# The Machine Learning Landscape

Gauranga Kumar Baishya

August 27, 2025

# What Is Machine Learning?

### Core Idea

Machine Learning is the science (and art) of programming computers so they can **learn from data**.

# What Is Machine Learning?

## Core Idea

Machine Learning is the science (and art) of programming computers so they can **learn from data**.

## Key Definitions

- **Arthur Samuel, 1959:** The field of study that gives computers the ability to learn without being explicitly programmed.

# What Is Machine Learning?

## Core Idea

Machine Learning is the science (and art) of programming computers so they can **learn from data**.

## Key Definitions

- **Arthur Samuel, 1959:** The field of study that gives computers the ability to learn without being explicitly programmed.
- **Tom Mitchell, 1997:** A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

# What Is Machine Learning?

## Core Idea

Machine Learning is the science (and art) of programming computers so they can **learn from data**.

## Key Definitions

- **Arthur Samuel, 1959:** The field of study that gives computers the ability to learn without being explicitly programmed.
- **Tom Mitchell, 1997:** A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

  For example, Task (T): Flag spam for new emails, Experience (E): Training data of example spam and non-spam emails & Performance (P): Accuracy of spam detection.

# Why Use Machine Learning?

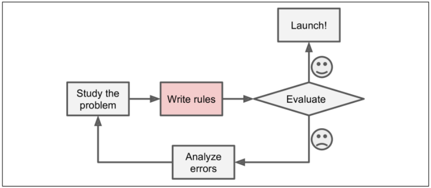



*Figure 1-1. The traditional approach*

# Why Use Machine Learning?

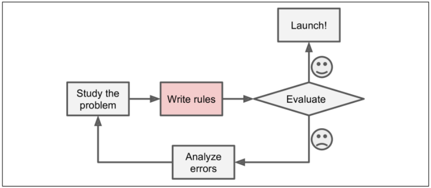



*Figure 1-1. The traditional approach*

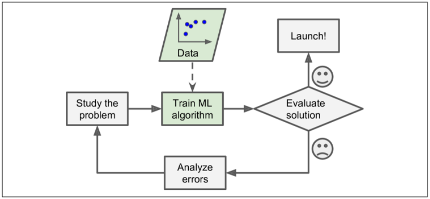



*Figure 1-2. Machine Learning approach*

# Why Use Machine Learning?

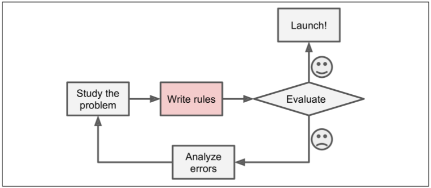



*Figure 1-1. The traditional approach*

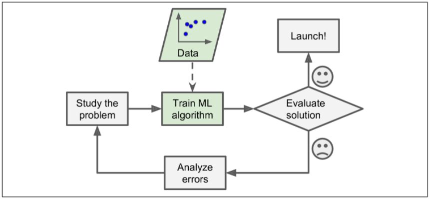



*Figure 1-2. Machine Learning approach*

## ML is Great For:

- Problems requiring lots of hand-tuning or long lists of rules.
- Complex problems with no traditional solution.
- Fluctuating environments (ML systems can adapt).
- Getting insights from large amounts of data (Data Mining).

# Types of Machine Learning Systems

ML systems can be classified based on:

1. **Human Supervision:** Whether or not they are trained with human supervision.
    - Supervised
    - Unsupervised
    - Semisupervised
    - Reinforcement Learning

# Types of Machine Learning Systems

ML systems can be classified based on:

1. **Human Supervision:** Whether or not they are trained with human supervision.
   - Supervised
   - Unsupervised
   - Semisupervised
   - Reinforcement Learning
2. **Learning on the Fly:** Whether or not they can learn incrementally on the fly.
   - Batch Learning
   - Online Learning

# Types of Machine Learning Systems

ML systems can be classified based on:

1. **Human Supervision:** Whether or not they are trained with human supervision.
   - Supervised
   - Unsupervised
   - Semisupervised
   - Reinforcement Learning
2. **Learning on the Fly:** Whether or not they can learn incrementally on the fly.
   - Batch Learning
   - Online Learning
3. **Generalization Strategy:** Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do.
   - Instance-Based Learning
   - Model-Based Learning

# 1. Human Supervision: Supervised Learning

## Core Concept

The training data fed to the algorithm includes the desired solutions, called **labels**. The system learns from a "teacher."

