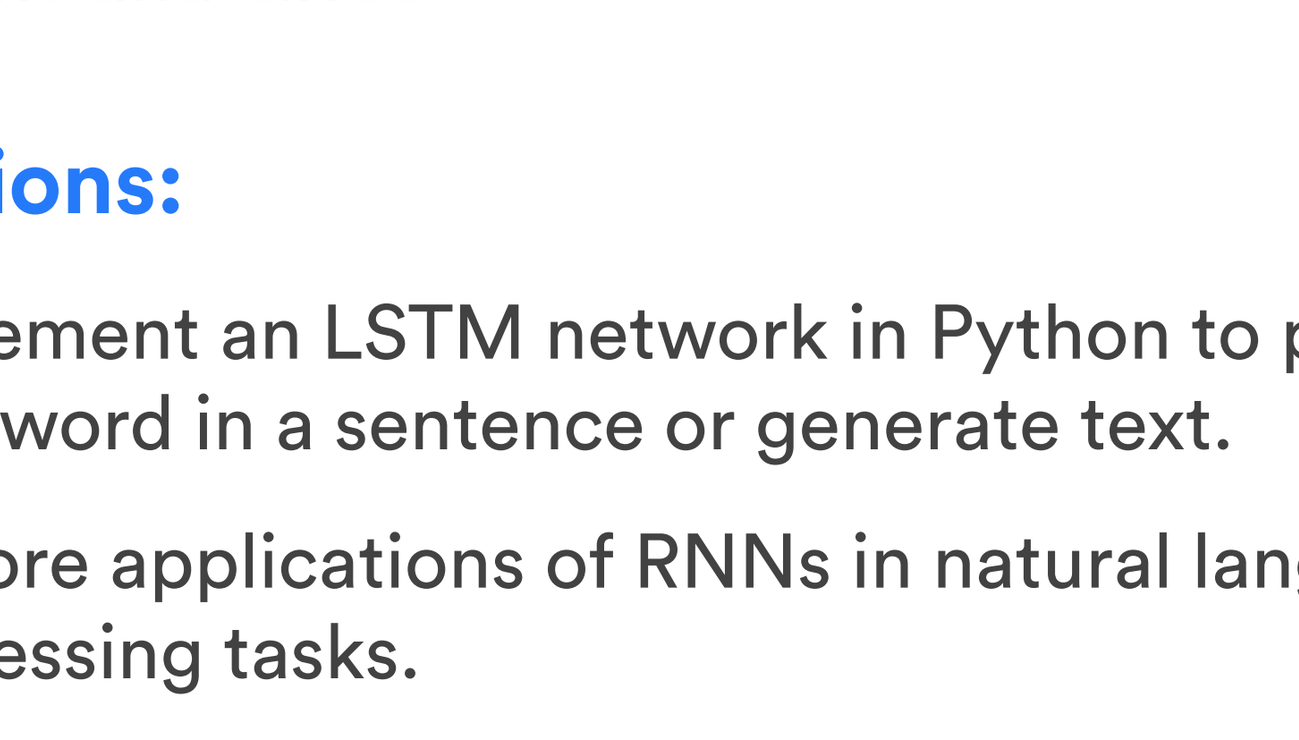
# Recurrent Neural Networks (RNNs)

## Concepts:

1. RNNs
2. LSTMs and GRUs

## Questions:

1. Implement an LSTM network in Python to predict the next word in a sentence or generate text.
2. Explore applications of RNNs in natural language processing tasks.



