Unit 1

17 September 2025 16:31

Machine Learning: Machine Learning is a field or subset of Artificial Intelligence that allows a system or a machine to learn from experience aka data and make predictions based on it. The machine learns from the dataset and understands the underlying pattern to generate output for similar but new data without being explicitly programmed for that.

Types of Data:

- Numerical Data
- Categorical Data
- Ordinal Data

ML Workflow: Data Collection -> Data Preprocessing -> Model Selection -> Training -> Evaluation -> Deployment

ML Lifecycle:

- Gathering Data.
- Data Preparation (load and randomize the data order).
- Data Exploration: Understanding the dataset(Correlation, trends, outliers, type of dataset).
- Data Wrangling: Handling inconsistencies, redundancy, missing values, noise reduction, etc.
- Data Preprocessing: Normalization, Encoding, Feature Engineering.
- Data Splitting: Training and Test data are split in 70/80-30/20 ratio.
- Model Selection: Selecting the correct ML model/algorithm for the dataset.
- Model Training: Model is trained on the training dataset.
- Model Evaluation: Trained model is evaluated based on its prediction on the test dataset.
- Hyperparameter Tuning: Adjusting hyperparameters such as learning rate, regularization strength, no. of decision trees, etc to avoid instability in the model.
- Model Deployment.
- Monitoring & Maintenance.

Examples of ML:

- Spam detection in email
- House price prediction
- Image recognition
- Object detection
- Recommendation systems
- Autonomous Vehicle (Self-driving)

Tools for ML:

- Scikit Learn (sklearn): a python library with ML algorithms such as regression, clustering, classification, etc.
- TensorFlow: Library for deep learning (neural network)
- PyTorch: Framework for building ML models, especially in the field of research.
- Google Colab: Cloud-based Jupyter like notebook with free GPU support.

Pros:

- Can handle large and complex datasets.
- Automates decision making process or repetitive tasks.
- Improves with more data.

Cons:

- Requires large & quality datasets for accurate results.
- Bias in the training dataset can reflect upon ML prediction.
- High computational costs.

Linear Regression: It is used to model the relationship between independent variables and a dependent variable by fitting a straight line through the observed data.

- Formula: $y = wx+b$.

- Here, y= dependent variable
- w= weight or slope of the line
- x= independent variable
- b= bias or intercept.

In linear regression, we try to find the best values of weight(w) and bias(b) such that the predicted values y in testing dataset are as close as possible to the actual value y in the training/observed dataset. Different variations of weight and bias are compared using the mean squared error or MSE.

- Formula: MSE= $(1/N) \cdot \Sigma (y_i - \hat{y}_i)^2$
- Here, N= total number of datapoints
- Y_i= Actual output
- $\hat{Y}_i$= Predicted output by the model

Workflow of LR:

- Initialize parameters w & b randomly or with zeros.
- Compute predictions for all datapoints using the current w and b.
- Calculate MSE (loss).
- Use Gradient Descent or Stochastic Gradient Descent to update w and b:
  - Update rule:
    - w := w - learning_rate × ∂MSE/∂w
    - b := b - learning_rate × ∂MSE/∂b
- Repeat steps 2–4 for several iterations (epochs) until convergence (loss stabilizes).

Types of Linear Regression:

- Simple LR: Output variable is dependent on only one independent variable.
- Multivariate LR: Output variable is dependent on multiple independent variables.

Limitations of LR:

- Linearity Assumption.
- Multicollinearity.
- Outliers Sensitivity.
- Overfitting and Underfitting.

Non Linear Regression: Non-Linear Regression is a type of regression analysis where the relationship between the independent variables (features) and the dependent variable (target) is modelled as a non-linear function. It assumes a non-linear relationship, allowing for flexible and more accurate modelling of complex relationships.

- Formula: $y = a \cdot x^2 + b \cdot \sin(x) + c$
  Where
  - a, b, c= Parameters to learn
  - x= Input feature
  - y= Target variable

Example of NLR:

- Bacteria growth.
- Stock prices.
- Physical phenomenon (population dynamics).