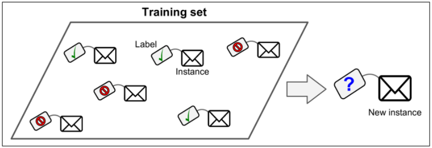



*Figure 1-5. A labeled training set for supervised learning (e.g., spam classification)*

# 1. Human Supervision: Supervised Learning

**Core Concept**

The training data fed to the algorithm includes the desired solutions, called **labels**. The system learns from a "teacher."

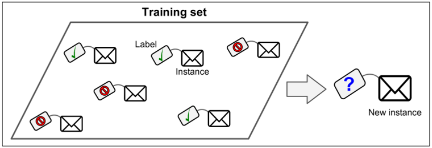



*Figure 1-5. A labeled training set for supervised learning (e.g., spam classification)*


*Figure 1-6. Regression*

# 1. Human Supervision: Supervised Learning

**Core Concept**

The training data fed to the algorithm includes the desired solutions, called **labels**. The system learns from a "teacher."

**Classification**

- Predicts a class or category.
- **Example:** Spam or not spam?

**Regression**

- Predicts a target numeric value.
- **Example:** What is the price of this car given its mileage and age?

# 1. Human Supervision: Supervised Learning

### Core Concept

The training data fed to the algorithm includes the desired solutions, called **labels**. The system learns from a "teacher."

**Classification**

- Predicts a class or category.
- **Example:** Spam or not spam?

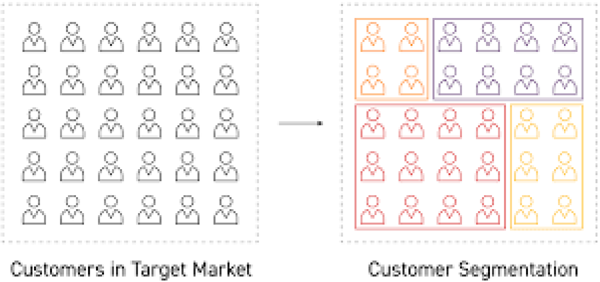
**Regression**

- Predicts a target numeric value.
- **Example:** What is the price of this car given its mileage and age?

### Common Algorithms

k-Nearest Neighbors, Linear Regression, Support Vector Machines (SVMs), Decision Trees & Random Forests, Neural Networks.

# 1. Human Supervision: Unsupervised Learning



# 1. Human Supervision: Unsupervised Learning

## Core Concept

The training data is **unlabeled**. The system tries to learn without a teacher, finding hidden patterns on its own.

- **Clustering:** Detect groups of similar instances.
  - *Example:* Segmenting blog visitors into groups with similar interests.

# 1. Human Supervision: Unsupervised Learning

**Core Concept**

The training data is **unlabeled**. The system tries to learn without a teacher, finding hidden patterns on its own.

- **Clustering:** Detect groups of similar instances.
  - *Example:* Segmenting blog visitors into groups with similar interests.
- **Anomaly Detection:** Identify unusual instances.
  - *Example:* Detecting credit card fraud.

# 1. Human Supervision: Unsupervised Learning

**Core Concept**

The training data is **unlabeled**. The system tries to learn without a teacher, finding hidden patterns on its own.

- **Clustering:** Detect groups of similar instances.
  - *Example:* Segmenting blog visitors into groups with similar interests.
- **Anomaly Detection:** Identify unusual instances.
  - *Example:* Detecting credit card fraud.
- **Dimensionality Reduction:** Simplify data by merging correlated features.
  - *Example:* Combining a car's age and mileage into a "wear and tear" feature.

