MACHINE LEARNING ALGORITHMS

K-MEANS CLUSTERING, PRINCIPAL COMPONENT ANALYSIS,LSTM,GRADIENT BOOSTING(STOCHASTIC, ADABOOST AND HISTOGRAM-BASED)

K MEANS CLUSTERING

DEFINITION:

- K Means is an unsupervised machine learning algorithm that is used for clustering.
- It clusters the data into K groups , where K is the number of clusters.
- It aims to minimize the distance between the data points and its cluster centroids

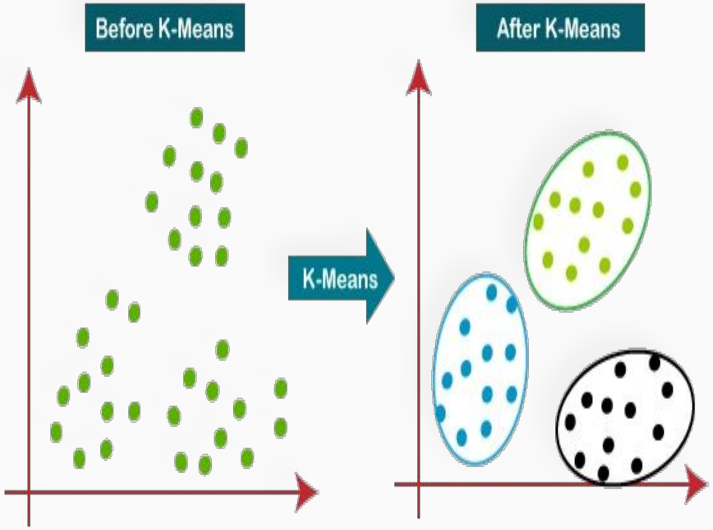
GOAL: The point in a cluster should be more similar to other points in the same cluster then to points in other clusters.

CORE PRINCIPLES:

- **Similarity drives grouping:** Data points are assigned to nearest cluster centers.
- **Iterative refinement:** Clusters are re-assigned iteratively until the cluster assignments does not change or certain criteria are met.
- **Distance metrics:** Euclidean,Manhattan are most widely used.

KEY PROPERTIES:

- **Unsupervised:**Works with unlabelled data
- **Centroid-based:** Each cluster is represented by the mean of its points.
- **Hard clustering:** Each data point belongs to exactly one cluster.
- **Scalable:**Works well on large datasets.



K MEANS CLUSTERING

WORKFLOW:

- **Choose K**: Carefully choosing the number of clusters is a must.Use methods like Elbow or Silhouette, but also consider stability checks,run K-means multiple times with the same K and see if clusters are consistent.
- **Initialize Centroids:** Data points as assigned as cluster centroids randomly which might also lead to poor results.Domain knowledge can be used.
- **Assign Points:** Each point is assigned to the nearest centroid calculated by distance metrics.
- **Update Centroids:** Compute the mean of all points in a cluster and update centroid location.
- **Repeat:** Re assign the points to the new cluster centers and update the clusters until convergence.

CHOOSING K:

- **Elbow Method**: Look for the point where adding more clusters gives diminishing returns in reducing inertia.
- **Silhouette Score**: Measures how well points fit within their clusters.
- **Domain knowledge**:Sometimes K is chosen based on the context of the problem

EXAMPLE:K-means is like grouping customers into “big spenders” and “occasional shoppers” based on their spending and visit frequency .

K MEANS CLUSTERING

PROS AND CONS:

- **PROS:**
  - Simple and easy to implement.
  - Works efficiently on large datasets.
  - Fast convergence compared to other clustering methods.
  - Provides clear, interpretable clusters.

- **CONS:**
  - Requires pre-specifying K.
  - Sensitive to initialization (bad starting points may give poor results).
  - Struggles with non-spherical clusters or varying cluster sizes.
  - Affected by outliers.

HYPERPARAMETERS:

- **K (Number of Clusters)** – Number of groups the data will be divided into.
- **Initialization method** – How starting centroids are chosen .
- **Max iterations** – Maximum steps allowed for updating centroids.
- **Tolerance** – Minimum centroid shift required to continue iterations.

