Introduction to Machine Learning & Generative AI

A Technical Deep Dive into Predictions, Neural Networks, and Modern AI

AI/ML FUNDAMENTALS

MENTOR DISCUSSION

Session Agenda

01

Machine Learning Foundations

Core concepts and real-world importance

02

Prediction & Regression

How models make forecasts and learn patterns

03

Neural Networks & Perceptrons

Architecture, training, and key differences

04

Generative AI

Modern applications and transformative impact

What is Machine Learning?

Definition

Machine Learning enables computers to learn patterns from data without explicit programming, improving performance through experience.

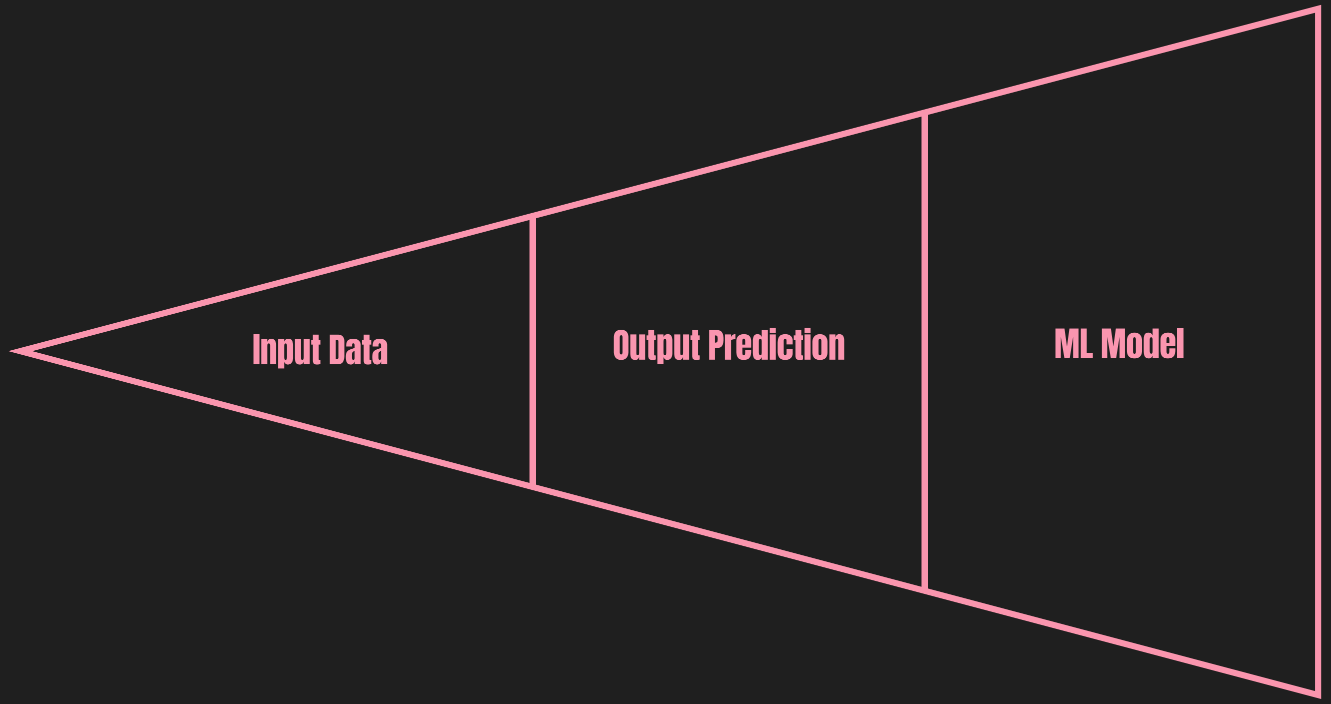
Why It Matters

ML powers everything from medical diagnosis to autonomous vehicles, transforming industries by uncovering insights humans might miss.