# 1. Human Supervision: Unsupervised Learning

## Core Concept

The training data is **unlabeled**. The system tries to learn without a teacher, finding hidden patterns on its own.

- **Clustering:** Detect groups of similar instances.
  - *Example:* Segmenting blog visitors into groups with similar interests.
- **Anomaly Detection:** Identify unusual instances.
  - *Example:* Detecting credit card fraud.
- **Dimensionality Reduction:** Simplify data by merging correlated features.
  - *Example:* Combining a car's age and mileage into a "wear and tear" feature.
- **Association Rule Learning:** Discover interesting relations between attributes.
  - *Example:* Discovering that customers who buy barbecue sauce also tend to buy steak.

# 1. Human Supervision: Other Methods

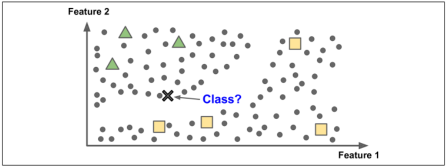


*Figure 1-11. Semisupervised learning*

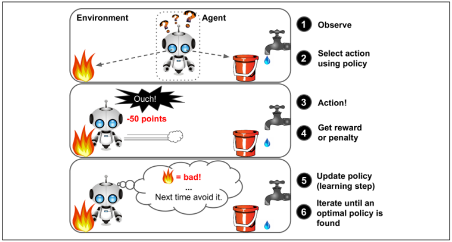


*Figure 1-12. Reinforcement Learning*

# 1. Human Supervision: Other Methods

## Semisupervised Learning

Deals with partially labeled data: mostly unlabeled data with a small amount of labeled data.

- **Example:** Google Photos clusters faces automatically (unsupervised) and then asks you to label just one photo per person (supervised).

# 1. Human Supervision: Other Methods

## Semisupervised Learning

Deals with partially labeled data: mostly unlabeled data with a small amount of labeled data.

- **Example:** Google Photos clusters faces automatically (unsupervised) and then asks you to label just one photo per person (supervised).

## Reinforcement Learning

An **agent** learns by performing actions and getting rewards or penalties. It learns the best strategy (**policy**) to maximize its reward over time.

- **Example:** A robot learning to walk, or DeepMind's AlphaGo learning to play Go.

# 2. Learning Method: Batch vs. Online

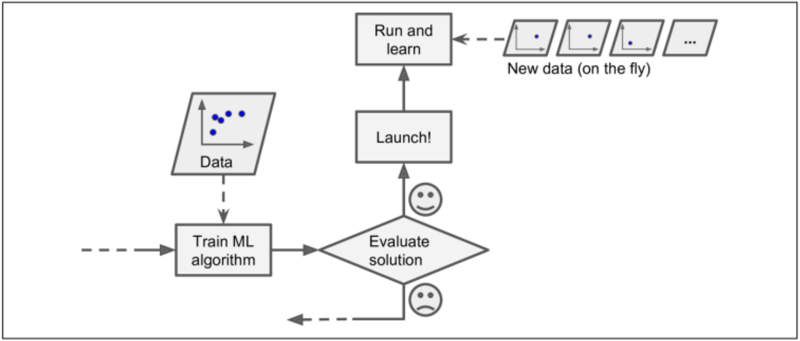



*Figure 1-13. Online learning*

# 2. Learning Method: Batch vs. Online

**Batch Learning (Offline)**

- Must be trained using all available data at once.
- Cannot learn incrementally.
- To learn new data, you must retrain a new version from scratch.
- Time and resource-intensive.

# 2. Learning Method: Batch vs. Online

**Batch Learning (Offline)**

- Must be trained using all available data at once.
- Cannot learn incrementally.
- To learn new data, you must retrain a new version from scratch.
- Time and resource-intensive.

**Online Learning (Incremental)**

- Trains the system by feeding it data instances sequentially (or in mini-batches).
- Learns about new data on the fly.
- Great for systems with continuous data flow (e.g., stock prices).
- Can handle huge datasets that don't fit in memory (*out-of-core learning*).

