Machine learning:-

① Linear Regression :-

(1) Fitting the line
(2) making predictions.

---

② Logistic Regression :-

(1) Linear combination.
(2). Sigmoid function.
(3). Decision boundary.
(4). Loss function : Log-loss (Cross Entropy loss).

Assumptions:

(1) Binary Outcome
(2). Independent observations.
(3) Linearity of log-odds.
(4). No multicollinearity.

(3) Decision trees:-

(1) Root node

(2) splitting.

(3) Branches

(4) leaf Nodes.

Criteria for splitting

(*) Classification trees:-

(1) Gini Impurity :- $G = 1 - \sum_{i=1}^{n} p_i^2$

p_i is proportion of data points in class i

(2) Entropy (Information gain)

$$H(S) = - \sum_{i=1}^{n} p_i \log_2 (p_i)$$

(*) Regression trees:-

(1) ~~MSE~~ Mean Squared Error.

(4) Random Forest:

(1) Bootstrap Sampling (Bagging).
(2) Random Feature selection.
(3) Growing Multiple trees.
(4) Aggregating predictions.

---

(5) PCA:-

(1) Standardize the data.
(2) Compute covariance matrix
(3) Compute eigen values and eigen vectors.
(4) Select principal components.
(5) Transform the data.

---

(6) t-SNE (t-Distributed stochastic Neighbour Embedding) :-

(1) High dimensional similarity (uses gaussian distribution)

(2) Low dimensional similarity. (uses students t-distribution)

(3) Minimize the difference :-
cost function is Kullback-Leibler (KL) divergence.

Hyperparameter :- perplexity.

---

(7) LDA (Linear Discriminant Analysis) :-

(1) Compute mean for each class.

(2) Computer scatter Matrices :-

* Within-class scatter matrix.
* Between-class scatter matrix.

(3) Compute linear discriminants.

(4) Project the data.

Limitations

(1) Assumes linearity

(2) Gaussian Assumption

(3) Sensitive to outlier.

(4) Multicollinearity Issues.

Limitations of PCA :-

(1) loss of interpretibility.

(2) Assumes linearity.

(3) Sensitive to scaling.

(4) Sensitive to Outliers.

Limitations of t-SNE :-

(1) Non-Deterministic

(2) Dont preserve global structure.

---

⑧. Support Vector Machines :-

(1) Hyperplane.

- Margin.
- Support vectors.

(2) Maximize Margin or Minimize Margin Error.

* Kernel trick:

* Soft margin & Hard margin SVM.

Limitations of SVM :-

(1) Computation Intensive.

(2) Difficult to interpret.

(3) Choosing the right kernel.

(4) Not suitable for large datasets.

(9) K-Means Clustering:-

(1) Select the no. of clusters.

(2) Initialize centroids.

(3). Assign data points to nearest centroid. (using Euclidean distance)

(4). Update the centroids.

(5). Repeat until convergence.

* Un Supervised learning.

* K-Means aim to minimize WCSS (within-cluster sum of scores).

$$WCSS = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

* Choosing the value of K (ie no. of clusters).

(1) Elbow Method.

(2) Silhouette score.

Limitations of k-means:-

(1) Need to specify k.

(2) Sensitive to initialization.

(3) Assumes spherical clusters

(4). Sensitive to outliers.

KNN (K- Nearest Neighbours) 10.

(1) Select the No. of Neighbours (k).

(2) Calculate distance. (Euclidean distance).
or
- Manhattan distance
- Minkowski distance.
- Cosine similarity.

(3) Identify k Nearest neighbours.

(4). Predict the output.

Limitations of kNN:-

(1) Computationally expensive.

(2) Sensitive to irrelevant features.

(3) Curse of dimensionality

(4) Imbalanced data.

---

(11). MARS (Multivariate adaptive regression splines).

(1). Piecewise Linear Functions (Basis Functions)

* MARS breaks input data into segments and fits linear models. (called splines) within each segment.

* These segments are connected at knots. (points at which behaviour of model changes).

* . Basis Functions (BF's).

3. Model Construction in two phases:-

- Forward pass.
- backward pass.

4. Interaction terms.

Limitations of MARS:-

(1) Computationally Intensive.
(2) Interpretability
(3) Sensitive to knots.

---

(12) Gradient Descent:- (*)

(1) Objective Function:-
(Minimize loss Function)

(2) Gradient calculation.

(3) Update the parameters.

Types of gradient descent:-

(1) Batch gradient descent.

(2) Stochastic gradient descent.

