Chapter 1

Machine Learning

> field of study that gives computers the ability to learn without being explicitly programmed.

Ex

- Spam Filters
- Fraud Detection
- Music Recommendation
- Voice Recogination

Labeld Training Set: Is a dataset that contains a target variable (Dependant Variable)

Supervised Tasks

1. Regression
2. Classification

Unsupervised Tasks

1. **Clustering:** Grouping similar data points together.
2. **Dimensionality Reduction:** Reducing the number of features in a dataset while preserving its essential information.
3. **Anomaly Detection:** Identifying unusual data points that differ significantly from the majority.
4. **Association Rule Learning:** Discovering interesting relationships between variables in large datasets.
5. **Self-Organizing Maps (SOM):** A type of neural network used for visualization and clustering.
6. **Density Estimation:** Estimating the probability distribution of a dataset.
7. **Matrix Factorization:** Decomposing a matrix into factors to identify latent structures.

Difference Between Supervised and Unsupervised Learning

Aspect	Supervised Learning	Unsupervised Learning
Definition	Learning with labeled data	Learning with unlabeled data
Objective	Predict outcomes or classify data based on labels	Discover hidden patterns or structures in data
Input Data	Labeled data (input-output pairs)	Unlabeled data (input only)
Algorithms	- Linear Regression	- K-Means Clustering
	- Logistic Regression	- Hierarchical Clustering
	- Decision Trees	- DBSCAN
	- Random Forests	- Principal Component Analysis (PCA)
	- Support Vector Machines (SVM)	- t-Distributed Stochastic Neighbor Embedding (t-SNE)
	- Neural Networks (NN)	- Gaussian Mixture Models (GMM)
	- K-Nearest Neighbors (K-NN)	- Self-Organizing Maps (SOM)
	- Naive Bayes	- Apriori Algorithm (for Association Rule Learning)
Output	Predictive model or classifier	Cluster assignments, reduced dimensions, or patterns
Evaluation Metrics	- Accuracy	- Silhouette Score
	- Precision	- Davies-Bouldin Index
	- Recall	- Inertia (for K-Means)
	- F1 Score	- Explained Variance (for PCA)
	- Mean Absolute Error (MAE)	- Anomaly Detection Scores
	- Mean Squared Error (MSE)	
Training Process	Trains with labeled data to minimize error	Learns patterns or clusters without labels
Use Cases	- Spam detection	- Customer segmentation

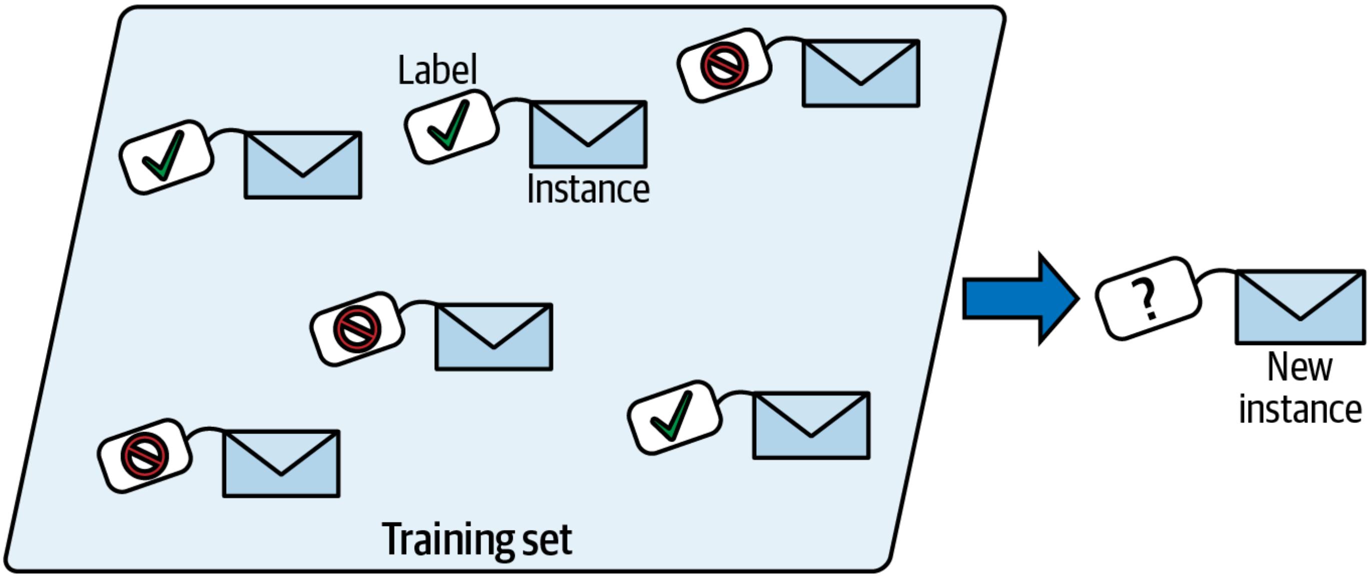
Aspect	Supervised Learning	Unsupervised Learning
	- Sentiment analysis	- Anomaly detection
	- Image classification	- Market basket analysis
	- Predictive maintenance	- Data visualization
	- Fraud detection	- Dimensionality reduction
	- Medical diagnosis	- Topic modeling in NLP
Complexity	Often simpler to understand and implement	Can be more complex to interpret and evaluate
Scalability	Can be computationally intensive for large datasets	Often scalable, but depends on the algorithm
Dependency	Highly dependent on the quality and quantity of labeled data	Less dependent on labeled data, focuses on the inherent structure

Fine-tuning: is a process in machine learning and deep learning where a pre-trained model is further trained on a new dataset to improve its performance on a specific task.

