**Machine Learning**

**Name:** Prince Maurya

**Mobile No:** 9987742369

**Emil-ID:** [princemaurya8879@gmail.com](mailto:princemaurya8879@gmail.com)

**Basic Fundamentals of AI & ML**

**What is AI (Artificial Intelligence)?**

Artificial Intelligence (AI) is the field of computer science that enables machines to simulate human intelligence. AI systems can learn, reason, problem-solve, perceive, and understand language. AI can be classified into:

1. **Narrow AI (Weak AI)** – AI designed for specific tasks (e.g., Siri, Google Assistant).
2. **General AI (Strong AI)** – AI that can perform any intellectual task like a human (still theoretical).
3. **Super AI** – AI that surpasses human intelligence (a future concept).

**What is ML (Machine Learning)?**

Machine Learning (ML) is a subset of AI that enables systems to learn from data and improve their performance without being explicitly programmed. ML models identify patterns in data and make predictions.

**Types of Machine Learning:**

1. **Supervised Learning** – The model is trained on labeled data (e.g., spam email detection).
2. **Unsupervised Learning** – The model finds patterns in unlabeled data (e.g., customer segmentation).
3. **Reinforcement Learning** – The model learns by trial and error, getting rewards for good actions (e.g., game-playing AI).

|  |  |  |
| --- | --- | --- |
| **Feature** | **Artificial Intelligence (AI)** | **Machine Learning (ML)** |
| **Definition** | AI is a broad concept where machines mimic human intelligence | ML is a subset of AI that learns from data to make decisions |
| **Goal** | To create smart systems that can perform complex tasks | To develop models that can learn and improve from experience |
| **Scope** | Includes ML, deep learning, expert systems, NLP, robotics, etc. | Focuses only on learning from data and improving accuracy |
| **Dependency** | AI can exist without ML | ML needs AI to exist |
| **Example** | Chatbots, self-driving cars, voice assistants (Alexa, Siri) | Spam filters, recommendation systems, fraud detection |

**Machine learning** is the science of getting computers to learn and act like humans do.

**Machine Learning Application:**

* Self-Driving Car
* Gamified Learning and Education
* E-Commerce Websites
* Medical Diagnosis

**Machine Learning Life Cycle**

Step1. Data Collection

Step2. Data Preparation

Step3. Data Wrangling

Step4. Data Modelling

Step5. Model Training

Step6. Model Testing

Step7. Deployment

**Step1. Data Collection**

Goal over here is to gather as much as relevant data as possible

* Identify various source of information
* Gather data
* Combine the data acquired from various data source

**Step2. Data Preparation**

* Data preparation deals with exploration of your data to generate far better results.

**Step 3. Data Wrangling**

Data wrangling is the process of cleaning and converting raw data into a useable format.

* Filtering / cleaning up of raw data
* Filtering Noise
* Recognizing and removing outliers
* Removing or filling the missing values

**Step 4. Data Modelling**

Data modelling is the steps in which we take the data and select a machine learning algorithm to built a model

* Selecting machine learning algorithm
* Building the models
* Validating the results

**Step 5. Model Training**

* A machine learning training model is a process in which a machine learning (ML)

Algorithm is fed with sufficient training data to learn from.

**Step 6. Model Testing**

* This stage of the machine learning lifecycle involves checking for the accuracy of the model by providing with the inputs that are unseen.

**Step 7. Depolyment**

* This is final step in the machine learning life cycle where we have a brilliant model ready to go to production.

**Best Languages for Machine Learning**

* Python [Most popular Language for ML]
* Java
* R
* JavaScript
* Scala

**Install a Python IDE**

**Artificial Intelligence**

Artificial intelligence is the science and engineering of making computers capable of performing tasks that typically human intelligence.

Based on the capabilities, AI classified as:

* Applied AI (Weak AI)
* Generalized AI (Strong AI)

**Types of Machine Learning**

Machine Learning (ML) is divided into three main types based on how the model learns from data:

1. **Supervised Learning**

* The model is trained using **labeled data** (input-output pairs).
* The algorithm learns from past data to **predict outcomes for new data**.
* Example:
  + **Email Spam Detection** → Emails are labeled as **spam** or **not spam**, and the model learns to classify new emails accordingly.