K MEANS CLUSTERING

EVALUATION METRICS:

- **INERTIA:**
    - Inertia is the sum of squared distances between each point and its assigned cluster centroid.
    - Lower Inertia means points are tightly clustered together
    - Inertia always decreases as K increases, so it's used with the Elbow Method to find a balance.

- **SILHOUETTE SCORE**:
    - Measures how similar a point is to its own cluster compared to other clusters.
    - Score ranges from -1 to 1.
    - Closer to 1 : well clustered; Closer to 0: on the boundary; Closer to -1: belongs to the wrong cluster

- **DAVIES–BOULDIN INDEX**:
    - Measures average similarity between each cluster and its most similar cluster.
    - Lower index suggests that clusters are more compact .
    - Higher index suggests that clusters are overlapping or scattered.

COST FUNCTION:

The K-Means cost function is the **Within-Cluster Sum of Squares (WCSS)**, which measures the squared distance of each point from its cluster centroid. It minimizes intra-cluster variance by adjusting centroids until the total distance is as small as possible.

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

USE CASES:

CUSTOMER SEGMENTATION

- Groups customers with similar purchase behavior into clusters which is useful for targeted marketing.
- Simple, scalable, and works well with large customer datasets.

DOCUMENT CLUSTERING

- Groups articles/documents by similarity when represented as vectors ,eg. embeddings.
- Helps in organizing large collections of text for search, categorization, or topic modeling.

ANOMALY DETECTION

- Outliers don't fit well into any cluster ,so it's easy to identify unusual data points.
- Useful in fraud detection, error detection, or rare event identification.

PRINCIPAL COMPONENT ANALYSIS

DEFINITION:

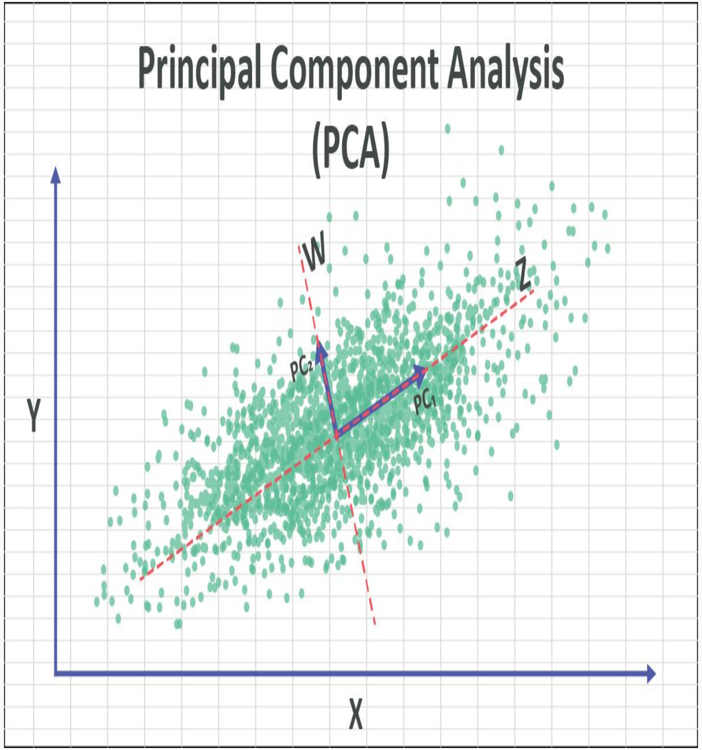
- PCA is a dimensionality reduction method that converts large high-dimensional data into fewer uncorrelated variables called principal components.
- These components keep most of the important variance in the data while using fewer dimensions.

CORE PRINCIPLES:

- **Variance Maximization**: PCA chooses directions (principal components) where the data shows the most variation.
- **Orthogonality**: Components are uncorrelated and orthogonal to each other
- **Linear Transformation**: PCA projects the original data into a new coordinate system formed by these components.

KEY PROPERTIES:

- **Unsupervised**: Does not use labels, works purely on input features.
- **Non-parametric**: It doesn't assume any specific data distribution.
- **Order of Components**: The 1st principal component captures the most variance, the 2nd captures the next, and so on.