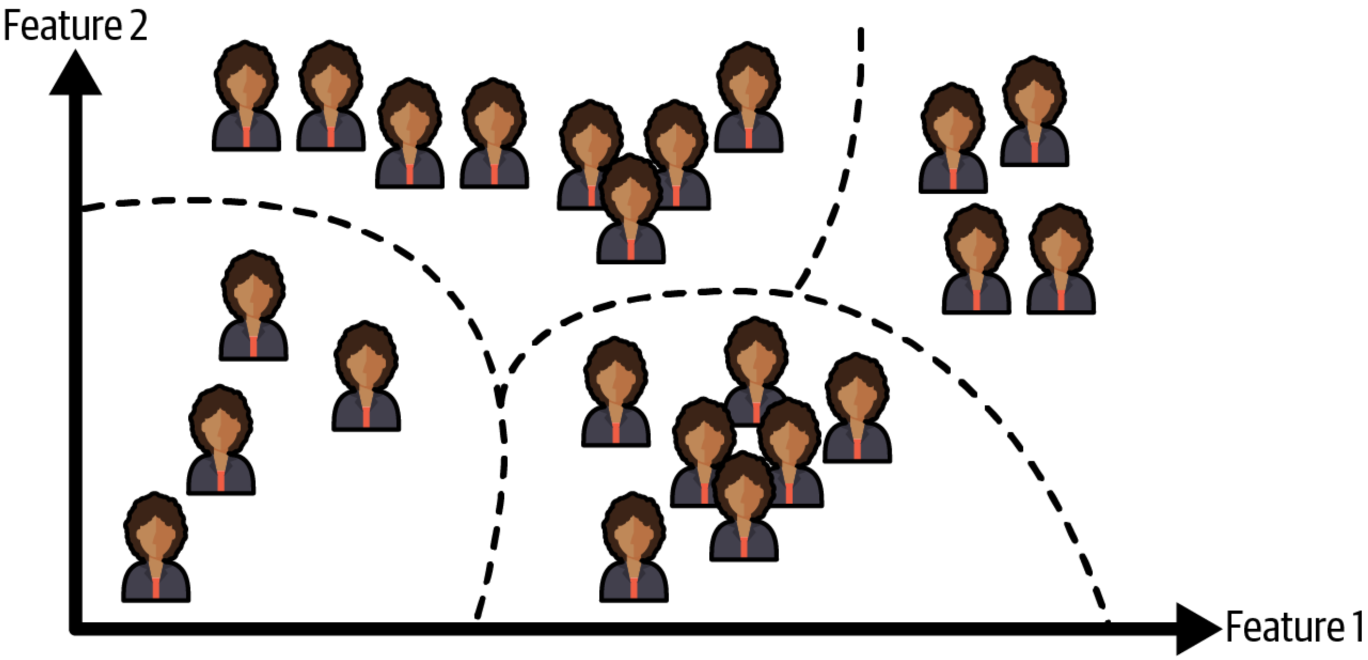
> This approach leverages the knowledge and patterns learned by the model during its initial training on a large, general dataset, and then adapts this knowledge to the new, often smaller and more specific, dataset.