Real-World Example

Netflix's recommendation engine analyzes viewing habits to suggest content, keeping users engaged through predictive algorithms.

Prediction in Machine Learning

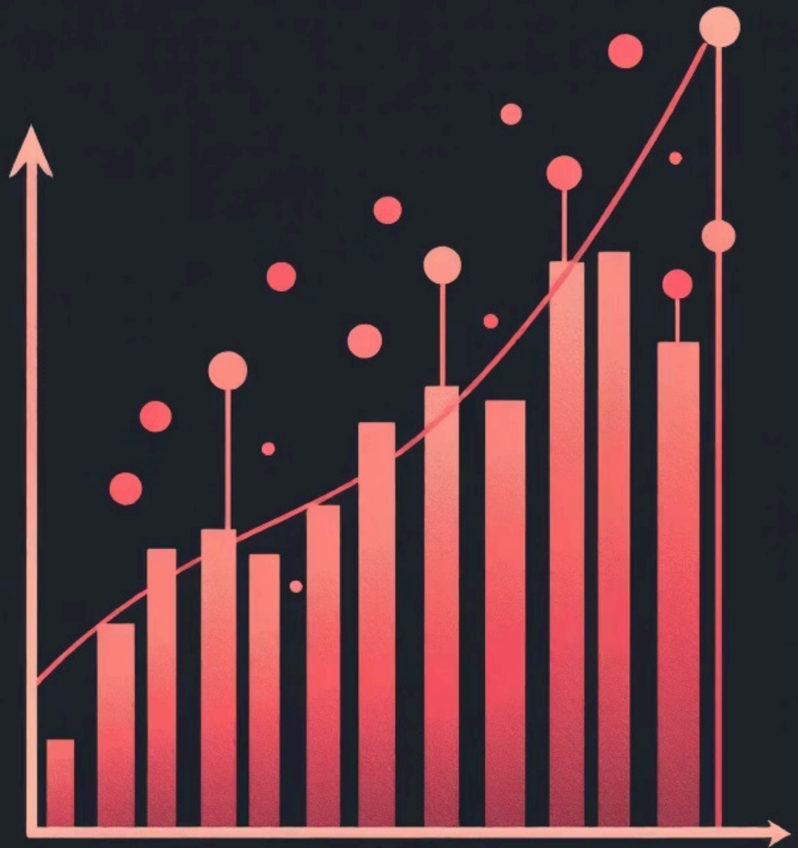


How Prediction Works

Prediction is the process where a trained model takes new input data and generates an output based on learned patterns.

- Input: Features like age, income, or past behavior
- Model: Trained algorithm that processes inputs
- Output: Predicted value or classification

Example: Predicting house prices based on location, size, and amenities using historical sales data.



Understanding Regression

Linear Regression

Models the relationship between a single input variable and output using a straight line. Perfect for simple cause-effect relationships.

Multiple Regression

Extends linear regression to handle multiple input variables simultaneously, capturing complex real-world scenarios.

Common Use Cases

Sales forecasting, stock price prediction, risk assessment, customer lifetime value estimation, and demand planning.

Regression in Action

The Formula

$$y = mx + c$$

Where:

- **y** = predicted output
- **m** = slope (weight)
- **x** = input feature
- **c** = y-intercept (bias)

Python Implementation

```
from sklearn.linear_model import LinearRegression

# Training data
X = [[1], [2], [3], [4], [5]]
y = [2, 4, 6, 8, 10]

# Create and train model
model = LinearRegression()
model.fit(X, y)

# Make prediction
prediction = model.predict([[6]])
print(prediction) # Output: [12]
```

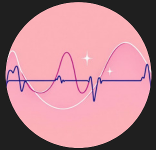
Neural Networks Fundamentals

Artificial Neuron

Mimics biological neurons by receiving inputs, applying weights, and producing an output through an activation function.