Workflow of NLR:

- Define a non-linear model (polynomial, log, logistic, etc).
- Use an iterative optimization algorithm to minimize the loss function.
- Check performance using metrics such as, RMSE(Root Mean Squared Error), MAE(Mean Absolute Error), R^2.
- Compare predicted curve vs actual curve.
- Hyperparameter Tuning.

Types of Non Linear Regression:

- Polynomial Regression
- Exponential Regression

- Logarithmic Regression
- Logistic Regression (Classification)

Limitations of NLR:

- More complex models require more data to generalize well.
- Higher risk of overfitting if the model is too complex.
- Harder to interpret compared to linear regression.

Key Points:

- **Loss Function:** A loss function measures how far are the model's predicted values from the actual values and it gives something to the optimizer to minimize.

  Choosing loss function metrics:
  - **MSE / RMSE:** Best when large errors should be penalized more (common in linear regression).
  - **MAE:** Best when robustness to outliers is needed.

- **Optimization Algorithms:** Method used to minimize or maximize objective function (usually loss function) to find the best parameters that give the lowest possible error between predicted and actual values. Since we can't randomly guess the coefficients in a regression model, we start with random or zero coefficients then switch to optimization algorithms to systematically adjust the coefficients so the model learns from the data and gives the closest possible prediction to actual values.

  There are 2 types of optimization algorithms:
  - Closed-form solutions: When we can solve for the best coefficient mathematically in one step. It provides exact solution without any iteration but limited to simple problems.
  - Iterative Optimization: Algorithms which iteratively adjust the coefficients for best result.

- **Hyperparameter Tuning:** Parameters are learned from data while Hyperparameters are set by us before training to control the learning process. Hyperparameter tuning is the process of systematically searching for the best hyperparameter values that given the most accurate and generalizable model.

  - In Linear regression, the basic Ordinary Leas Square (OLS) has no hyperparameters, only parameters. But with regularized versions (to handle overfitting and multicollinearity), hyperparameters come into play:
    - Ridge Regression (L2 Regularization): $J(\beta)=\sum(y-\hat{y})^2+\lambda\sum\beta^2$
      - Hyperparameter: λ (controls penalty strength).
      - Large λ: shrinks coefficients toward zero (reduces variance).

    - Lasso Regression (L1 Regularization): $J(\beta)=\sum(y-\hat{y})^2+\lambda\sum|\beta|$
      - Hyperparameter: λ
      - Can shrink some coefficients to exactly zero (feature selection effect).

    - Elastic Net: $J(\beta)=\sum(y-\hat{y})^2+\lambda_1\sum|\beta|+\lambda_2\sum\beta^2$
      - Hyperparameters: λ_1,λ_2
      - Mix of Ridge + Lasso.

  - In Non Linear Regression, more hyperparameters exists such as the polynomial degree, kernel type, learning rate, etc.

  - Tuning Methods:
    - **Grid Search**: Try all combinations of hyperparameters (exhaustive, slow).
    - **Random Search**: Randomly samples hyperparameters (faster, often effective).
    - **Bayesian Optimization/AutoML**: Uses probability models to search smartly.

- **Cross Validation:** It is a model evaluation technique used to test how well a machine learning model generalizes to unseen data.
  - Instead of training on one dataset and testing on a fixed split, cross-validation repeatedly splits the dataset

into different train/test subsets.
- This reduces bias from any single train-test split and gives a more reliable estimate of model performance.

Types of Cross validation:

- **K-Fold Cross-Validation**
    - Dataset split into k folds (say, k=5).
    - Train on k—1 folds, test on the remaining fold.
    - Repeat k times, each fold used once as test set.
    - Final performance = average of all folds.
    - Most common and reliable.
- **Leave-One-Out (LOO)**
    - Extreme case of k-fold with k=n (number of data points).
    - Train on all but one sample, test on the remaining one.
    - Very accurate, but **computationally expensive**.
- **Stratified K-Fold**
    - Ensures each fold has the same class distribution (important in classification, less in regression).