DAY 19

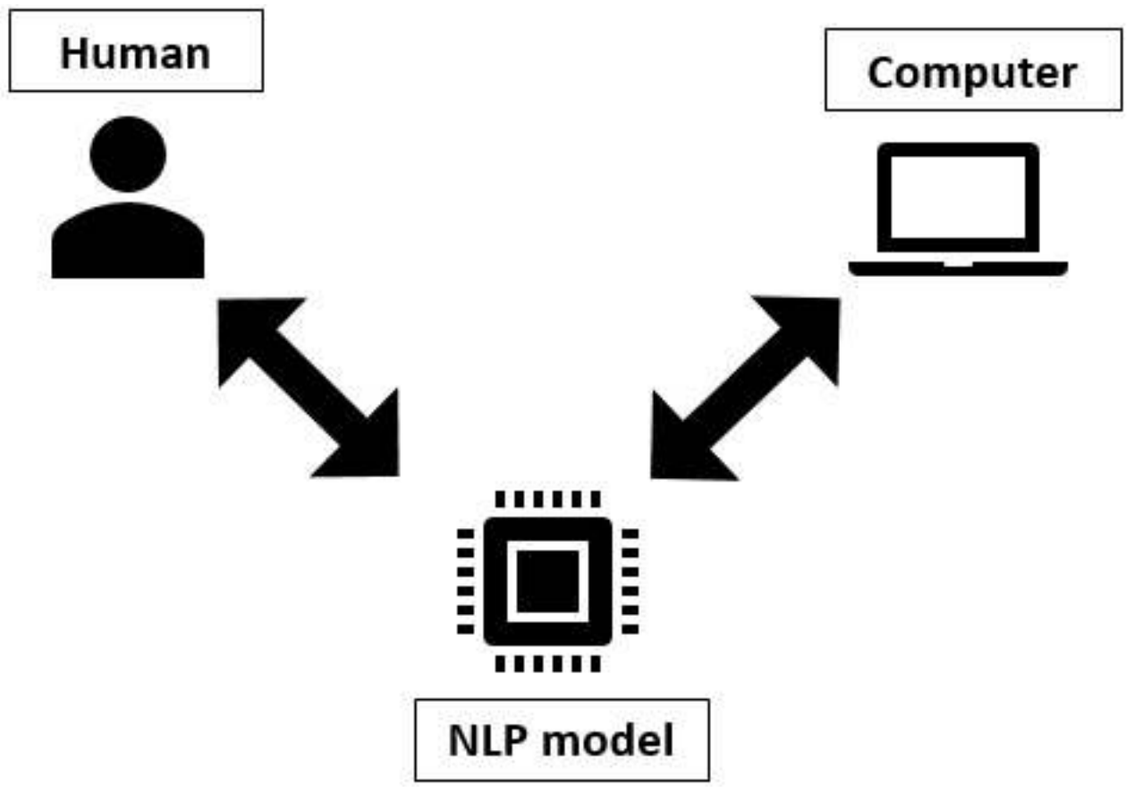
# Natural Language Processing (NLP) Basics

## Concepts:

1. Text preprocessing techniques (tokenization, stemming, lemmatization).
2. Techniques for text representation (Bag of Words, TF-IDF)