# 2. Learning Method: Batch vs. Online

### Batch Learning (Offline)

- Must be trained using all available data at once.
- Cannot learn incrementally.
- To learn new data, you must retrain a new version from scratch.
- Time and resource-intensive.

### Online Learning (Incremental)

- Trains the system by feeding it data instances sequentially (or in mini-batches).
- Learns about new data on the fly.
- Great for systems with continuous data flow (e.g., stock prices).
- Can handle huge datasets that don't fit in memory (*out-of-core learning*).

**Challenge with Online Learning**

Performance can decline if bad data is fed to the system.

# 3. Generalization: Instance vs. Model-Based

The true goal of ML is to perform well on **new instances** it has never seen before. This is called **generalization**.

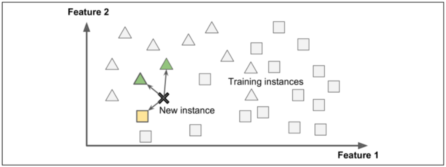



*Figure 1-15. Instance-based learning*

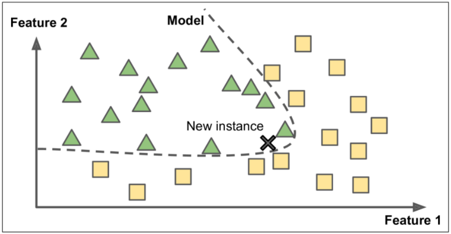



*Figure 1-16. Model-based learning*

# 3. Generalization: Instance vs. Model-Based

The true goal of ML is to perform well on **new instances** it has never seen before. This is called **generalization**.

**Instance-Based Learning**

- The system learns the examples by heart.
- Generalizes to new cases using a **similarity measure**.
- **Example:** k-Nearest Neighbors. To predict a new instance, it looks at the 'k' most similar instances in the training data.

# 3. Generalization: Instance vs. Model-Based

The true goal of ML is to perform well on **new instances** it has never seen before. This is called **generalization**.

**Instance-Based Learning**

- The system learns the examples by heart.
- Generalizes to new cases using a **similarity measure**.
- **Example:** k-Nearest Neighbors. To predict a new instance, it looks at the 'k' most similar instances in the training data.

**Model-Based Learning**

- Builds a **model** from the training examples.
- Uses that model to make predictions.
- **Example:** Linear Regression. The algorithm finds the best-fit line (the model) to the data.

# Main Challenges in ML: Bad Data

Your main task is to select an algorithm and train it on data. The two things that can go wrong are "bad algorithm" and "bad data."

## Examples of Bad Data

- **Insufficient Quantity of Training Data:** Most ML algorithms need thousands of examples to work properly.

# Main Challenges in ML: Bad Data

Your main task is to select an algorithm and train it on data. The two things that can go wrong are “bad algorithm” and “bad data.”

## Examples of Bad Data

- **Insufficient Quantity of Training Data:** Most ML algorithms need thousands of examples to work properly.
- **Nonrepresentative Training Data:** If the training data doesn’t represent the cases you want to generalize to, the model will be biased. This is called *sampling bias*.

# Main Challenges in ML: Bad Data

Your main task is to select an algorithm and train it on data. The two things that can go wrong are "bad algorithm" and "bad data."

## Examples of Bad Data

- **Insufficient Quantity of Training Data:** Most ML algorithms need thousands of examples to work properly.
- **Nonrepresentative Training Data:** If the training data doesn't represent the cases you want to generalize to, the model will be biased. This is called *sampling bias*.
- **Poor-Quality Data:** Errors, outliers, and noise make it harder for the system to detect underlying patterns.

# Main Challenges in ML: Bad Data

Your main task is to select an algorithm and train it on data. The two things that can go wrong are "bad algorithm" and "bad data."