**Types of Supervised Learning:**

1. **Classification** → Predicts categories (e.g., spam vs. not spam).
2. **Regression** → Predicts continuous values (e.g., predicting house prices).

✅ **Example Algorithms:**

* Linear Regression
* Decision Trees
* Random Forest
* Support Vector Machines (SVM)

**2. Unsupervised Learning**

* The model is trained on **unlabeled data** (no predefined categories).
* It finds **hidden patterns and structures** in the data.
* Example:
  + **Customer Segmentation** → Grouping customers based on purchasing behavior without knowing predefined groups.

**Types of Unsupervised Learning:**

1. **Clustering** → Groups similar data points (e.g., customer segmentation).
2. **Association Rule Learning** → Finds relationships between variables (e.g., market basket analysis).

✅ **Example Algorithms:**

* K-Means Clustering
* Hierarchical Clustering
* Apriori Algorithm

**3. Reinforcement Learning (RL)**

* The model **learns by trial and error** through **rewards and penalties**.
* The agent takes actions in an **environment** to maximize a **reward** over time.
* Example:
  + **Game Playing AI (AlphaGo, Chess AI)** → The AI learns the best moves by playing thousands of games and improving its strategy.

**Key Concepts in RL:**

* **Agent** → The learner or decision-maker (e.g., a robot).
* **Environment** → Where the agent interacts (e.g., a chessboard).
* **Action** → What the agent does (e.g., moving a chess piece).
* **Reward** → Positive or negative feedback (e.g., winning or losing a game).

✅ **Example Algorithms:**

* Q-Learning
* Deep Q-Networks (DQN)
* Proximal Policy Optimization (PPO)

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Supervised Learning** | **Unsupervised Learning** | **Reinforcement Learning** |
| **Training Data** | Labeled Data | Unlabeled Data | Reward-Based Learning |
| **Main Task** | Predict outcomes | Find hidden patterns | Learn optimal actions |
| **Example** | Spam detection | Customer segmentation | Self-playing Chess AI |
| **Use Case** | Medical diagnosis | Market basket analysis | Robotics, Gaming |
| **Algorithms** | Decision Trees, SVM | K-Means, DBSCAN | Q-Learning, DQN |

**1. Regression Algorithms (Predict Continuous Values)**

**1.1 Linear Regression**

**Concept:**

* Used for predicting a continuous output based on input variables.
* Finds the best-fit line (y = mx + c) that minimizes the error between predicted and actual values.
* Uses **Least Squares Method** to reduce the error.

**Formula:**

y=β0​+β1​x+ε

Where:

* y = predicted output
* x = input feature
* β0​ = intercept
* β1​ = slope
* ε = error term

**Example:**

* Predicting **house prices** based on square footage.

✅ **Advantages:**  
✔️ Simple & easy to implement.  
✔️ Works well for linearly related data.

❌ **Disadvantages:**  
❌ Fails for complex relationships (non-linear data).

**1.2 Multiple Linear Regression**

* Extends **Linear Regression** for multiple input features.
* Instead of a single feature **x**, it considers multiple variables **x1​,x2​,...,xn​​**.

**Formula:**

y=β0​+β1​x1​+β2​x2​+⋯+βn​xn​+ε

**Example:**

* Predicting **salary** based on experience, age, and education level.

✅ **Advantages:**  
✔️ Can handle multiple input variables.

❌ **Disadvantages:**  
❌ Assumes **linear relationships** among features.

**1.3 Polynomial Regression**

* Extends Linear Regression by introducing polynomial (squared, cubic) terms to capture **non-linearity**.

**Formula:**

y=β0​+β1​x+β2​x2+β3​x3+⋯+βn​xn+ε

**Example:**

* Predicting **sales trends** that follow a curve instead of a straight line.

✅ **Advantages:**  
✔️ Works well for non-linear data.