PRINCIPAL COMPONENT ANALYSIS

WORKFLOW:

- **Standardize the Data**: Mean = 0, variance = 1 (important when features have different scales).
- **Compute Covariance Matrix**: Measure relationships between features.
- **Eigen Decomposition**:Find eigenvalues & eigenvectors of covariance matrix.
  - Eigenvectors : directions of principal components.
  - Eigenvalues : amount of variance captured.
- **Sort Components**:Order by descending eigenvalues.
- **Select Top k Components**: Choose based on explained variance ratio.
- **Project Data**:Transform original data into new subspace with reduced dimension

COST FUNCTION:

The PCA objective function maximizes the variance captured by the selected components.
Equivalently, it minimizes the **reconstruction error** (difference between original data and projection onto principal components).

$$J = \|X - X_k\|_F^2$$

EXAMPLE: In finance,PCA is like combining many stock indicators into a few main “market trends” that explain most of the ups and downs.

PRINCIPAL COMPONENT ANALYSIS

PROS AND CONS:

- **PROS:**
  - Reduces dimensionality, speeding up training and visualization.
  - Removes multicollinearity (correlated features).
  - Captures most important patterns in fewer features.

- **CONS:**
  - Harder interpretability :transformed features are combinations of original ones.
  - Linear method :struggles if data relationships are non-linear.
  - Sensitive to feature scaling.

HYPERPARAMETERS:

- **Number of components(n_components)** : how many principal components (dimensions) to keep.

- **Solver method(svd_solver)** :the algorithm used for decomposition (e.g., SVD, randomized, eigen).

- **Whitening(whiten)** :option to normalize variance across components for decorrelated features.

PRINCIPAL COMPONENT ANALYSIS

EVALUATION METRICS:

- **EXPLAINED VARIANCE RATIO:**
  - Shows the percentage of total variance captured by the selected principal components.
  - Helps decide the optimal number of components to retain while preserving most information.

- **RECONSTRUCTION ERROR**:
  - Measures the difference between original data and reconstructed data from reduced components.
  - Lower error indicates less information loss and better dimensionality reduction quality.

- **CUMULATIVE EXPLAINED VARIANCE:**
  - It is the running total of variance explained as you add more principal components.
  - It helps choose the smallest number of components that together capture a desired amount of variance

USE CASES:

IMAGE COMPRESSION

- Reduces high-dimensional pixel data while retaining key structures.
- Captures most visual information with fewer components.

NOISE REDUCTION

- Keeps major variance directions (signal).
- Discards minor components often dominated by noise.

PREPROCESSING FOR ML

- Simplifies models by reducing input features.
- Helps avoid overfitting in high-dimensional spaces.

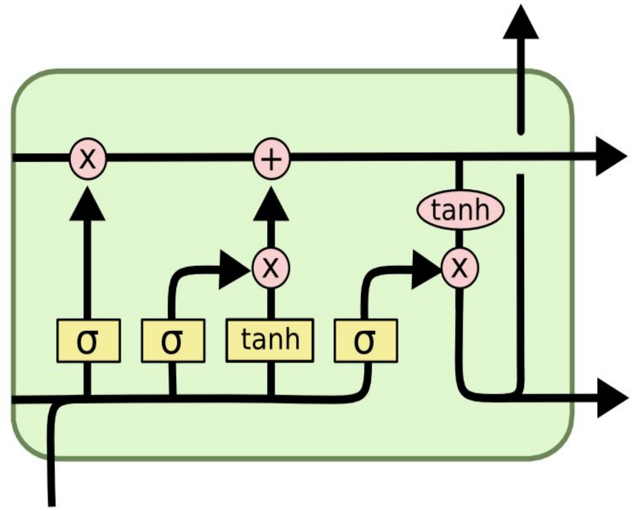
LONG SHORT-TERM MEMORY (LSTM)

DEFINITION:

- LSTM is a type of Recurrent Neural Network (RNN) designed to learn long-term dependencies in sequential data.
- Unlike normal RNNs, it solves the vanishing/exploding gradient problem using a special memory cell structure with gates.