## Examples of Bad Data

- **Insufficient Quantity of Training Data:** Most ML algorithms need thousands of examples to work properly.
- **Nonrepresentative Training Data:** If the training data doesn't represent the cases you want to generalize to, the model will be biased. This is called *sampling bias*.
- **Poor-Quality Data:** Errors, outliers, and noise make it harder for the system to detect underlying patterns.
- **Irrelevant Features:** "Garbage in, garbage out." The success of an ML project depends on **feature engineering** (selecting, extracting, and creating good features).

# Main Challenges in ML: Bad Algorithm

## Overfitting the Training Data

The model performs well on the training data, but it does not generalize well to new instances.

- **Cause:** The model is too complex relative to the amount and noisiness of the data. It detects patterns in the noise itself.
- **Solutions:**
  - Simplify the model (fewer parameters).
  - Gather more training data.
  - Reduce noise in the data.
  - Constrain the model (**Regularization**).

# Main Challenges in ML: Bad Algorithm

## Underfitting the Training Data

The opposite of overfitting. The model is too simple to learn the underlying structure of the data.

- **Solutions:**
  - Select a more powerful model (more parameters).
  - Feed better features to the algorithm (feature engineering).
  - Reduce the constraints on the model (e.g., reduce regularization).

# Testing and Validating

How do you know how well a model will generalize to new cases?

## Splitting Your Data

Split your data into two (or three) sets:

- **Training Set:** Used to train the model.
- **Test Set:** Used to estimate the **generalization error** (error on new cases). You test the model on this set only at the very end.

# Testing and Validating

How do you know how well a model will generalize to new cases?

## Splitting Your Data

Split your data into two (or three) sets:

- **Training Set:** Used to train the model.
- **Test Set:** Used to estimate the **generalization error** (error on new cases). You test the model on this set only at the very end.

## The Hyperparameter Tuning Problem

If you measure the generalization error multiple times on the test set to find the best model hyperparameters, your model will be tuned for that specific test set and may not perform well on new data.

# Testing and Validating

How do you know how well a model will generalize to new cases?

## Solution: Validation Set

- Hold out a third set, the **validation set**.
- Train multiple models on the training set.
- Select the best model by comparing performance on the validation set.
- After you have your final model, you perform a single final test on the test set.
- **Cross-validation** is a common technique to avoid wasting too much training data in a single validation set. For example: 10-fold cross validation.

# Summary: The Big Picture

- Machine Learning is about enabling machines to learn from data.

## Summary: The Big Picture

- Machine Learning is about enabling machines to learn from data.
- Systems can be supervised/unsupervised, batch/online, and instance/model-based.

# Summary: The Big Picture

- Machine Learning is about enabling machines to learn from data.
- Systems can be supervised/unsupervised, batch/online, and instance/model-based.
- A typical project involves gathering data, feeding it to a learning algorithm, and tuning a model to make predictions.

# Summary: The Big Picture

- Machine Learning is about enabling machines to learn from data.
- Systems can be supervised/unsupervised, batch/online, and instance/model-based.
- A typical project involves gathering data, feeding it to a learning algorithm, and tuning a model to make predictions.
- Success depends on having enough high-quality, representative data and a good set of features.

# Summary: The Big Picture

- Machine Learning is about enabling machines to learn from data.
- Systems can be supervised/unsupervised, batch/online, and instance/model-based.
- A typical project involves gathering data, feeding it to a learning algorithm, and tuning a model to make predictions.
- Success depends on having enough high-quality, representative data and a good set of features.
- The model must find a balance between being too simple (**underfitting**) and too complex (**overfitting**).

# Summary: The Big Picture

- Machine Learning is about enabling machines to learn from data.
- Systems can be supervised/unsupervised, batch/online, and instance/model-based.
- A typical project involves gathering data, feeding it to a learning algorithm, and tuning a model to make predictions.
- Success depends on having enough high-quality, representative data and a good set of features.
- The model must find a balance between being too simple (**underfitting**) and too complex (**overfitting**).
- Always evaluate your model's ability to generalize using a holdout test set.

# Questions?

Thank You!