## Questions:

1. Perform text preprocessing and analysis on a sample text dataset.



DAY 20-21

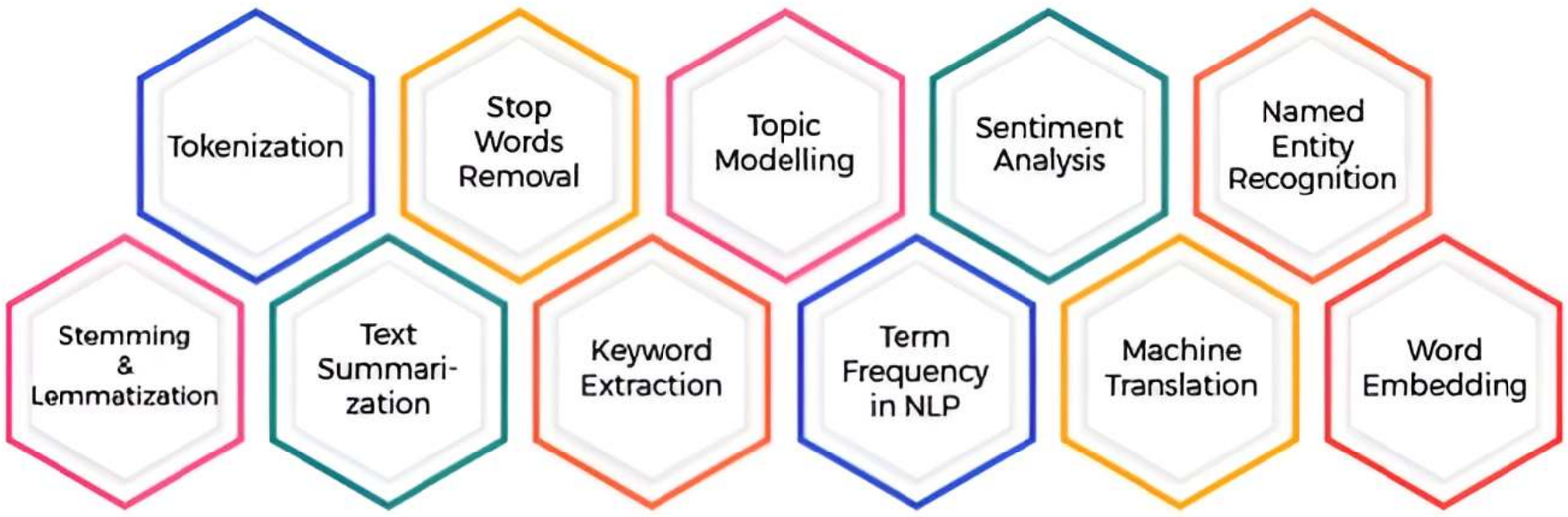
# Advanced NLP Techniques

## Concepts:

1. Advanced NLP techniques (word embeddings, sequence-to-sequence models).
2. Pre-trained language models (BERT, GPT).
3. Sentiment analysis, named entity recognition, and text classification.