CORE PRINCIPLES:

- **Sequential modeling**: Handles data where order matters (time series, text, speech).
- **Memory cell**: Stores information across time steps.
- **Gates**: Decide what to remember, update or forget.
- **Backpropagation Through Time (BPTT)**: Used for training with gradients.



KEY PROPERTIES:

- Captures long-range dependencies better than normal RNNs.
- Flexible for variable-length sequences.
- Handles noisy or incomplete sequential data.
- Can be stacked in layers for deeper representations.

LSTM

WORKFLOW:

- **Input Sequence**: Feed sequential data (e.g., words, stock prices).
- **Forget Gate**: Decides which information to discard.
- **Input Gate**: Decides which new information to add.
- **Cell State Update**: Maintains memory across time steps.
- **Output Gate**: Decides what information to pass to the next layer/time step.
- **Prediction Layer**: Uses final hidden state for classification/regression.

COST FUNCTION:

- LSTMs do not have a specific cost function , but it uses task specific cost functions
  - Cross-Entropy Loss : for sequence classification.
  - Mean Squared Error (MSE) : for sequence regression.
- Training is done via Backpropagation Through Time (BPTT), where gradients flow across many time steps.

KEY POINTS:

- Activation Functions:
  - Gates use Sigmoid (σ) to output values between 0 and 1 (how much to forget/remember).
  - Memory update uses Tanh to squash values between -1 and 1.
- Regularization: Dropout and gradient clipping are commonly used to prevent overfitting/exploding gradients.
- Optimization: Usually trained with Adam or RMSprop optimizers for stability on sequential data.

LSTM

ACTIVATION FUNCTIONS:

- **Sigmoid (σ):**
    - Outputs values between 0 and 1, used in gates (forget, input, output) to decide “how much to keep.”
    - Provides a probabilistic interpretation of retaining or discarding information.

- **Tanh (tanh):**
    - Outputs values between -1 and 1, used to regulate the cell state (candidate memory and hidden state).
    - Helps normalize values inside the LSTM cell, keeping updates stable.

HYPERPARAMETERS:

- **hidden_size** : Number of units in each LSTM cell
- **num_layers**: Number of stacked LSTM layers
- **dropout:** Fraction of neurons randomly dropped during training
- **learning_rate**: Step size for optimization
- **sequence_length**: Number of time steps fed at once
- **batch_size**: Number of sequences processed per update
- **optimizer**: Method to update weights

LSTM

EVALUATION METRICS:

- **For Sequence Classification**: Accuracy, Precision, Recall, F1, Confusion Matrix.

- **For Sequence Regression (time series)**: MSE, RMSE, MAE, R^2.

- **Special NLP metrics**: BLEU (for translation), Perplexity (for language models).

PROS AND CONS:

- **PROS:**
  - Solves vanishing gradient problem.
  - Good at capturing long-term dependencies.
  - Effective for sequential tasks (NLP, time series).

- **CONS:**
  - Computationally heavy (more parameters than RNNs/GRUs).
  - Slower training.
  - Sometimes too complex: simpler GRUs can perform equally well.

USE CASES:

SPEECH RECOGNITION

- Captures temporal dependencies in sound waves.
- Handles varying-length inputs.
- Remembers patterns in phonemes over time, improving recognition of words in continuous speech.

STOCK PRICE PREDICTION

- Remembers long-term market trends.
- Learns temporal patterns in noisy data.
- Captures sequential dependencies in past price movements to forecast future values.

TEXT GENERATION

- Learns long-range dependencies in language.
- Generates coherent sequences word by word.
- Uses hidden states to keep track of prior context, enabling generation of coherent word sequences.

STOCHASTIC GRADIENT BOOSTING

DEFINITION:

- Stochastic Gradient Boosting (SGB) is an ensemble learning method that builds multiple decision trees in sequence. Unlike standard Gradient Boosting, it adds randomness by:
  - Row subsampling :stochastic sampling of training data
  - Feature subsampling: choosing a subset of features at each split
- This reduces overfitting and improves generalization.

