Unit-3: Understanding Machine Learning

Contents

3.1 Types of Machine Learning

Machine Learning (ML) is the science of enabling computers to learn from data and improve their performance automatically. The learning is broadly categorized into **Supervised Learning, Unsupervised Learning, and Reinforcement Learning** (though reinforcement is not in your unit, it is often mentioned).

3.1.1 Supervised Learning

- **Definition**:
  Supervised learning uses **labeled data** to train a model. Each training example has both the input (features) and the correct output (label). The model learns the relationship so it can predict outcomes for unseen data.
- **How it Works**:
  1. Provide data with known answers (features + labels).
  2. Train the model on this data.
  3. Test the model on new data to check if it can generalize.
- **Examples**:
  - Predicting **house prices** using features like area, location, and number of rooms (Regression).
  - Classifying **emails as spam or not spam** (Classification).
  - Detecting **whether a tumor is malignant or benign**.
- **Advantages**:
  - Produces accurate predictions when sufficient labeled data is available.
  - Well-established algorithms and evaluation metrics.
- **Disadvantages**:
  - Requires large, well-labeled datasets (expensive to prepare).
  - Models may overfit if training data is too small or noisy.

3.1.2 Unsupervised Learning

- **Definition**:
  Unsupervised learning deals with **unlabeled data**, where the goal is to discover hidden patterns, structures, or relationships in the data.
- **How it Works**:
  1. Input dataset without labels.
  2. Algorithm groups similar items or reduces dimensions.
  3. Results are clusters, associations, or compressed data.
- **Applications**:
  - Customer segmentation in retail (grouping customers by shopping habits).
  - Market basket analysis (finding items often bought together).
  - Image compression (dimensionality reduction).
- **Advantages**:
  - Works even without labeled data.
  - Helps explore unknown data patterns.
- **Disadvantages**:
  - Results can be subjective and harder to evaluate.
  - No ground truth to compare accuracy.

3.1.3 Applications of Machine Learning

Machine Learning is widely applied across industries because it allows systems to **learn from data, detect patterns, and make decisions or predictions with minimal human intervention**.

1. Healthcare

- **Disease Prediction & Diagnosis** → ML models predict diseases like cancer, diabetes, or heart problems based on medical records, lab results, or imaging.
- **Medical Imaging** → Used in X-rays, MRIs, CT scans for detecting tumors, fractures, or anomalies.

- **Drug Discovery** → Helps in analyzing molecular structures and predicting which compounds may become effective drugs.
- **Personalized Medicine** → Recommends treatments based on individual patient data.

Example: IBM Watson uses ML for oncology research and personalized healthcare recommendations.

2. Finance & Banking

- **Fraud Detection** → ML algorithms detect unusual spending or transaction patterns to prevent fraud.
- **Credit Scoring & Risk Assessment** → Banks use ML to analyze customer financial history and predict loan repayment capability.
- **Algorithmic Trading** → Automated stock trading using ML-based predictions of market trends.
- **Customer Support** → Chatbots and virtual assistants powered by ML handle customer queries.

Example: PayPal uses ML to detect fraudulent transactions in real-time.

3. Retail & E-commerce

- **Recommendation Systems** → Suggests products to customers based on browsing and purchasing history (e.g., Amazon, Flipkart).
- **Inventory Management** → Predicts product demand to optimize stock.
- **Customer Sentiment Analysis** → ML analyzes reviews and feedback to understand customer satisfaction.
- **Dynamic Pricing** → Adjusts product prices automatically based on demand, supply, and competition.

Example: Netflix and Amazon use ML recommendation systems to boost sales and user engagement.

4. Transportation & Autonomous Systems

- **Self-driving Cars** → ML powers object detection, lane detection, traffic sign recognition, and decision-making.
- **Route Optimization** → Predicts traffic congestion and suggests faster routes.
- **Predictive Maintenance** → Identifies potential vehicle breakdowns before they occur.

Example: Tesla's Autopilot system uses ML for real-time driving decisions.

5. Manufacturing & Industry

- **Quality Control** → ML detects defects in production lines through image recognition.
- **Predictive Maintenance** → Predicts equipment failures to reduce downtime.
- **Supply Chain Optimization** → ML forecasts demand and improves logistics planning.

Example: General Electric uses ML for predictive maintenance of industrial machines.

6. Agriculture

- **Crop Monitoring** → ML with sensors and drones detects crop diseases and monitors growth.
- **Yield Prediction** → Predicts crop production based on weather and soil data.
- **Smart Irrigation** → Automates water supply based on soil moisture and weather predictions.

Example: John Deere uses ML in smart farming equipment.

7. Education

- **Personalized Learning** → Adaptive learning platforms customize lessons based on a student's performance.

- **Automated Grading** → ML systems grade assignments, quizzes, and exams.
- **Learning Analytics** → Predicts students at risk of dropping out or failing.

Example: Coursera and Duolingo use ML to personalize learning experiences.

8. Natural Language Processing (NLP) Applications

- **Virtual Assistants** → Siri, Alexa, Google Assistant use ML for speech recognition and response.
- **Language Translation** → Google Translate applies ML for real-time translation.
- **Chatbots** → Customer service bots that learn from conversations.

9. Cybersecurity

- **Intrusion Detection** → Detects unusual behavior in systems to prevent hacking.
- **Spam & Phishing Detection** → Filters malicious emails automatically.
- **Malware Detection** → Identifies harmful software before execution.

10. Entertainment & Media

- **Content Recommendation** → YouTube, Spotify, and Netflix suggest videos/music.
- **Image & Video Recognition** → ML helps in facial recognition and tagging photos.
- **Game Development** → AI opponents in games learn and adapt to player behavior.

