Machine Learning is the science of teaching machines how to learn by themselves. Machine Learning is re-shaping and revolutionizing the world and disrupting industries and job functions globally.

Machine learning is so extensive that you probably use it numerous times a day without even knowing it. From unlocking your mobile phones using your face to giving your attendance using a biometric machine, machine learning is being used in almost every stage.

* Smartphones detecting faces while taking photos or unlocking themselves
* Facebook, LinkedIn or any other social media site recommending your friends and ads you might be interested in
* Amazon recommending you the products based on your browsing history
* Banks using Machine Learning to detect Fraud transactions in real-time

Machine Learning problems can be divided into 3 broad classes:

1. **Supervised Machine Learning** (input + known output → predict output)
   1. Regression (predict numbers)
   2. Classification (predict categories)
2. **Unsupervised Machine Learning** (no labels, find structure)
   1. Clustering
   2. Dimensionality Reduction
3. **Reinforcement Learning** (agent learns by reward/punishment) — just at a high level for now

**Regression:**

1. Linear Regression

What it is: Draws a straight line to fit the data.

Use case: When the relationship between inputs and output looks roughly linear.

Example: Predicting house price based on size.

2. Multiple Linear Regression

What it is: Like linear regression but with several input variables instead of one.

Use case: Predicting price using size, location, and number of rooms all together.

3. Polynomial Regression

What it is: Fits a curved line to capture more complex relationships.

Use case: When your data forms a curve rather than a straight line.

4. Ridge Regression (L2 Regularization)

What it is: Linear regression with a penalty to shrink large coefficients.

Use case: Helps prevent overfitting when you have many correlated features.

5. Lasso Regression (L1 Regularization)

What it is: Similar to Ridge, but can also push some coefficients to zero (automatic feature selection).

Use case: When you want a simpler model that ignores unimportant features.

6. Elastic Net Regression

What it is: Combines Ridge and Lasso penalties.

Use case: When you need a balance between Ridge and Lasso’s behavior.

7. Logistic Regression (for classification)

Special note: Despite its name, logistic regression is used for classification, not predicting numbers. (It predicts probabilities of categories.)

8. Other Variants (Advanced)

Stepwise Regression: Adds or removes predictors automatically.

Quantile Regression: Predicts a specific quantile (like median) instead of the mean.

Support Vector Regression (SVR): Uses SVM concepts for predicting numbers.

Decision Tree / Random Forest Regression: Nonlinear regression using tree-based models.

**Classification:**

Classification is a **supervised machine-learning** task where an algorithm learns to assign input data to one of several **predefined categories or labels**. Classification is like teaching a system to sort things into boxes. You show it examples of things and the box they belong to (this is your labeled data). Later, when you give it something new, it tries to put it in the right box based on what it has learned.

* **Input:** Features the details about the thing (data attributes - color, size, words in a message).
* **Output:** A discrete class label - The box or category it belongs to (e.g., spam or not spam, disease present or absent)

Classification itself has several **subtypes** depending on the number and nature of labels:

| **Type** | **Description** | **Example** |
| --- | --- | --- |
| **Binary Classification** | Two possible classes. Only two boxes to choose from. | Spam vs. Not Spam; Fraud vs. Not Fraud |
| **Multiclass Classification** | More than two classes. More than two boxes, but each item can only go into one. | Classifying animals as cat, dog, bird |
| **Multilabel Classification** | Each sample can belong to multiple labels simultaneously.  An item can go into several boxes at once. | A news article tagged as “Sports” and “Politics” |
| **Ordinal Classification** | Classes have a natural order. The boxes have a natural order (rank). | Movie ratings: Poor, Average, Good, Excellent |
| **Imbalanced Classification** | One class has far fewer samples than others, One box has way fewer examples than others, making it harder to learn. | Fraud detection (fraudulent transactions are rare) |

 **Classification:** Output is a **category** (discrete).

 **Regression:** Output is a **continuous number** (e.g., predicting price, temperature).

 **Clustering:** Unsupervised grouping (no predefined labels).

 **Recommendation/Ranking:** Suggesting or ordering items.

 Supervised vs. Unsupervised Learning

 Overfitting, underfitting, bias-variance tradeoff

 Cross-validation and performance metrics

**Supervised Learning Algorithms**

Learn the most widely used algorithms:

* **Regression:** Linear, polynomial, regularized (Ridge, Lasso)
* **Classification:** Logistic regression, k-NN, SVM, Decision Trees, Random Forests
* **Gradient Boosting:** XGBoost, LightGBM
* **Neural Networks:** Basic feed-forward networks

Practice by building models on datasets (Kaggle, UCI Repository).

| **Task** | **Algorithms to Cover First** |
| --- | --- |
| **Regression** | Linear regression, regularized regression (Ridge, Lasso) |
| **Classification** | Logistic regression, k-Nearest Neighbors, Decision Trees, Random Forests |
| **Clustering** | k-Means, Hierarchical |
| **Dimensionality Reduction** | PCA (Principal Component Analysis) |

## 5. ****Unsupervised Learning****

Discover patterns without labels:

* **Clustering:** k-Means, Hierarchical, DBSCAN
* **Dimensionality Reduction:** PCA, t-SNE, UMAP
* **Anomaly Detection**

**Model Evaluation & Tuning**

* Metrics: Accuracy, precision, recall, F1, AUC, RMSE
* Hyperparameter tuning: GridSearchCV, RandomizedSearchCV, Bayesian optimization
* Feature selection and engineering

**🟦 7. Deep Learning**

Once you’re comfortable with classic ML:

* **Neural Networks Basics**
* **Convolutional Neural Networks (CNNs):** For image data
* **Recurrent Neural Networks (RNNs), LSTM, GRU:** For sequences
* **Transformers:** For NLP
* Frameworks: **TensorFlow** or **PyTorch**

**🟨 8. Specialized Domains**

Choose based on interest:

* **Natural Language Processing (NLP):** Text classification, embeddings, BERT
* **Computer Vision:** Object detection, segmentation
* **Time Series Forecasting:** ARIMA, Prophet, LSTMs
* **Reinforcement Learning:** Q-learning, policy gradients

**🟧 9. ML in Production**

Learn how to deploy and maintain models:

* Saving models (joblib, pickle, ONNX)
* Serving APIs (Flask, FastAPI, Django)
* MLOps: CI/CD for ML, model monitoring, drift detection
* Cloud Platforms: AWS SageMaker, Azure ML, Google Vertex AI

**🟥 10. Projects & Portfolio**

Build a portfolio that shows **breadth and depth**:

* Spam email classifier (binary classification)
* Movie recommendation engine (collaborative filtering)
* Customer segmentation (clustering)
* Image classifier (CNN)
* NLP sentiment analyzer (transformers)

Host on GitHub, write short blog posts or LinkedIn summaries of your learnings.

**🟩 11. Keep Learning**

* Follow **Kaggle competitions** to practice.
* Read research papers or use Arxiv-sanity to stay updated.
* Take courses:
  + Andrew Ng’s ML Specialization (Coursera)
  + fast.ai Practical Deep Learning for Coders
  + Google’s ML Crash Course