(3) Mini-Batch gradient descent.

Gradient descent with momentum:-

Momentum helps the algorithm accelerate and smooth out the convergence by adding a fraction of previous update to current update.

Variants:-

(1) **Adam (Adaptive Moment Estimation):-**

Combines momentum with adaptive learning rates.

(2) **RMSProp:-**

Adapts learning rate based on recent gradients.

(3) <u>Adagrad</u>:-
Adapts learning rates for each parameter individually.

---

(13) <u>AdaBoost (Adaptive Boosting)</u>:-

(1) Initialize weights. (assign equal weights).

(2). Train a weak learner.

(3) Evaluate the weak learner.

(4) Update weights:
Increase weight of misclassified points. so that the next learner focuses more on these hard to predict points.

(5) Build a new learner.

(6). Final prediction:
Combine the prediction of all weak learners using weighted majority vote or weighted average.

Limitations of AdaBoost:-

(1) Sensitive to outliers.

(2). Requires careful tuning.

(3) Not scalable.

---

(14) Gradient Boosting:-

1. Initialize a simple model (eg: mean / logodds)
2. Compute Residuals (Errors).
3. Fit a new weak model to the residuals.
4. Update the prediction.
   - Add new tree's predictions to the previous predictions to improve the model.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_t(x_i).$$

$f_t(x_i)$ is prediction from new tree.

η is learning rate.

5. <u>Repeat until convergence</u>:-

Continue adding trees until a stopping criterion is met.

<u>Limitations of gradient Boosting</u>:-

(1) Computationally expensive.

(2) Sensitive to overfitting

(3). Requires parameter tuning.

<u>Variants of Gradient Boosting</u>:-

(1) <u>XGBoost</u> (Extreme gradient Boosting):-

A fast and optimized implementation of gradient Boosting with features like regularization and handling missing data

2. <u>Light GBM</u>:-

Designed for Large Datasets, with faster training times using histogram based approach.

3. CatBoost:-

Gradient boosting library that handles categorical features effectively.

---

(15) XGBoost (Extreme gradient Boosting):-

1. Initialize a Base model (often mean or logodds)
2. Compute Residuals.
3. Fit a new tree on Residuals.
4. Update the model.

$$\hat{y}_i^{(t)} = y_i^{(t-1)} + \eta \cdot f_t(x_i).$$

5. Regularization.
   - applies L1 and L2 regularization.
6. Repeat.
   - Continue building new trees until the stopping criterion is met.

Hyperparameters:-

(1) n_estimators :- No. of boosting rounds (trees).

(2) learning rate :- stepsize for each update.

(3). max_depth:

(4) subsample :- fraction of training data to use for each tree.

(5.) colsample_bytree.

(6). lambda. (L1 and L2 regularization).

---

(16) Neural networks:-

(1) weights and bias. (w and b)

(2) Forward propagation. ($z = \sum_{i=1}^{n} w_i x + b$).

(3). Activation Function.

(4).
- Sigmoid
- ReLU (Rectified linear unit)

- Softmax.

4. Loss function:-

- MSE
- Cross-Entropy.

5. Backward propagation (Backpropagation):-

- adjust weights using gradient descent. to minimize loss function.

6. Optimized algorithm:-

- gradient descent or advanced optimizers like Adam adjust the weights to minimize the loss function.

Hyperparameters in neural networks:-

(1) Learning rate.

(2) Batch size:- no. of samples used to update weights in one iteration.

3. Epochs:- no. of times the network preprocess the entire dataset during training.

4. No. of layers and neurons:-

5. Dropout:- A regularalization technique where neurons are randomly dropped during training to prevent overfitting.

Types of Neural networks:-

① Feed forward Neural network (FNN):-

- Data flows in one direction from input layer to output layer.
- Ex:- Basic classification and regression tasks.

2. Convolutional Neural network (CNN):-

- Specialized for image processing and recognition tasks.
- Uses convolutional layers to extract spatial features from Images.

3. Recurrent neural networks: (RNN):-

- Suitable for sequential data like time series or text.
- Maintains internal state to capture dependencies between previous and current inputs.

4. Long-short-term memory (LSTM):-

- An advanced type of RNN that addresses. the vanishing or exploding gradient problem, making it effective for long sequences.

5. Generative Adversial Networks :- (GAN)

- Consists of a generator and a discriminator that compete, leading to improved inputs. (eg, image generation).

6. Autoencoder :-

A type of neural network used for dimensionality reduction and anamoly detection. by learning a compressed representation of input.