Training Supervision

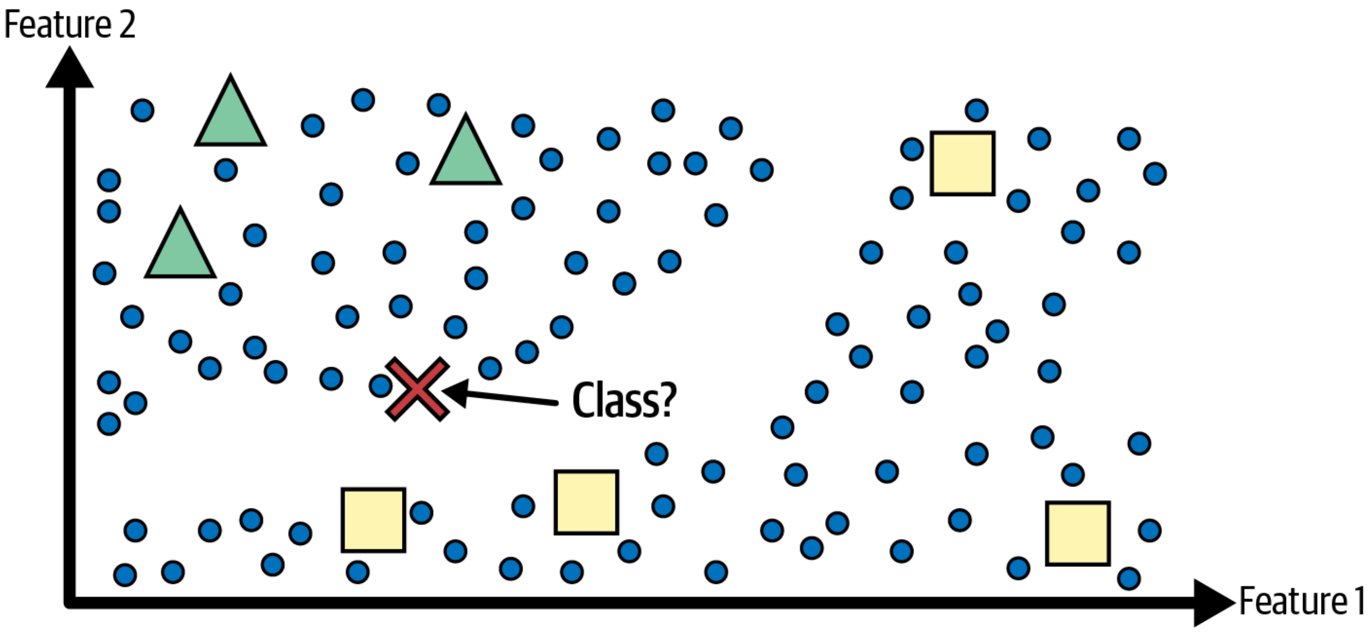
Supervised Learning: In supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels



Unsupervised learning:the training data is unlabeled



Semi-supervised learning: Some algorithms can deal with data that’s partially labeled.



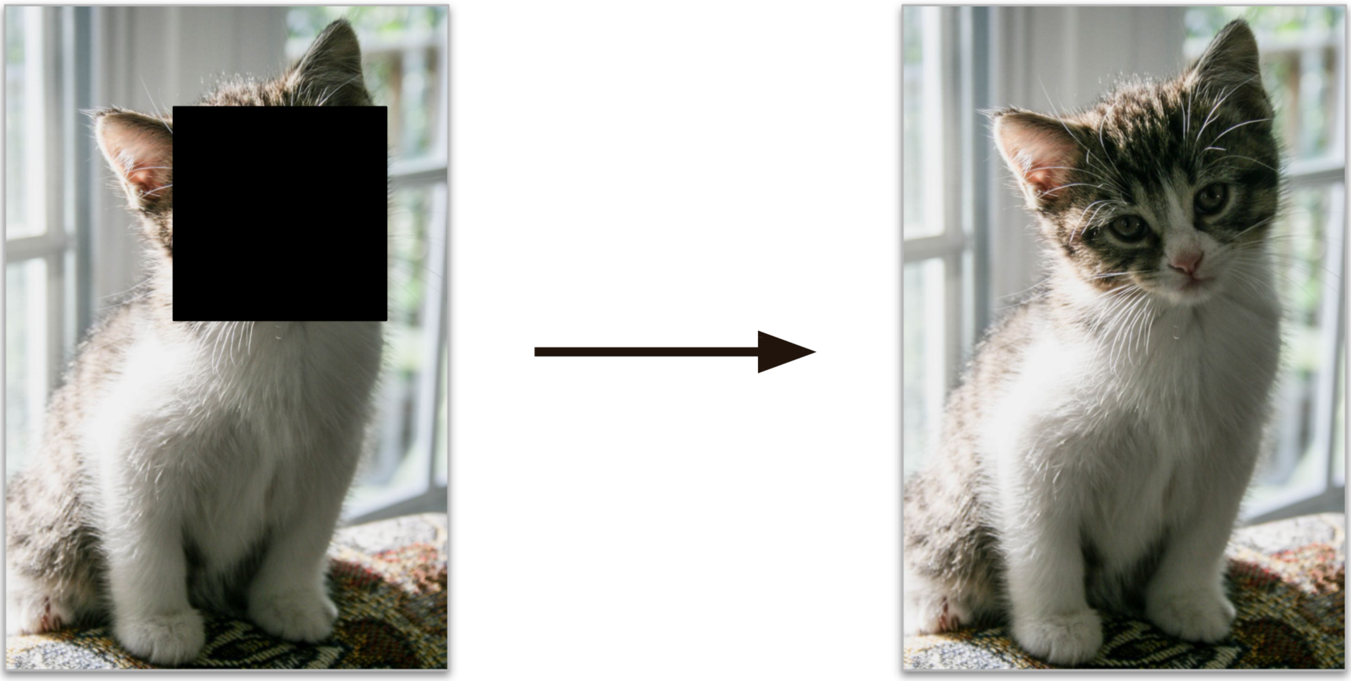
Semi-supervised learning with two classes (triangles and squares): the unlabeled examples (circles) help classify a new instance (the cross) into the triangle class rather than the square class, even though it is closer to the labeled squares

Example:

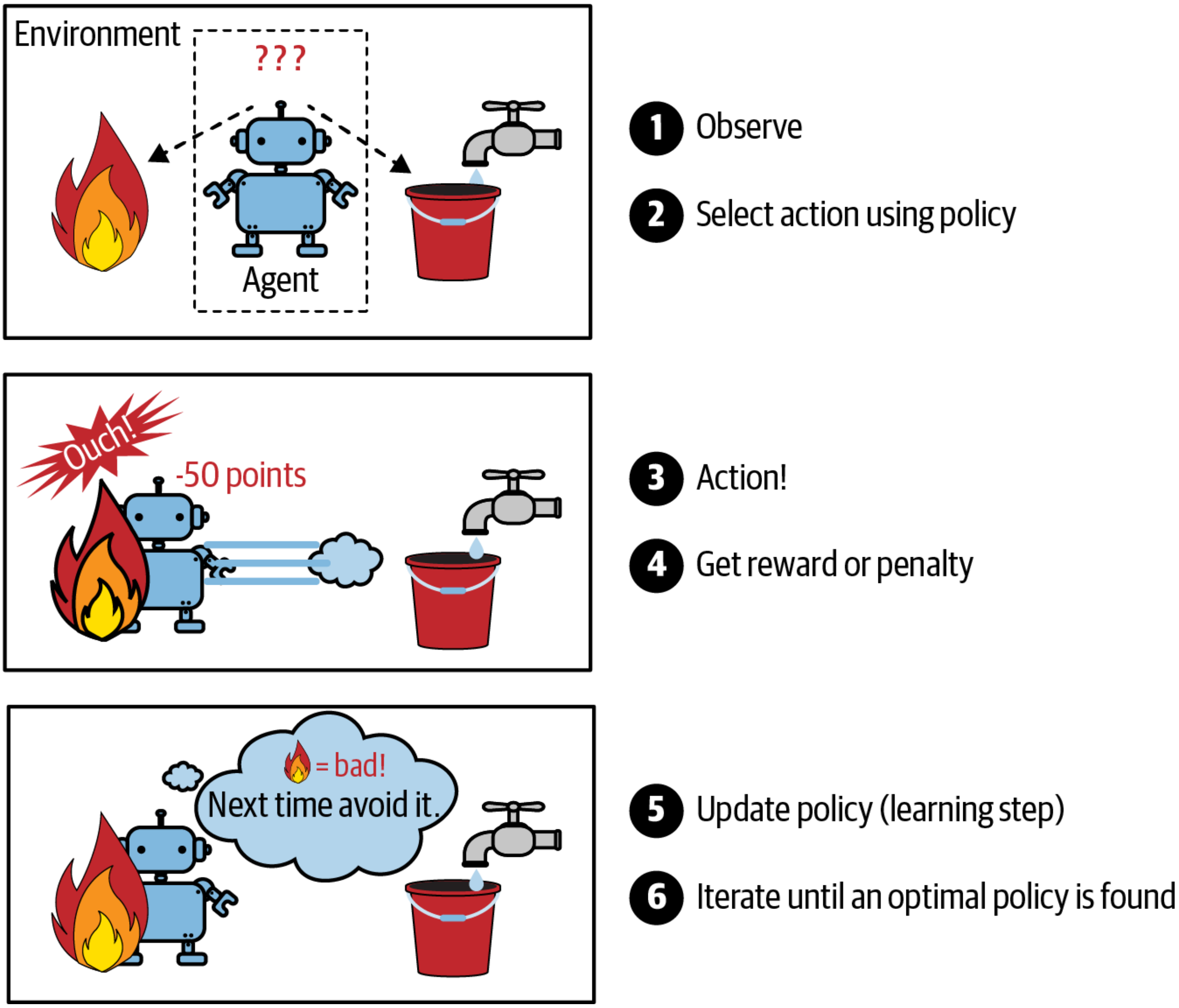
Some photo-hosting services, such as Google Photos, are good examples of this. Once you upload all your family photos to the service, it automatically recognizes that the same person A shows up in photos 1, 5, and 11, while another person B shows up in photos 2, 5, and 7. This is the unsupervised part of the algorithm (clustering). Now all the system needs is for you to tell it who these people are. Just add one label per person3 and it is able to name everyone in every photo, which is useful for searching photos.

Self-supervised learning: machine learning involves actually generating a fully labeled dataset from a fully unlabeled one.

For example, if you have a large dataset of unlabeled images, you can randomly mask a small part of each image and then train a model to recover the original image .During training, the masked images are used as the inputs to the model, and the original images are used as the labels.



Reinforcement learning: The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return,It must then learn by itself what is the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.


