01 Machine Learning Model

01 The ML Model

An ML model is a mathematical representation of a system that learns from data to make predictions or decisions

- **Supervised Learning**
  - (e.g., Linear Regression, Decision Trees)
  - The model is trained on labeled data (input-output pairs)
- **Unsupervised Learning**
  - (e.g., Clustering, Dimensionality Reduction, Association Learning, K-Means, PCA)
  - The model identifies patterns or structures in unlabeled data
- **Semi-Supervised Learning**
  - Falls between supervised and unsupervised learning
  - It uses both labeled and unlabeled data to build a model, which is particularly useful when labeling data is expensive or time-consuming, but a large amount of unlabeled data is readily available
- **Reinforcement Learning**
  - (e.g., Q-Learning)
  - The model learns through interactions with an environment to maximize a reward signal

02 Model Parameters and Hyperparameters

- **Model Parameters**
  - These are the internal variables that the model learns from the data during training
  - eg: in a linear regression model, the weights (coefficients) are the parameters
- **Hyperparameters**
  - These are the settings or configurations specified before training
  - They control the learning process and must be tuned
  - eg: learning rate in gradient descent, the number of trees in a random forest, or the number of clusters in K-means

03 Loss Function

The loss function quantifies how well the model's predictions match the actual data. It measures the error between the predicted output and the true output

- **Mean Squared Error (MSE)**
  - Used in regression, it measures the average squared difference between the predicted and actual values
- **Cross-Entropy Loss**
  - Used in classification, it measures the difference between the predicted probability distribution and the actual distribution (labels)

04 Gradient Descent

Gradient Descent is an optimization algorithm used to minimize the loss function. The basic idea is to update the model parameters in the opposite direction of the gradient of the loss function with respect to the parameters

- **Compute Gradient :** Calculate the gradient of the loss function with respect to each parameter
- **Update Parameters :** Adjust the parameters by moving them slightly in the direction that reduces the loss
- **Learning Rate :** A hyperparameter that controls the size of the step taken during each update

05 Train-Test Split

Before training the model, the dataset is typically split into

- **Training Set :** Used to train the model
- **Testing Set :** Used to evaluate the model's performance on unseen data
  The train-test split helps to ensure that the model generalizes well to new data and is not simply memorizing the training data

06 Training the Model

- Training involves feeding the training data to the model and adjusting the parameters based on the loss function and gradient descent
- The training process continues until the model converges (i.e., further training does not significantly reduce the loss)

07 Model Evaluation

After training, the model is evaluated on the test set using various metrics depending on the task

- **Accuracy**
  - The proportion of correctly predicted labels (used in classification)
- **Precision and Recall**
  - Measures of how many true positives are correctly identified versus false positives (used in classification)
- **F1 Score**
  - The harmonic mean of precision and recall
- **R-squared**
  - A statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable (used in regression)
- **Confusion Matrix**
  - A table used to describe the performance of a classification model

08 Hyperparameter Tuning

To improve the model's performance, hyperparameters are tuned

- **Grid Search :** Testing all possible combinations of hyperparameters
- **Random Search :** Randomly sampling hyperparameters
- **Bayesian Optimization :** Using a probabilistic model to choose hyperparameters more efficiently

09 Cross-Validation

- Cross-validation is a technique for assessing how the results of a model will generalize to an independent dataset
- It involves partitioning the dataset into a set of training and validation sets multiple times, training the model on each training set, and evaluating it on the corresponding validation set

10 Deployment, Monitoring and Maintenance