Foundations of Machine Learning

What is Machine Learning ?

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. For example, A spam filter (classification).

Types of Machine Learning

1. Supervised / Unsupervised Learning

Supervised Learning (with labels)

The training data provided to the algorithm includes the desired solutions, called **labels**.

Typical tasks include:

- **Classification** — predicting a category or class.
- **Regression** — predicting a continuous value.

Common supervised learning algorithms include: *Logistic Regression, k-Nearest Neighbors (kNN), Linear Regression, Support Vector Machines (SVMs), Decision Trees, and Random Forests.*

Unsupervised Learning (without labels)

The training data is unlabeled, and the system tries to learn patterns without guidance.

Typical tasks include:

- **Clustering** — grouping similar data points based on features without predefined labels.
- **Visualization** — reducing complex, high-dimensional data into 2D or 3D for exploring patterns or clusters.
- **Dimensionality Reduction** — transforming data from a high-dimensional space (many features) into a lower-dimensional space (fewer features).
- **Association Rule Learning** — discovering relationships between variables in large datasets.

Semi-Supervised Learning (combination of supervised and unsupervised learning)

Some algorithms can handle partially labeled data — typically a small amount of labeled data and a large amount of unlabeled data.

Example: Deep Belief Networks (DBNs).

Reinforcement Learning

A learning paradigm where an **agent** interacts with an **environment** to achieve a goal. The agent learns from feedback in the form of **rewards** or **penalties**, rather than labeled data.

Example: A robot learns a winning strategy by playing millions of games against itself.

2. Batch and Online Learning

Batch Learning (Offline Learning)

In **batch learning**, the model is trained on the entire dataset at once. After training, the model is fixed — it does not update itself automatically when new data arrives. If new data comes in, you typically retrain the model from scratch (or partially) with the updated dataset.

Online Learning

In **online learning**, the model learns continuously — updating its parameters as new data arrives. An important parameter of online learning systems is how fast they should adapt to changing data: this is called the **learning rate**.

3. Instance-Based Versus Model-Based Learning

Instance-Based Learning (also called memory-based or lazy learning)

The algorithm stores the training data and defers learning until prediction time. When a new input arrives, it compares it to stored examples and makes a prediction based on similarity. *Example: k-Nearest Neighbors (kNN).*

Model-Based Learning

The algorithm learns a **parametric model** from training data that summarizes the relationship between inputs and outputs. It aims to find a set of parameters θ such that the model $f(x;\theta)$ best predicts the target y, typically by minimizing a loss function or maximizing likelihood (e.g., Maximum Likelihood Estimation).

Main Challenges of Machine Learning

1. Insufficient Quantity of Training Data

- Machine Learning requires large amounts of data.
- For good performance—especially in complex tasks such as image or speech recognition—we often need thousands to millions of examples.

2. Nonrepresentative Training Data

- A large dataset is not enough; it must be representative of the problem you want to solve.

3. Poor-Quality Data

- Clean, accurate, and consistent data is essential—poor-quality data leads to poor models, regardless of the algorithm's sophistication.

4. Irrelevant Features

- A model can only learn well if the training data includes useful, relevant features and avoids irrelevant or noisy ones.

5. Overfitting the Training Data

- Overfitting occurs when a model learns the training data too well, capturing not only real patterns but also noise or random fluctuations.
- The model is too complex for the amount or quality of training data.
- The training data may be too small or noisy, leading the model to memorize instead of generalize.
- **How to Reduce Overfitting:**
  - Collect more training data.
  - Apply regularization techniques.
  - Remove noise and outliers.

6. Underfitting the Training Data

- Underfitting occurs when a model is too simple to capture real patterns in the data.
- It performs poorly even on the training set, indicating insufficient learning.
- **How to Fix Underfitting:**
  - Use a more powerful or complex model.
  - Provide better or more relevant features.

Testing and Validating

- **Goal:** To evaluate how well a model generalizes to new, unseen data.
- **Approach:** Split your dataset into:
  - **Training set** — used to train the model.
  - **Test set** — used to evaluate performance on unseen data.
- **Generalization error (out-of-sample error):** The error rate on new cases, showing how well the model generalizes.
- **Overfitting:** Occurs when training error is low but generalization error is high — meaning the model memorizes training data instead of learning true patterns.
- **Common split:** 80% for training, 20% for testing.

The Validation Set and Overfitting the Test Set

- Repeatedly tuning your model using the test set causes indirect training on it, leading to **overfitting to the test set**.
- **Solution:** Introduce a **validation set**:
  - Train using the training set.
  - Tune hyperparameters on the validation set.
  - Test once on the test set for the final generalization estimate.
- **Cross-validation:** A technique that splits the training data into multiple subsets (folds). Each model is trained and validated on different combinations of these folds, improving the reliability of performance estimates.