Environment
???
Agent
1 Observe
2 Select action using policy
Ouch!
-50 points
3 Action!
4 Get reward or penalty
= bad!
Next time avoid it.
5 Update policy (learning step)
6 Iterate until an optimal policy is found

DeepMind’s AlphaGo program is also a good example of reinforcement

Batch Versus Online Learning

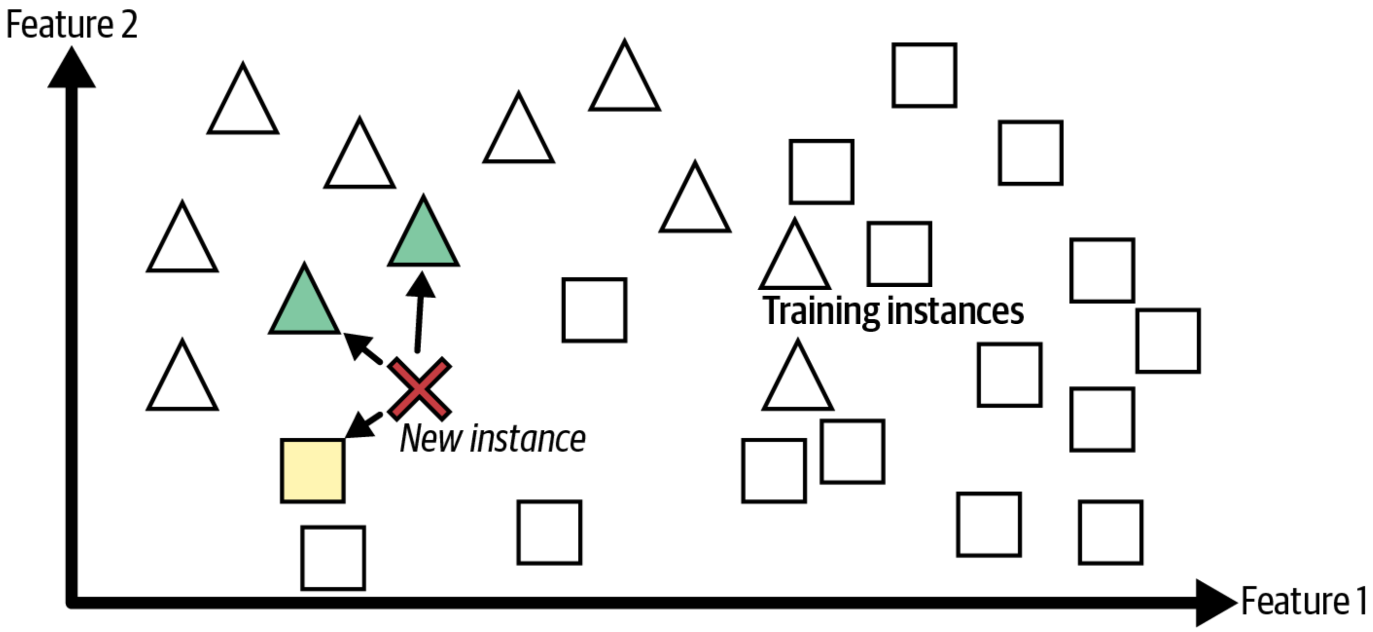
Feature	Batch Learning	Online Learning
Data Processing	Processes all training data at once.	Processes data incrementally, one or a few samples at a time.
Memory Usage	Requires large memory to handle entire dataset simultaneously.	Requires less memory as it processes data incrementally.
Training Time	Can be slow if the dataset is large, as it needs to process all data before updating the model.	Can be faster for initial training as it updates the model with each new sample.
Model Updates	Updates the model parameters after processing the entire dataset.	Continuously updates the model parameters with each new sample.
Adaptability	Less adaptable to new data; may require retraining on the entire dataset.	Highly adaptable to new data; can learn and update in real-time.
Use Cases	Suitable for static datasets where data does not change frequently.	Suitable for dynamic environments where data is continuously generated.
Complexity	Can be computationally expensive and complex for large datasets.	Generally simpler and more efficient for large, streaming datasets.
Examples	Standard implementations of algorithms like Random Forest, Support Vector Machines.	Algorithms like Stochastic Gradient Descent, Online Variants of Naive Bayes.
Convergence	May achieve better accuracy and stability due to processing full data.	Might require more careful tuning to achieve similar accuracy.
Batch Size	Utilizes the full dataset as a single batch.	Uses one or a few data points as a batch.
Scalability	Scalability can be an issue with very large datasets.	Scales well with large and continuously flowing datasets.
Resilience to Noise	Better at handling noise due to averaging effects over large datasets.	Can be sensitive to noisy data as each update can be influenced by noise.

`data drift or model rot problem`: a model’s performance tends to decay slowly over time, simply because the world continues to evolve while the model remains unchanged.