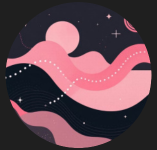
Layer Architecture

Networks consist of input layers (receive data), hidden layers (extract features), and output layers (make predictions).



Activation Functions

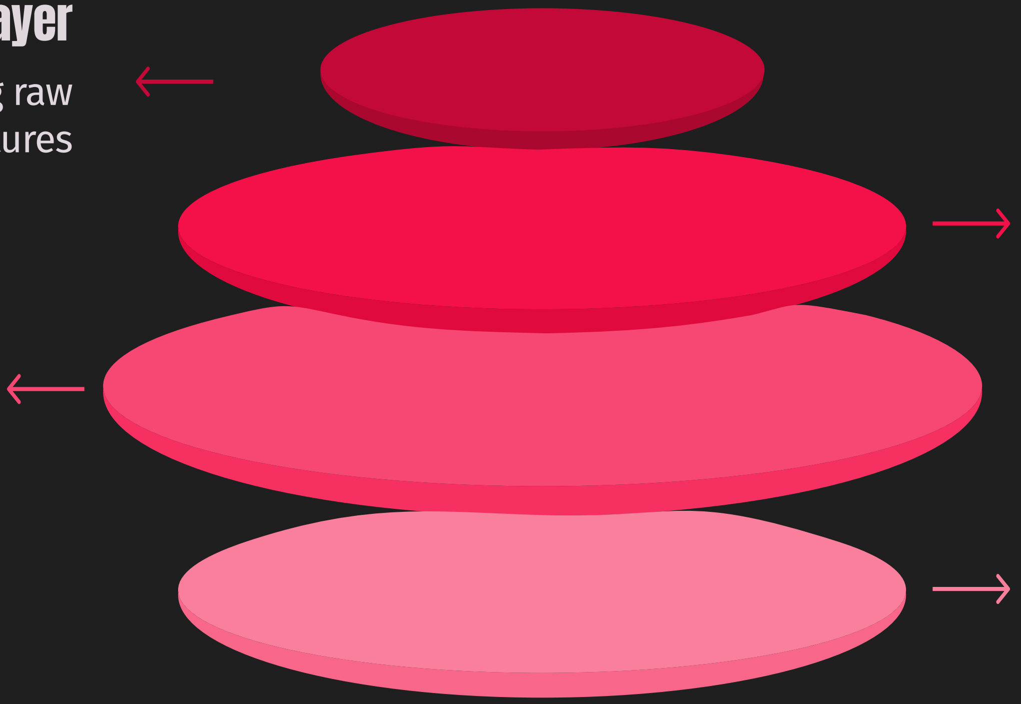
Non-linear functions like ReLU, sigmoid, and tanh introduce complexity, enabling networks to learn intricate patterns.



Training Process

Networks learn through backpropagation, adjusting weights iteratively to minimize prediction error using gradient descent.

Neural Network Architecture



Input Layer

3 nodes receiving raw features

Hidden Layer 1

4 neurons extracting features

Hidden Layer 2

4 neurons refining representations

Output Layer

2 nodes producing predictions

The Perceptron Model

Simplest Neural Unit

The perceptron is a single-layer neural network that performs binary classification by learning a linear decision boundary.

How It Works

1. Receives multiple weighted inputs
2. Sums inputs: $\sum(w_i \cdot x_i) + b$
3. Applies activation function (step function)
4. Outputs 0 or 1 based on threshold

Core Equation

$$y = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Perceptron vs Neural Network

Aspect	Perceptron	Neural Network
Layers	Single layer	Multiple layers (deep)
Complexity	Linear problems only	Non-linear patterns
Activation	Step function	ReLU, sigmoid, tanh
Learning	Simple weight updates	Backpropagation
Use Case	Basic classification	Image, text, speech recognition

Key Limitation: Perceptrons cannot solve problems that aren't linearly separable, like XOR. Neural networks overcome this with hidden layers.