❌ **Disadvantages:**  
❌ Overfitting risk if the polynomial degree is too high.

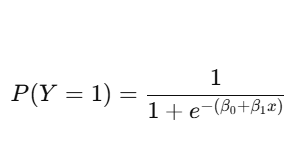
**2. Classification Algorithms (Predict Categorical Values)**

**2.1 Logistic Regression**

**Concept:**

* Used for **binary classification** (e.g., **Yes/No, 0/1** problems).
* Unlike Linear Regression, it outputs a probability (0 to 1) using the **Sigmoid Function**.

**Sigmoid Function:**



Where:

* e = Euler's number (2.718)
* P(Y=1) = probability of the positive class

**Example:**

* Predicting whether an email is **spam (1) or not spam (0)**.

✅ **Advantages:**  
✔️ Works well for **binary classification** problems.

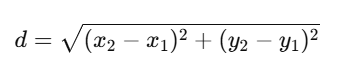
❌ **Disadvantages:**  
❌ Assumes a **linear relationship** between input features and log-odds.

**2.2 K-Nearest Neighbors (KNN)**

**Concept:**

* A **non-parametric** algorithm that classifies a new data point based on the majority of its **K-nearest neighbors**.
* Uses **Euclidean distance** to find the nearest points.

**Formula:**



Where:

* d = distance between two points

**Example:**

* Classifying whether a customer will **buy (1) or not buy (0)** a product based on their past behavior.

✅ **Advantages:**  
✔️ Simple & effective.  
✔️ Works well for small datasets.

❌ **Disadvantages:**  
❌ Computationally expensive for large datasets.

**2.3 Support Vector Machine (SVM)**

**Concept:**

* Finds the **best decision boundary** (hyperplane) that maximizes the **margin** between two classes.

**Types of SVM:**

1. **Linear SVM** → Used when data is linearly separable.
2. **Non-Linear SVM** → Uses **Kernel Trick** to handle non-linearity.

**Example:**

* Classifying **handwritten digits** as 0-9.

✅ **Advantages:**  
✔️ Works well with high-dimensional data.  
✔️ Handles non-linearly separable data using kernels.

❌ **Disadvantages:**  
❌ Computationally expensive for large datasets.

**2.4 Decision Tree**

**Concept:**

* A **tree-like structure** where each node represents a decision.
* Uses **Entropy & Information Gain** to split nodes.

**Example:**

* **Loan approval system** (classifying whether a loan should be approved or not).

✅ **Advantages:**  
✔️ Easy to interpret & visualize.

❌ **Disadvantages:**  
❌ **Overfitting risk** (solved by pruning).

**2.5 Random Forest**

**Concept:**

* **Ensemble method** that builds multiple Decision Trees and combines their outputs.
* Reduces overfitting compared to a single Decision Tree.

**Example:**

* **Credit card fraud detection**.

✅ **Advantages:**  
✔️ High accuracy & reduces overfitting.

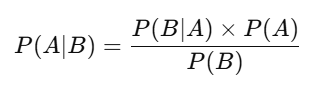
❌ **Disadvantages:**  
❌ Computationally expensive.

**2.6 Naïve Bayes**

**Concept:**

* Based on **Bayes’ Theorem**, assuming independence between features.
* Works well with **text classification** problems.

**Formula (Bayes' Theorem):**



Where:

* P(A∣B)) = Probability of A given B
* P(B∣A) = Probability of B given A
* P(A) = Prior probability of A
* P(B) = Prior probability of B

**Example:**

* **Spam email filtering**.

✅ **Advantages:**  
✔️ Fast and works well with text data.

