Type of Machine Learning Based on Training

Batch (Offline) Machine Learning

- In batch learning, the model is trained using a single, static set of data.
- Once trained, the model is deployed and does not update automatically with new data; retraining happens periodically as required.
- Most traditional machine learning workflows use this approach for scenarios where data does not change frequently or instant updates are not critical (e.g., movie recommendation systems, static prediction tasks).

Advantages

- Efficient use of computational resources, as training is done offline.
- Suitable for environments where internet connectivity is limited or unavailable.

Disadvantages

- Not responsive to rapidly changing data; requires manual retraining to stay updated.
- Can miss out on important trends between retraining intervals, especially in dynamic industries (e.g., social media platforms after major events).

Online Machine Learning

- Online learning allows the model to update continuously or incrementally as new data arrives in real-time.
- Suitable for applications where constant updates and adaptation are necessary (such as fraud detection or personalized recommendation engines).

Advantages

- Keeps the model current and responsive to new patterns or sudden changes in data.
- Useful in environments with streaming or frequently updating data, like social networks or IoT sensors.

Challenges

- Requires more complex systems to update models safely and efficiently without losing performance.
- Needs robust infrastructure for frequent training and deployment.

Type of Machine Learning Based on Training Timing

Instance-Based Learning

- Instance-based algorithms memorize all training examples and predict on new data by comparing its similarity to stored instances, rather than learning general rules.
- This strategy is also called memory-based learning or lazy learning because it delays computation until prediction; k-Nearest Neighbours is a classic example.
- Predictions rely on measures like distance or similarity between the new input and stored examples; this approach requires storing the entire training dataset and can be slow for large datasets, with worst-case time complexity O(n)$O(n)$, where n is the number of training records.

Model-Based Learning

- Model-based algorithms learn from training data by extracting an underlying pattern and creating a mathematical model, such as a regression equation or decision boundary.
- Once trained, the model summarizes learned rules in a compact form—parameters—so predictions are faster and do not require access to all training instances.
- Examples include linear regression, logistic regression, and most neural networks; these models can generalize rules for unseen data and require less storage once training is complete.

Key Differences

Feature	**Instance-Based Learning**	**Model-Based Learning**
Storage Needs	Stores all training instances	Stores only model parameters
Prediction Speed	Slower, depends on dataset size	Faster, independent of size
Generalization Strategy	Uses similarity to examples	Uses learned rules/models
Example Algorithms	k-NN, some network methods	Linear/logistic regression, SVM

Major Challenges in Machine Learning

- Data Collection Difficulty: Acquiring sufficient, relevant data is often hard, especially in real-world scenarios where data may be scarce, messy, or costly to gather.
- Insufficient Data: Poor model quality or biased results may arise from too little data or data that doesn't represent the broader problem domain properly.
- Non-Representative Data: Training data that is not a true reflection of the problem's variety leads to errors in model predictions (sampling bias).
- Poor Quality Data: Incomplete, inconsistent, or corrupted data can heavily degrade model performance; data cleaning and preprocessing consume a large portion of ML project time.
- Irrelevant Features: Including features that do not contribute meaningful insight adds noise, affecting model accuracy; careful feature selection and engineering are needed.
- Overfitting: Models that memorize training data too closely perform poorly on new data, limiting generalization ability.
- Underfitting: Models that are too simple or not well-trained fail to capture data patterns, resulting in low accuracy on both training and test data.
- Software Integration: Deploying ML models into existing software systems and platforms is complex; compatibility and maintenance issues occur frequently.
- Offline Learning/Deployment Challenges: Static models require retraining to incorporate new data, creating latency in adaptation to recent patterns.
- Cost Involved: Developing, training, deploying, and maintaining ML systems can be resource-intensive, involving computational, human, and infrastructure costs.