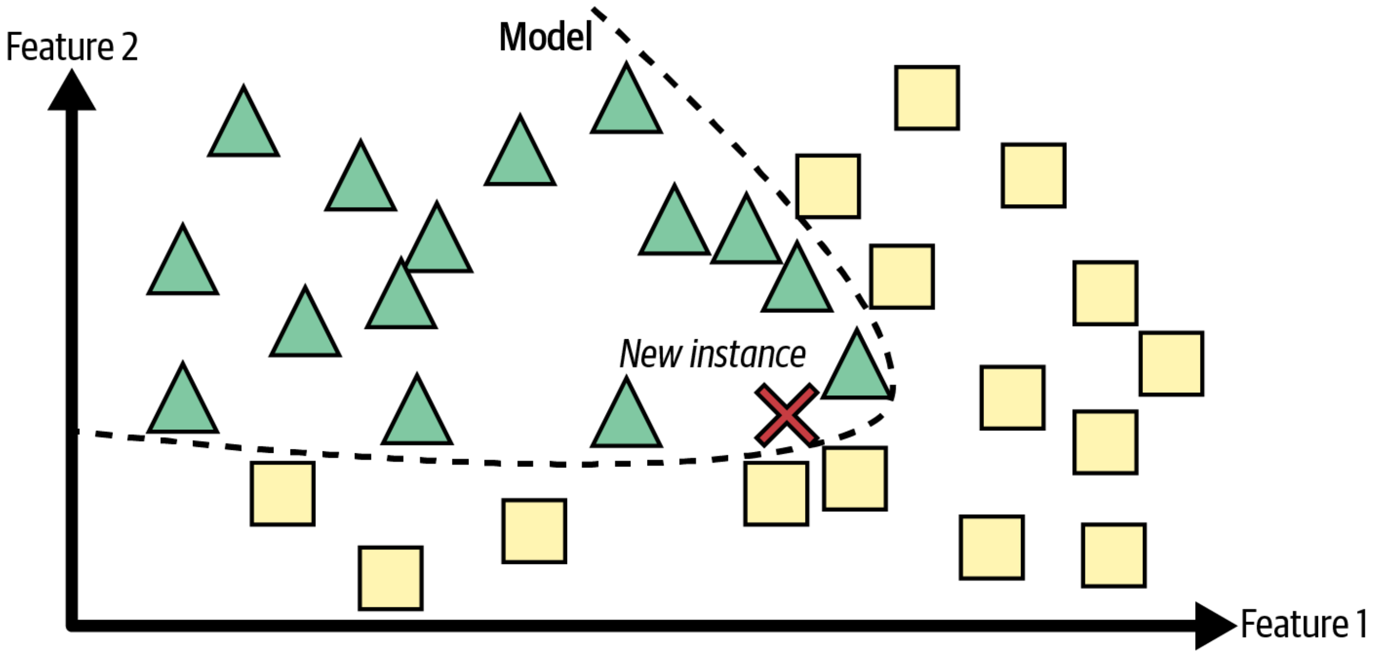
> The solution is to regularly retrain the model on up-to-date data.

Instance-Based Versus Model-Based Learning

Feature	Instance-Based Learning	Model-Based Learning
Definition	Memorizes the training instances and uses them to make predictions.	Builds a model based on the training data and uses this model to make predictions.
Examples of Algorithms	K-Nearest Neighbors (K-NN), Locally Weighted Learning	Linear Regression, Logistic Regression, Decision Trees, Neural Networks
Training Process	No explicit training phase; stores the instances.	Involves training phase where a model is learned from the data.
Prediction Process	Predictions are made by finding similar instances and averaging their outputs.	Predictions are made using the learned model.
Memory Usage	High memory usage as all instances need to be stored.	Lower memory usage after the model is built; only the model parameters are stored.
Speed of Prediction	Slower, as it requires searching through the entire dataset for similar instances.	Faster, as it uses the model directly to make predictions.
Adaptability	Easily adapts to new data; just add new instances.	Adaptation requires retraining or incremental learning.
Complexity	Simple to implement but can become computationally expensive with large datasets.	More complex to implement but efficient once the model is built.
Handling of Noise	Can be sensitive to noisy data points, which can affect predictions.	Can be more robust to noise if the model generalizes well.
Interpretability	Often less interpretable as it relies on the stored instances.	Often more interpretable, especially with simple models like linear regression.
Use Cases	Suitable for problems where the decision boundary is complex and not easily parameterizable.	Suitable for problems where the relationship between input and output can be captured by a model.