No Free Lunch Theorem

- There is no universally best model — performance depends on the specific problem and dataset.
- Every model makes assumptions about the data (e.g., linear models assume linear relationships).
- You must evaluate multiple models to determine which performs best for your case.

Exercises

1. How would you define Machine Learning?

   Machine Learning is about building systems that can learn from data. Learning means getting better at some task, given some performance measure.

2. Can you name four types of problems where it shines?

   Machine Learning is great for complex problems for which we have no algorithmic solution, to replace long lists of hand-tuned rules, to build systems that adapt to fluctuating environments, and finally to help humans learn (e.g., data mining).

3. What is a labeled training set?

   A labeled training set is a training set that contains the desired solution (a.k.a. a label) for each instance.

4. What are the two most common supervised tasks?

   The two most common supervised tasks are regression and classification.

5. Can you name four common unsupervised tasks?

   Common unsupervised tasks include clustering, visualization, dimensionality reduction, and association rule learning.

6. What type of Machine Learning algorithm would you use to allow a robot to walk in various unknown terrains?

   Reinforcement Learning is likely to perform best if we want a robot to learn to walk in various unknown terrains since this is typically the type of problem that Reinforcement Learning tackles. It might be possible to express the problem as a supervised or semisupervised learning problem, but it would be less natural.

7. What type of algorithm would you use to segment your customers into multiple groups?

   If you don't know how to define the groups, then you can use a clustering algorithm (unsupervised learning) to segment your customers into clusters of similar customers. However, if you know what groups you would like to have, then you can feed many examples of each group to a classification algorithm (supervised learning), and it will classify all your customers into these groups.

8. Would you frame the problem of spam detection as a supervised learning problem or an unsupervised learning problem?

   Spam detection is a typical supervised learning problem: the algorithm is fed many emails along with their label (spam or not spam).

9. What is an online learning system?

   An online learning system can learn incrementally, as opposed to a batch learning system. This makes it capable of adapting rapidly to both changing data and autonomous systems, and of training on very large quantities of data

10. What is out-of-core learning?

    Out-of-core algorithms can handle vast quantities of data that cannot fit in a computer's main memory. An out-of-core learning algorithm chops the data into mini-batches and uses online learning techniques to learn from these mini-batches.

11. What type of learning algorithm relies on a similarity measure to make predictions?

    An instance-based learning system learns the training data by heart; then, when given a new instance, it uses a similarity measure to find the most similar learned instances and uses them to make predictions.

12. What is the difference between a model parameter and a learning algorithm's hyperparameter?

    A model has one or more model parameters that determine what it will predict given a new instance (e.g., the slope of a linear model). A learning algorithm tries to find optimal values for these parameters such that the model generalizes well to new instances. A hyperparameter is a parameter of the learning algorithm itself, not of the model (e.g., the amount of regularization to apply).

13. What do model-based learning algorithms search for? What is the most common strategy they use to succeed? How do they make predictions?

    Model-based learning algorithms search for an optimal value for the model parameters such that the model will generalize well to new instances. We usually train such systems by minimizing a cost function that measures how bad the system is at making predictions on the training data, plus a penalty for model complexity if the model is regularized. To make predictions, we feed the new instance's features into the model's prediction function, using the parameter values found by the learning algorithm.

14. Can you name four of the main challenges in Machine Learning?

    Some of the main challenges in Machine Learning are the lack of data, poor data quality, non-representative data, uninformative features, excessively simple models that underfit the training data, and excessively complex models that overfit the data.

15. If your model performs great on the training data but generalizes poorly to new instances, what is happening? Can you name three possible solutions?

    If a model performs great on the training data but generalizes poorly to new instances, the model

is likely overfitting the training data (or we got extremely lucky on the training data). Possible solutions to overfitting are getting more data, simplifying the model (selecting a simpler algorithm, reducing the number of parameters or features used, or regularizing the model), or reducing the noise in the training data.

16. What is a test set and why would you want to use it?

    A test set is used to estimate the generalization error that a model will make on new instances, before the model is launched in production.

17. What is the purpose of a validation set?

    A validation set is used to compare models. It makes it possible to select the best model and tune the hyperparameters.

18. What can go wrong if you tune hyperparameters using the test set?

    If you tune hyperparameters using the test set, you risk overfitting the test set, and the generalization error you measure will be optimistic (you may launch a model that performs worse than you expect).

19. What is cross-validation and why would you prefer it to a validation set?

    Cross-validation is a technique that makes it possible to compare models (for model selection and hyperparameter tuning) without the need for a separate validation set. This saves precious training data.