❌ **Disadvantages:**  
❌ Assumes **independence** between features (which may not be true).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Type** | **Use Case Example** | **Pros** | **Cons** |
| Linear Regression | Regression | House price prediction | Simple, interpretable | Doesn't handle non-linear data |
| Logistic Regression | Classification | Spam detection | Works well for binary problems | Assumes linear relationship |
| KNN | Classification | Customer segmentation | Easy to implement | Slow for large datasets |
| SVM | Classification | Image recognition | Handles complex relationships | Computationally expensive |
| Decision Tree | Classification | Loan approval | Easy to interpret | Prone to overfitting |
| Random Forest | Classification | Fraud detection | Reduces overfitting | Requires more computation |
| Naïve Bayes | Classification | Text classification | Works well with text data | Assumes feature independence |

**What is Deep Learning?**

**Deep Learning** is a subset of **Machine Learning (ML)** that focuses on using **Neural Networks** with multiple layers (deep neural networks) to learn from data. It mimics the way the human brain processes information.

Deep Learning is effective for tasks like:  
✔️ Image recognition (e.g., facial recognition)  
✔️ Natural language processing (e.g., chatbots, voice assistants)  
✔️ Autonomous driving (e.g., self-driving cars)  
✔️ Medical diagnosis (e.g., cancer detection from scans)

**Types of Deep Learning Models**

**1. Feedforward Neural Networks (FNN)**

* The most basic type of neural network.
* Information moves **in one direction** (from input to output).
* Used for simple classification and regression tasks.
* Example: Predicting house prices based on features like size and location.

**2. Convolutional Neural Networks (CNN)**

* Designed for **image processing** and **computer vision** tasks.
* Uses **convolutional layers** to detect patterns (edges, textures, shapes).
* Used in **image classification, object detection, and facial recognition**.
* Example: Self-driving cars detecting pedestrians.

**3. Recurrent Neural Networks (RNN)**

* Used for **sequential data** like time series, speech, and text.
* Maintains a memory of previous inputs using **hidden states**.
* Example: Speech recognition, text prediction, machine translation.
* Variants:
  + **Long Short-Term Memory (LSTM)** – Handles long-term dependencies.
  + **Gated Recurrent Units (GRU)** – A simpler version of LSTMs.

**4. Autoencoders**

* Used for **unsupervised learning** and **feature extraction**.
* Composed of an **encoder** (compresses data) and **decoder** (reconstructs data).
* Used in **anomaly detection, image denoising, and compression**.
* Example: Detecting fraudulent transactions.

**5. Generative Adversarial Networks (GANs)**

* Used for **generating new data** that resembles real data.
* Has two neural networks:
  + **Generator** – Creates fake data.
  + **Discriminator** – Detects if data is real or fake.
* Used in **image generation, deepfake videos, and art creation**.
* Example: AI-generated human faces that don’t exist.

**6. Transformer Models**

* Used for **Natural Language Processing (NLP)**.
* Processes entire sequences of text at once instead of word by word.
* Example: **GPT (like me!), BERT, and T5**.

**7. Deep Reinforcement Learning (DRL)**

* Combines **Deep Learning** with **Reinforcement Learning**.
* AI learns by interacting with the environment and getting rewards.
* Used in **robotics, game playing (AlphaGo, Dota 2 AI), and self-driving cars**.

**What is scikit-learn (sklearn)?**

scikit-learn (or sklearn) is a powerful Machine Learning (ML) library in Python. It provides simple and efficient tools for data analysis and modeling. It is built on top of NumPy, SciPy, and Matplotlib.

**Why Use scikit-learn?**

🔹 **Easy to use** – Simple and intuitive API  
🔹 **Efficient** – Optimized for performance  
🔹 **Versatile** – Supports both supervised & unsupervised learning  
🔹 **Pre-built ML algorithms** – No need to implement from scratch  
🔹 **Data preprocessing** – Handle missing values, scaling, and feature engineering

**Key Features of scikit-learn**

**1. Supervised Learning Algorithms**

Used when data has **input (X) and output (Y)** labels.  
✔ **Regression**: Linear Regression, Logistic Regression, Decision Trees  
✔ **Classification**: Support Vector Machines (SVM), Random Forest, KNN

**2. Unsupervised Learning Algorithms**

Used for **clustering and pattern detection** when no labels are available.  
✔ **Clustering**: K-Means, DBSCAN, Hierarchical Clustering  
✔ **Dimensionality Reduction**: PCA, t-SNE

