Machine Learning

What is Machine Learning?

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that enables machines to learn patterns from data and make decisions or predictions without being explicitly programmed.

Example: Email spam filters, recommendation systems, self-driving cars.

Why is ML Important?

- Automates repetitive tasks.
- Improves decision-making with data.
- Enables real-time insights and predictions.
- Powers AI-driven products (e.g., Google Search, Netflix).

ML Project Lifecycle / Pipeline

1. Problem Definition
2. Data Collection
3. Data Preprocessing
4. Feature Engineering
5. Model Selection
6. Model Training
7. Model Evaluation
8. Hyper parameter Tuning
9. Model Deployment
10. Monitoring & Maintenance

Data – The Heart of ML

- **Types**: Structured (tables), Unstructured (text, images), Semi-structured (JSON, XML)
- **Data Quality**: Missing values, outliers, noise, duplicates
- **Data Splitting**:
    - Training Set (e.g., 70%)
    - Validation Set (e.g., 15%)
    - Test Set (e.g., 15%)

Data Preprocessing

- Handling missing values (mean, median, delete)
- Removing duplicates
- Encoding categorical variables (One-Hot, Label)
- Normalization / Standardization
- Dealing with class imbalance (SMOTE, oversampling)

Feature Engineering

- **Feature Extraction**: Create new features from raw data (e.g., extract date from timestamp)
- **Feature Selection**: Choose the most relevant features (e.g., using correlation, Chi-Square test)

Types of Machine Learning

Supervised Learning

- Input: Features + Labels
- Used for: Classification & Regression
- Examples: Linear Regression, Logistic Regression, Decision Trees, Random Forest, SVM

Unsupervised Learning

- Input: Features only (no labels)
- Used for: Clustering, Anomaly Detection
- Examples: K-Means, DBSCAN, PCA

Reinforcement Learning

- Agent interacts with an environment to maximize reward
- Examples: AlphaGo, self-driving cars

Important Algorithms

Task	**Algorithms**
Regression	Linear Regression, Decision Tree Regressor
Classification	Logistic Regression, KNN, SVM, Naive Bayes
Clustering	K-Means, Hierarchical
Dimensionality Reduction	PCA, t-SNE
Reinforcement Learning	Q-Learning, Policy Gradient

Bias-Variance Trade-off

- **Bias**: Error due to overly simplistic assumptions (underfitting)
- **Variance**: Error due to sensitivity to noise in training data (overfitting)
- Goal: Find a balance between bias and variance for optimal performance

Ensemble Learning

- **Bagging**: Build multiple models (e.g., Random Forest)
- **Boosting**: Sequentially build models that correct predecessors (e.g., XGBoost, AdaBoost)
- **Stacking**: Combine outputs of multiple models with a meta-model

Cross-Validation

- **K-Fold Cross Validation**: Split dataset into k parts and rotate training/testing
- Ensures stability and generalization

Model Training

- **Loss Function**: Measures prediction error (MSE, Cross-Entropy)
- **Optimizer**: Minimizes the loss (e.g., Gradient Descent, Adam)
- **Back propagation**: Used in neural networks to update weights

Model Evaluation

Classification Metrics:

- Accuracy
- Precision, Recall, F1-Score
- Confusion Matrix
- ROC-AUC

Regression Metrics:

- MAE, MSE, RMSE
- R^2 Score

Classification Metrics:

Accuracy

Accuracy=TP+TN/TP+TN+FP+FN

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

Precision

Precision=TP/TP+FP

How many predicted positives were actually correct?

Recall (Sensitivity / True Positive Rate)

Recall=TP/TP+FN

How many actual positives did we correctly identify?

F1-Score (Harmonic Mean of Precision & Recall)

F1-Score=2×Precision×Recall/Precision+Recall

Confusion Matrix

A 2×2 matrix for binary classification:

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

ROC-AUC Score

- **ROC**: Receiver Operating Characteristic curve — plots TPR vs FPR.
- **AUC**: Area under ROC curve (0 to 1). Higher is better.

No fixed formula, but computed using integration of the ROC curve.

Regression Metrics

Mean Absolute Error (MAE)

$MAE=1/n * \sum^{n}_{i=1}|y_i-y\hat{}_i|$ (summation value from i=1 to n)

Average of absolute differences between predicted and actual values.

Mean Squared Error (MSE)

$MSE=1/n * (\sum^{n}_{i=1}(y_i-y\hat{}_i)^2$ (summation value from i=1 to n)

Penalizes large errors more than MAE.

Root Mean Squared Error (RMSE)

$RMSE=root(MSE)= root(1/n * (\sum^{n}_{i=1}(y_i-y\hat{}_i)^2)$ (summation value from i=1 to n)

R^2 Score (Coefficient of Determination)

$R^2 = 1- (\sum^{n}_{i=1}(y_i-y\hat{}_i)^2/\sum^{n}_{i=1}(y_i-y^{-})^2)$ (summation value from i=1 to n)

Where:

- y_i = actual value

- $\hat{y}_i$= predicted value
- $\bar{y}$ = mean of actual values

  Measures how well the model explains the variance in data (1 = perfect fit, 0 = worst fit).

Model Interpretability Techniques

- **Feature Importance**
- **SHAP values**
- **LIME**
- **Partial Dependence Plots (PDP)**

Hyper parameter Tuning

- Learning Rate, Max Depth, Number of Epochs, etc.
- **Techniques**: Grid Search, Random Search, Bayesian Optimization

Model Deployment

- Tools: Flask, FastAPI, Docker, Streamlit/Gradio, AWS/GCP/Azure

Model Monitoring

- Track: Model Drift, Data Drift, Performance, Retraining triggers

Experiment Tracking

- Track metrics, parameters, artifacts, versions
- Tools: MLflow, Weights & Biases

Data Versioning & ML Pipelines

- Tools: DVC, Kubeflow, Airflow
- Purpose: Ensure reproducibility and automation

Tools & Libraries

Purpose	Tools
Data Handling	Pandas, NumPy
Visualization	Matplotlib, Seaborn, Plotly
ML Models	Scikit-learn, XGBoost, LightGBM
Experiment Tracking	MLflow, Weights & Biases
Notebooks	Jupyter, Google Colab

Dimensionality Reduction (Advanced)

- Reduce computation and noise
- Algorithms: PCA, t-SNE, UMAP

Anomaly Detection

- Identify outliers/fraud
- Algorithms: Isolation Forest, One-Class SVM, Autoencoders

Real-World Applications

- Healthcare: Diagnosis, Imaging
- Finance: Risk, Fraud Detection
- Retail: Recommendations
- Education: Adaptive learning
- Agriculture: Yield prediction
- Transportation: Self-driving, Route optimization

Challenges in ML

- Data Privacy & Security
- Model Bias & Fairness
- Interpretability
- Overfitting/Underfitting
- Scaling to large datasets
- Deployment in production

Ethics & Responsible AI

- Fairness, Transparency, Privacy, Safety
- Avoid harmful outcomes

Future of ML

- Federated Learning
- AutoML
- Quantum ML
- Explainable AI (XAI)
- Edge AI