## Questions:

1. Implement an advanced NLP model for sentiment analysis or text classification.



DAY 22-23

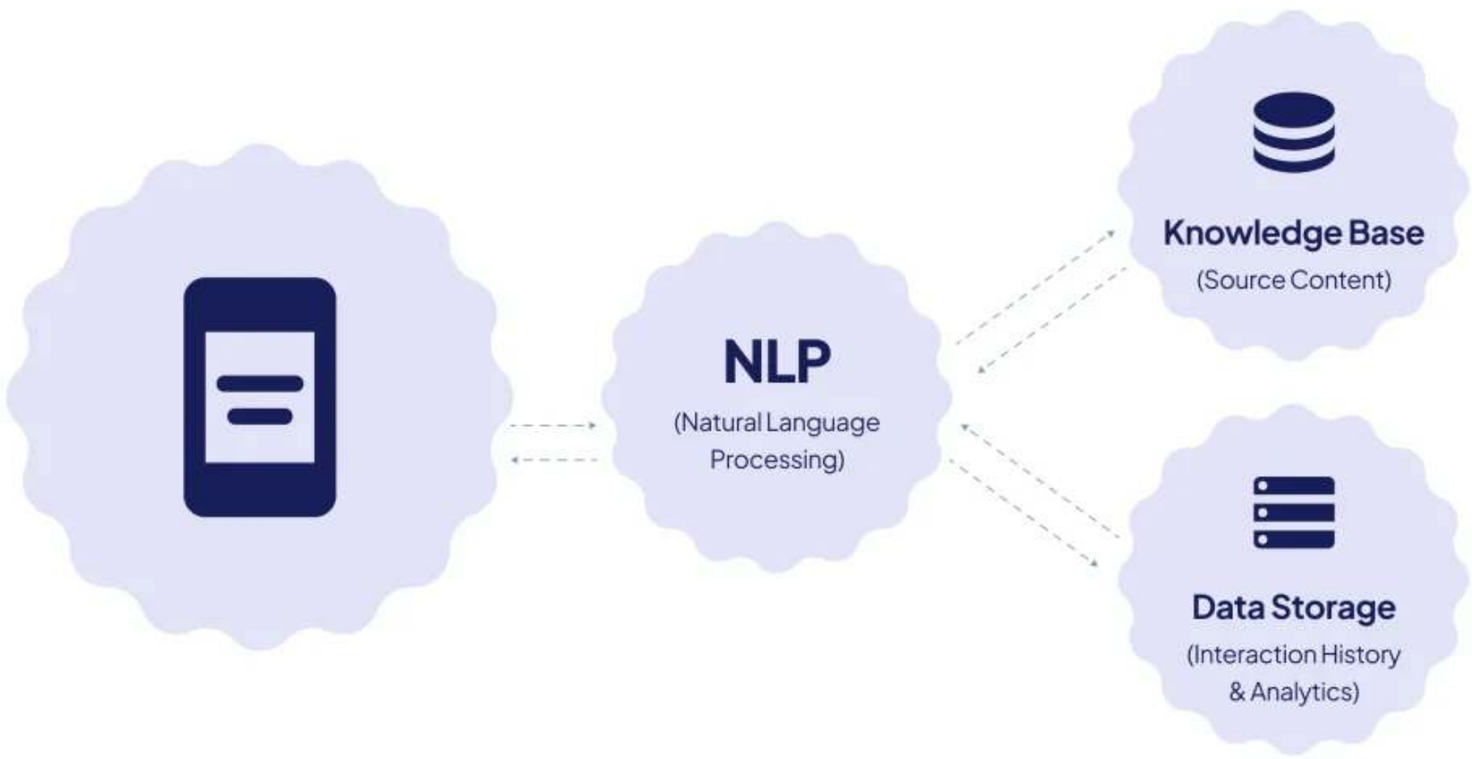
# NLP for Chatbots & Conversational AI

## Concepts:

1. Building chatbots
2. Designing conversational interfaces, and incorporating NLP techniques.

## Questions:

1. Develop a simple chatbot using libraries like Rasa or Dialogflow.
2. Explore advanced conversational AI techniques like dialogue management and personalization.



DAY 24

# Advanced ML Topics

## Concepts:

1. Model evaluation
2. Hyperparameter tuning and cross-validation.
3. Manual grid search, random search, and Bayesian optimization

## Questions:

1. Implement grid search and random search on a simple model to optimize a specific hyperparameter (e.g., learning rate). Analyze the impact on performance and compare the techniques.
2. Implement early stopping to prevent overfitting and compare its impact on performance.

DAY 25

# Machine Learning Ops

## Concepts:

1. Model serialization
2. Flask
3. Containerisation, Docker
4. Deployment Platforms - Heroku, AWS SageMaker

## Questions:

1. Prepare your model for deployment by serializing it. Explore model deployment platforms like Heroku or AWS SageMaker.
2. How can you scale your Flask API to handle increasing traffic?
3. What are the key components of a Dockerfile for packaging your ML model?

DAY 26-30

# Project Implementation and Portfolio Building

- Choose a machine learning project idea or find a dataset to work on.
- Implement the project from start to finish, including data preprocessing, model building, evaluation, and deployment (if applicable).
- Document the project and create a portfolio showcasing your work.
- Present your completed project, including code, documentation, and results, to peers or potential employers for feedback.

## 💡 Project Ideas:

### Easy:

1. **Movie Recommendation System:** Build a system that recommends movies to users based on their past viewing history or ratings (Movielens dataset).
2. **Spam Email Classification:** Classify emails as spam or not spam using text analysis (public email datasets).

3. **Handwritten Digit Recognition:** Train a model to recognize handwritten digits from images (MNIST dataset).

**Medium:**

4. **Sentiment Analysis:** Analyze the sentiment of text (positive, negative, neutral) in reviews, tweets, or social media posts (Twitter Sentiment 140 dataset).
5. **Image Segmentation:** Segment images into different regions (foreground, background) or objects (medical images, satellite imagery).

**Hard:**

6. **Natural Language Generation:** Train a model to generate human-quality text, like poems, code, scripts, or emails (GPT-2, Jurassic-1 Jumbo dataset).
7. **Music Composition:** Create music pieces based on existing songs or styles using deep learning techniques (Magenta project).
8. **Medical Diagnosis Support:** Assist doctors in diagnosing diseases by analyzing medical images (X-rays, CT scans) or other data.

BOSSCODER ACADEMY

# WHY BOSSCODER?

- **750+** Alumni placed at Top Product-based companies.
- More than **136% hike** for every **2 out of 3** working professional.
- Average package of **24LPA.**

> The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**
>
> **Lavanya**
> Meta

> Course is very well **structured** and **streamlined** to crack any **MAANG** company
>
> **Rahul .**
> Google

**EXPLORE MORE**