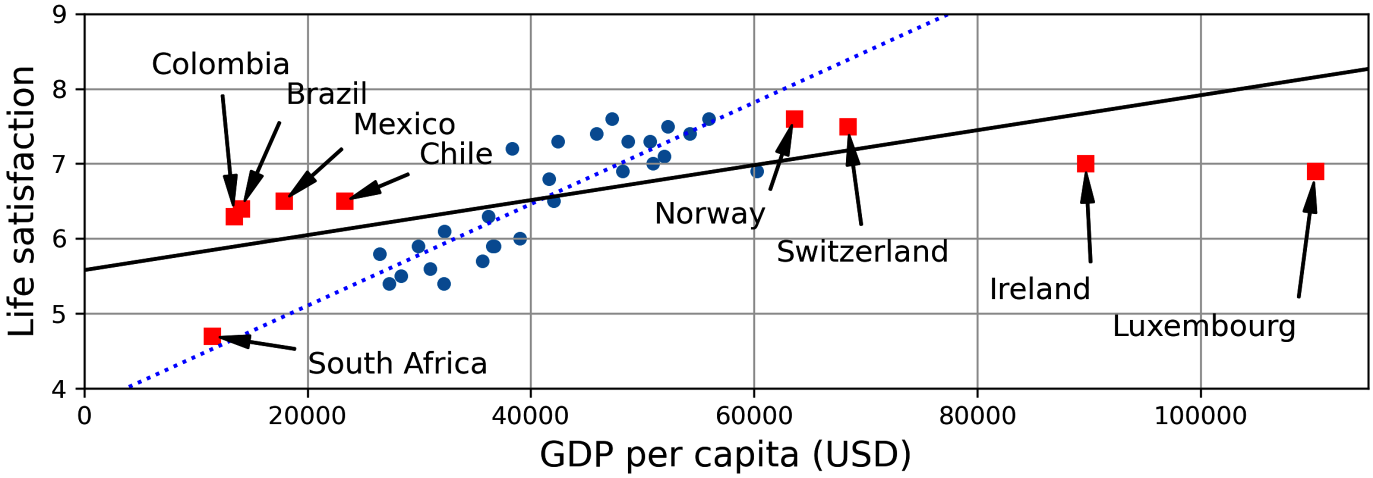
Instance-based learning



Model-based learning

Training a model means running an algorithm to find the model parameters that will make it best fit the training data, and hopefully make good predictions on new data.

Sampling Bias: occurs when a sample is not representative of the population from which it is drawn, leading to skewed or inaccurate conclusions. This bias can significantly affect the results and reliability of a study or analysis.



Types of Sampling Bias

1. **Selection Bias**: Occurs when the sample is not randomly selected. For example, surveying only the residents of a wealthy neighborhood to generalize about an entire city's income levels.
2. **Survivorship Bias**: Arises when only surviving subjects are considered, ignoring those that didn't make it through the selection process. For instance, studying the performance of companies listed in a stock index without considering those that went bankrupt.
3. **Volunteer Bias**: Happens when individuals self-select into a study, often leading to a non-representative sample. For example, a study on dietary habits where participants are volunteers may over-represent health-conscious individuals.
4. **Non-response Bias**: Occurs when a significant number of selected participants do not respond. For example, if a large portion of a survey's target group fails to respond, the responses collected might not reflect the views of the entire group.

Examples and Consequences

Example 1: Medical Study

- **Scenario**: A study on the effectiveness of a new drug is conducted using participants who volunteer for the trial.
- **Bias**: Volunteer Bias
- **Consequence**: The results might overestimate the drug's effectiveness because volunteers might be more health-conscious or more ill than the general population.

Example 2: Customer Feedback

- **Scenario**: A company collects customer feedback through an online survey.
- **Bias**: Non-response Bias

- **Consequence**: The feedback may be overly positive or negative depending on who chooses to respond, leading to skewed perceptions of customer satisfaction.

Identifying and Mitigating Sampling Bias

Identifying Sampling Bias

1. **Statistical Tests**: Use tests to compare sample statistics with known population parameters.
2. **Visual Inspection**: Compare the sample distribution with the population distribution.
3. **Comparison with Known Characteristics**: Check if the sample reflects the known characteristics of the population (e.g., age, gender, income distribution).

Mitigating Sampling Bias

1. **Random Sampling**: Use random sampling methods to ensure every individual has an equal chance of being selected.
2. **Stratified Sampling**: Divide the population into strata and randomly sample from each stratum to ensure representation.
3. **Increase Sample Size**: A larger sample size can reduce the impact of sampling bias.
4. **Weight Adjustments**: Adjust weights during analysis to compensate for over or under-represented groups.
5. **Ensure Representation**: Make sure the sampling method captures the diversity of the population.

Practical Example: Reducing Bias in a Study

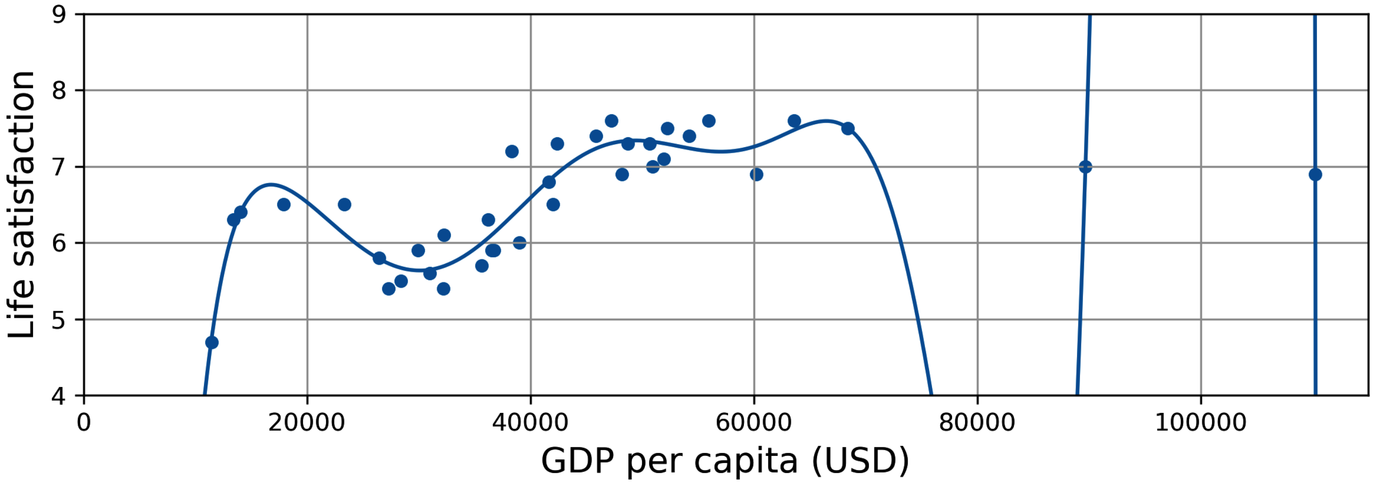
Consider a study aiming to understand the health behaviors of a city’s residents.