**3. Model Selection & Evaluation**

✔ Train-Test Split (train\_test\_split)  
✔ Cross-validation (cross\_val\_score)  
✔ Performance Metrics (accuracy\_score, mean\_squared\_error)

**4. Data Preprocessing**

✔ Handling Missing Values (SimpleImputer)  
✔ Feature Scaling (StandardScaler, MinMaxScaler)  
✔ Encoding Categorical Data (OneHotEncoder, LabelEncoder)

**5. Neural Networks**

✔ Implements basic Neural Networks using MLPClassifier and MLPRegressor

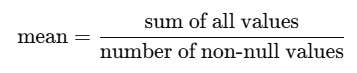
**Machine Learning Life Cycle**

**Step1. Data Collection(pandas)**

* Collects data from sources like CSV files, databases, APIs, or web scraping.
* Libraries used: pandas (for loading and handling datasets).

**Step2. Data Preparation(pandas, numpy)**

* Handles missing values, duplicates, and selects relevant features.
* Libraries used: pandas (for handling missing data), numpy (for numericaloperations).
* **What is the Mean?**
* The **mean** (or average) is calculated by:



**Step3. Data Wrangling(Feature Engineering & Transformation) (pandas, sklearn.preprocessing)**

Data Wrangling is the process of transforming raw data into a clean and usable format for machine learning models. This includes:

* Handling missing values
* Converting categorical (string) data to numerical format
* Scaling and normalizing numerical data

**Step4. Data Modelling(Splitting the dataset) (sklearn.model\_selection)**

* Splits data into training and testing sets.
* Library used: train\_test\_split from sklearn.model\_selection.

**Step5. Model Training** **(sklearn.linear\_model)**

* Trains a Machine Learning model on the dataset.
* Library used: sklearn.linear\_model for Linear Regression.

**Step6. Model Testing (Evaluation) (sklearn.metrics)**

* Evaluates the model using metrics like MSE, accuracy, R² score, etc.
* Library used: sklearn.metrics.

**Step7. Deployment** **(Data Visualization & Deployment)**

* Helps understand data distribution, correlations, and predictions.
* Libraries used: matplotlib, seaborn.

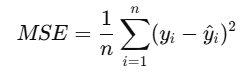
**Linear Regression model:**

**1. Mean Squared Error (MSE)**

mse = mean\_squared\_error(y\_test, y\_predict)

**What is it?**

* MSE calculates the **average of the squared differences** between actual (y\_test) and predicted (y\_predict) values.
* Formula:



Where:

* yi = Actual value
* y^i ​ = Predicted value
* n = Number of observations

✅ **Why it matters?**

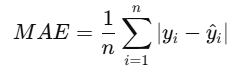
* It penalizes larger errors more than smaller ones because the errors are squared.
* Lower MSE indicates better performance.

2. **Mean Absolute Error (MAE)**

mbe = mean\_absolute\_error(y\_test, y\_predict)

✅ **What is it?**

* MAE calculates the **average of the absolute differences** between actual and predicted values.
* Formula:



✅ **Why it matters?**

* It gives an idea of how much the model's predictions deviate from actual values **on average**.
* It treats all errors equally (unlike MSE which squares them).

💡 **Key Difference:**

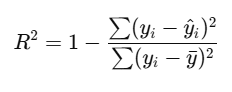
* MSE penalizes large errors, but MAE gives a more balanced estimate.

3. **R² Score (R-Squared)**

r2 = r2\_score(y\_test, y\_predict)

✅ **What is it?**

* R² measures the proportion of variance in the target variable that the model can explain.
* Formula:



Where:

* yˉ = Mean of actual values
* ∑(yi−y^i) = Sum of squared residuals
* ∑(yi−yˉ) = Total variance in the target variable

✅ **Why it matters?**

* R² ranges from **0 to 1**:
  + **1** → Perfect model (explains 100% variance).
  + **0** → Model explains no variance.
  + **Negative** → Worse than a horizontal line (bad model).