Machine Learning Concepts and Algorithms

Rishabh Johri

[rjohri@deloitte.com](mailto:rjohri@deloitte.com)

# Introduction to Machine Learning

## What is Machine Learning

Machine Learning (ML) is a field of Artificial Intelligence where computers learn patterns from data and make predictions or decisions without explicit programming. Instead of writing fixed rules, we feed a model with historical data so it can recognize trends and generalize to unseen data. ML systems improve as they get more data, enabling automation of complex decision-making.

## Difference between AI, ML, and Deep Learning

Artificial Intelligence (AI): The broad goal of creating systems that mimic or extend human intelligence, including reasoning, problem solving, and natural language understanding.

Machine Learning (ML): A subset of AI focused on algorithms that can learn patterns and improve performance automatically from data.

Deep Learning (DL): A subset of ML that uses deep neural networks with many layers to learn complex representations, especially effective for images, text, audio, and sequential data.

## Types of Learning

Supervised Learning: The algorithm is trained on labeled data (inputs with known outputs). The goal is to predict future outputs. Example: predicting house prices based on size and location.

Unsupervised Learning: The algorithm explores unlabeled data to find hidden structures, such as grouping customers into clusters or reducing dimensionality.

Semi-supervised Learning: Uses a small amount of labeled data and a large amount of unlabeled data to improve performance when labeling is expensive.

Reinforcement Learning: An agent interacts with an environment, taking actions to maximize a reward signal over time. This is how systems like AlphaGo learn to play games.

## ML Workflow or Pipeline

A typical ML process involves:  
- Problem Definition  
- Data Collection & Cleaning  
- Feature Engineering & Selection  
- Model Selection & Training  
- Evaluation & Hyperparameter Tuning  
- Deployment & Monitoring

## Real-world Applications

Personalized recommendations on streaming and e-commerce platforms  
Fraud detection in banking and payments  
Predictive maintenance in manufacturing  
Self-driving vehicles and robotics  
Healthcare applications such as disease detection and drug discovery

# Core ML Concepts

## Bias-Variance Tradeoff

Bias is the error introduced by making simplifying assumptions in the model. High bias leads to underfitting (model is too simple). Variance is the model’s sensitivity to training data fluctuations. High variance leads to overfitting (model fits noise instead of the signal). The goal is to achieve a balance where both bias and variance are reasonably low, minimizing total error.

## Cross-Validation Techniques

Cross-validation evaluates model performance reliably by splitting the dataset into training and validation sets multiple times:  
- K-Fold Cross-Validation: Split the data into k parts; train on k-1 and test on the remaining part. Repeat k times.  
- Stratified K-Fold: Keeps class distribution balanced across folds.  
- Leave-One-Out (LOOCV): Each single data point becomes the validation set once. Useful for small datasets but computationally heavy.  
- Time Series CV: Keeps the time order intact, expanding training data over time.

## Confusion Matrix

A confusion matrix shows how a classifier performs by comparing predicted and actual classes.  
Accuracy = (TP + TN)/Total  
Precision = TP/(TP + FP)  
Recall = TP/(TP + FN)  
F1-score = Harmonic mean of precision and recall.

## ROC Curve and AUC

The ROC Curve plots the trade-off between True Positive Rate and False Positive Rate at various thresholds. AUC (Area Under Curve) measures overall ability to distinguish classes. AUC=1 means perfect separation; 0.5 means random guessing.

## Precision-Recall Tradeoff

Precision and recall often conflict: higher precision means fewer false positives but possibly more missed positives; higher recall means catching more positives but possibly with more false alarms. Choice depends on the problem (e.g., cancer detection needs high recall, spam filtering favors high precision).

## Overfitting vs Underfitting

Overfitting: Model captures noise and performs poorly on unseen data. Happens with high complexity and low regularization.  
Underfitting: Model is too simple, failing to learn important patterns.  
Prevention: use cross-validation, regularization, pruning, early stopping, or more training data.

## Regularization Techniques

Regularization adds a penalty to model complexity to improve generalization:  
- L1 (Lasso): Forces some coefficients to zero (feature selection).  
- L2 (Ridge): Shrinks large weights but keeps all features.  
- ElasticNet: Combines L1 and L2.  
Other methods include dropout in deep networks, early stopping, and data augmentation.

# Classification Algorithms

## Logistic Regression

Models the probability of a class using the logistic (sigmoid) function.  
Equation: P(y=1|X)=σ(w^T X + b)  
Pros: Simple, fast, interpretable.  
Cons: Limited to linear decision boundaries.

## k-Nearest Neighbors (kNN)

Classifies a point based on the majority label among its k closest neighbors.  
Pros: Simple, no explicit training.  
Cons: Computationally heavy for large datasets, sensitive to scaling and irrelevant features.

## Support Vector Machines (SVM)

Finds the optimal hyperplane that maximizes the margin between classes. Can use kernels for non-linear data.  
Pros: Works well for high-dimensional data.  
Cons: Can be slow for very large datasets and needs careful parameter tuning.

## Decision Trees

Splits data based on features to maximize information gain or minimize impurity.  
Pros: Easy to interpret and visualize, handles both numerical and categorical data.  
Cons: Prone to overfitting without pruning.

## Random Forest

An ensemble of decision trees built using bagging (bootstrap aggregating).  
Pros: Reduces overfitting, handles missing values and noise well.  
Cons: Less interpretable than a single tree.

## Naive Bayes

Uses Bayes’ theorem with the assumption that features are conditionally independent given the class.  
Pros: Fast, efficient for text classification and spam detection.  
Cons: Independence assumption rarely holds but often works surprisingly well.

## Gradient Boosting (XGBoost, LightGBM, CatBoost)

Builds trees sequentially where each new tree fixes the errors of the previous ones.  
XGBoost: Regularized and highly optimized.  
LightGBM: Faster with histogram-based splits.  
CatBoost: Handles categorical features natively and avoids overfitting.  
Pros: State-of-the-art accuracy for structured data.  
Cons: Requires careful tuning and can overfit if not regularized.

## Evaluation Metrics for Classification

Accuracy: Simple but can mislead on imbalanced data.  
Precision & Recall: Better for imbalanced classes.  
F1-score: Balance between precision and recall.  
ROC-AUC: Overall ability to separate positive and negative classes.

# Regression Algorithms

## Linear Regression

Models the relationship between input features and a continuous target using a straight line.  
Equation: y = wX + b  
Assumes linearity, independence of errors, and equal variance (homoscedasticity).

## Polynomial Regression

Extends linear regression by adding polynomial terms of the features, enabling modeling of curved relationships.

## Regularization for Regression

Ridge Regression (L2): Adds squared magnitude penalty to shrink coefficients.  
Lasso Regression (L1): Can force some coefficients to exactly zero, performing feature selection.  
ElasticNet: Combines L1 and L2 for balanced regularization.

## Evaluation Metrics for Regression

Mean Absolute Error (MAE): Average absolute difference between predictions and actuals.  
Mean Squared Error (MSE): Penalizes larger errors more heavily.  
Root Mean Squared Error (RMSE): Square root of MSE, same units as target.  
R²: Measures how much variance in the target is explained by the model.