1. **Initial Plan**: Send a survey to residents who visit a local health clinic.
2. **Potential Bias**: Selection Bias and Volunteer Bias
3. **Mitigation**:
    - Use **random sampling** to select households across different neighborhoods.
    - Employ **stratified sampling** to ensure different socioeconomic strata are represented.
    - Provide incentives for participation to reduce **non-response bias**.
    - Use follow-up contacts to encourage responses from non-respondents.

Feature engineering

- *Feature selection*: selecting the most useful features to train on among existing features
- *Feature extraction*: combining existing features to produce a more useful one

Overfitting: means that the model performs well on the training data, but it does not generalize well.



solution

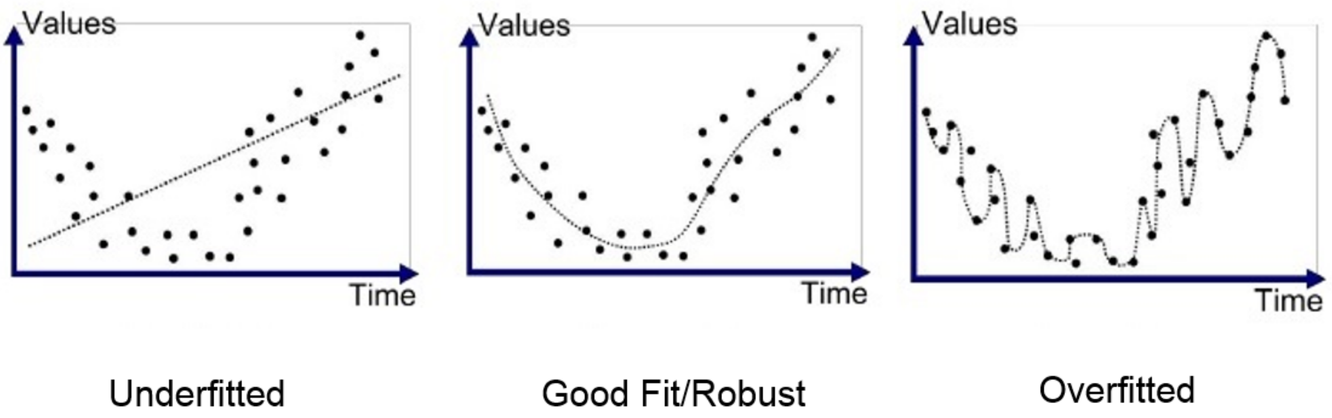
• Simplify the model by selecting one with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data, or by constraining the model. • Gather more training data. • Reduce the noise in the training data (e.g., fix data errors and remove outliers).

---

regularization: Constraining a model to make it simpler and reduce the risk of overfitting

The amount of regularization to apply during learning can be controlled by a hyper- parameter. A hyperparameter is a parameter of a learning algorithm (not of the model).

Underfitting: it occurs when your model is too simple to learn the underlying structure of the data.



solution:

• Select a more powerful model, with more parameters. • Feed better features to the learning algorithm (feature engineering). • Reduce the constraints on the model (for example by reducing the regularization hyperparameter).

Fine-tuning: make the model good fit.

Hyperparameters Tuning and Model Selection

Hyperparameter tuning and model selection are critical steps in the machine learning workflow. They help in optimizing the performance of machine learning models by finding the best parameters and selecting the most appropriate model for a given task.

Hyperparameter Tuning

Hyperparameters are parameters that are not learned during training but are set before the training process. Examples include learning rate, number of trees in a random forest, or the kernel in an SVM. Tuning these hyperparameters can significantly improve model performance.

Methods for Hyperparameter Tuning:

1. **Grid Search**: A brute-force approach where you specify a set of hyperparameters and try all possible combinations.
2. **Random Search**: Randomly samples the hyperparameter space and evaluates performance.
3. **Bayesian Optimization**: Uses a probabilistic model to find the best hyperparameters.
4. **Genetic Algorithms**: Uses evolutionary techniques to optimize hyperparameters.

5. **Hyperband**: Combines random search with early stopping to find a good set of hyperparameters efficiently.

Model Selection

Model selection involves choosing the best model among different candidates based on their performance. It can be achieved by evaluating models using metrics such as accuracy, precision, recall, F1 score, AUC-ROC, etc., on validation data.