3.2 Overview of Supervised Learning

3.2.1 Concepts of Supervised Learning

- Works with **labeled data**.
- The algorithm learns a **mapping function**:

  f(X)→Yf(X) \to Yf(X)→Y

  where X = inputs (features), Y = outputs (labels).
- **Goal**: Generalization → The model should work well on unseen data, not just the training set.

3.2.2 Classification vs. Regression

1. **Classification**:
   - Predicts **categorical outputs** (discrete values).
   - Example:
     - Whether an email is **spam or not spam**.
     - Diagnosing whether a patient is **sick or healthy**.
   - Common algorithms: Logistic Regression, Decision Trees, Naive Bayes, SVM, Random Forest.
2. **Regression**:
   - Predicts **continuous outputs** (real values).
   - Example:
     - Predicting **stock prices** or **temperature**.
   - Common algorithms: Linear Regression, Polynomial Regression, Support Vector Regression.

Key Difference:

Classification answers **"Which category?"**, while Regression answers **"How much?"**

Aspect	Classification	Regression
Output type	Discrete categories (Yes/No, Class A/B)	Continuous numeric values
Examples	Spam email detection, disease prediction	Predicting temperature, house price
Algorithms	Logistic Regression, Decision Tree, SVM	Linear Regression, Polynomial Regression
Evaluation	Accuracy, Precision, Recall, F1-score	**MSE** (Mean Squared Error.), **MAE** (Mean Absolute Error.), **R^2** (Coefficient of Determination)

3.3 Basic Terminologies

3.3.1 Dataset

- A dataset is a collection of data points used to train and evaluate ML models.
- Typically arranged in **rows (examples)** and **columns (features/labels)**.
- Example (house price dataset):

Size (sqft)	Location	Price ($)
1000	City A	120,000
1500	City B	200,000

3.3.2 Features and Labels

- **Features (X)**: Input variables used to make predictions.
  - Example: Size, Location, Number of rooms.
- **Label (Y)**: Output we want to predict.
  - Example: House price.

3.3.3 Training, Testing, and Validation Data

- **Training Data (70-80%)**: Used to teach the model patterns.
- **Validation Data (10-15%)**: Used to tune hyperparameters and avoid overfitting.
- **Testing Data (10-15%)**: Used to measure final performance.

Analogy:

- Training = **studying**.
- Validation = **practicing mock tests**.
- Testing = **final exam**.

3.3.4 Overfitting and Underfitting

1. Overfitting

- **Definition**: Overfitting happens when a machine learning model learns the **training data too well**, including its **noise, outliers, and random fluctuations**.
- As a result, it performs **very well on training data** but **poorly on unseen/test data**.
- The model becomes **too complex** (e.g., too many parameters, deep decision trees, or excessive polynomial terms).

◆ **Causes of Overfitting:**

- Using a very **complex model** for limited data (e.g., deep neural network on small dataset).
- **Too many features** without proper feature selection.
- **Too Much training data**.

- **Lack of regularization**.

◆ **Signs of Overfitting:**

- High accuracy on training set but low accuracy on validation/test set.
- Model predictions vary a lot (high variance).

◆ **Solutions to Overfitting:**

- Simplify the model (reduce parameters, prune decision trees).
- Use **regularization** (L1/Lasso, L2/Ridge).
- Use **cross-validation** to tune hyperparameters.
- Collect more training data.
- Apply **dropout** (in neural networks).
- Perform **feature selection/dimensionality reduction** (e.g., PCA).

◆ **2. Underfitting**

- **Definition**: Underfitting happens when the model is **too simple** to capture the underlying patterns in data.
- It performs **poorly on both training and test data**.
- The model cannot learn enough, so it has **high bias**.

◆ **Causes of Underfitting:**

- Using a **too simple model** (e.g., linear regression for highly non-linear data).
- **Insufficient training** (not enough epochs in neural networks).
- **Over-regularization** (penalizing the model too much).
- Ignoring important features.

◆ **Signs of Underfitting:**

- Low accuracy (or high error) on both training and test sets.
- Model fails to capture trends (underestimates relationships).

◆ **Solutions to Underfitting:**

- Increase model complexity (e.g., use polynomial regression instead of simple linear regression).
- Reduce regularization strength.

- Train for longer (more epochs).
- Add more meaningful features.

3.4 Loss Functions

Loss functions measure **the error between predictions and actual values**. They guide optimization during training.

3.4.1 Mean Squared Error (MSE)

3.4.1.1 Definition

- The average of squared differences between actual and predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

3.4.1.2 Computation and Properties

- **Steps**:
  1. Subtract prediction from actual.
  2. Square the difference.
  3. Average across all points.
- **Properties**:
  - Heavily penalizes large errors.
  - Smooth and differentiable (suitable for gradient descent).
  - Sensitive to outliers (one large error can dominate).

3.4.2 Mean Absolute Error (MAE)

3.4.2.1 Definition

- The average of absolute differences between actual and predicted values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

3.4.2.2 Computation and Properties

- **Steps**:
  1. Subtract prediction from actual.
  2. Take absolute value.
  3. Average across all points.
- **Properties**:
  - Treats all errors equally (no extra penalty).
  - Robust to outliers compared to MSE.
  - Not differentiable at 0 (can complicate optimization).

MSE vs MAE

Aspect	**MSE (Mean Squared Error)**	**MAE (Mean Absolute Error)**
Penalization	Larger errors penalized more (squared).	All errors treated equally.
Sensitivity	Very sensitive to outliers.	More robust to outliers.
Differentiable	Yes (smooth).	No (not smooth at zero).
Use cases	Regression tasks where big errors matter.	Data with noise or outliers.