KEY CHARACTERISTICS:

- **Sequential Training** – Trees are built one after another, each correcting errors of the previous.
- **Randomization** – Uses sampling of data/features, which makes it more robust.
- **Learning Rate** – Controls contribution of each tree; small values → slower but more accurate.

COST FUNCTION:

- Same as Gradient Boosting:
  - Regression: Mean Squared Error
  - Classification: Log Loss

EVALUATION METRICS:

- Same as Gradient Boosting:
  - Regression:MSE,MAE,RMSE,R^2
  - Classification: Accuracy,Precision,Recall,F1 score,Confusion matrix

USE CASES:

CUSTOMER CHURN PREDICTION

- Learns complex non-linear patterns in customer behavior.
- Subsampling helps avoid overfitting in highly imbalanced churn datasets.

FRAUD DETECTION (FINANCE/BANKING)

- Detects rare fraudulent transactions by focusing on misclassified cases.
- Randomization increases robustness against noisy/high-dimensional data.

MEDICAL RISK PREDICTION

- Models patient data (age, lab results, history) to predict disease likelihood.
- Handles missing values and reduces overfitting in small sample medical datasets.

ADABOOST

DEFINITION:

- AdaBoost (Adaptive Boosting) is an ensemble learning algorithm that combines multiple weak learners (usually decision stumps) to form a strong classifier.
- It works by assigning weights to each training instance:
  - Misclassified samples get higher weights.
  - Correctly classified samples get lower weights.
- Each new weak learner focuses more on the hard-to-classify examples.

KEY CHARACTERISTICS:

- **Sequential Training** :Learners are added one by one, each correcting mistakes of the previous.
- **Weighted Voting** :Final prediction is a weighted majority vote of all weak learners.
- **Sensitivity to Noise** :Can overfit if too many noisy points are present.

COST FUNCTION:

- AdaBoost minimizes the exponential loss function:

$$L = \sum_{i=1}^{n} e^{-y_i f(x_i)}$$

EVALUATION METRICS:

- Same as Gradient Boosting:
  - Regression:MSE,MAE,RMSE,R^2
  - Classification: Accuracy,Precision,Recall,F1 score,Confusion matrix

USE CASES:

SPAM EMAIL DETECTION

- Learns from weak rules (e.g., presence of certain keywords) to classify spam vs. not spam.
- Assigns higher weight to tricky emails that were misclassified earlier.

CUSTOMER CREDIT SCORING

- Combines multiple weak decision stumps to predict likelihood of default.
- Focuses on difficult-to-classify customers with borderline credit histories.

FACE DETECTION

- Used in early real-time face detection systems (e.g., Viola–Jones algorithm).
- Efficiently boosts weak classifiers (like Haar features) into strong detectors.

HISTOGRAM BASED GRADIENT BOOSTING

DEFINITION:

- Histogram-Based Gradient Boosting (HGB) is a variant of Gradient Boosting that speeds up training by:
  - Binning continuous features into histograms (fixed number of bins).
  - Using these histograms to find the best splits instead of scanning all feature values.
- This reduces memory usage and training time, especially for large datasets.

KEY CHARACTERISTICS:

- **Histogram Approximation** :Features are bucketed into bins, reducing computation.
- **Scalability** : Works efficiently on large datasets with high-dimensional features.
- **Regularization** :Supports shrinkage, subsampling, and depth constraints to prevent overfitting.

COST FUNCTION:

- Same as Gradient Boosting:
  - Regression: Mean Squared Error
  - Classification: Log Loss

EVALUATION METRICS:

- Same as Gradient Boosting:
  - Regression:MSE,MAE,RMSE,R^2
  - Classification: Accuracy,Precision,Recall,F1 score,Confusion matrix

USE CASES:

Click-Through Rate Prediction

- Handles massive datasets with millions of rows efficiently.
- Histogram binning speeds up training while maintaining accuracy.

Fraud Detection

- Can process high-cardinality categorical + numeric features quickly.
- Robust against noisy features due to binning.

Search Ranking & Personalization

- Used in ranking models where latency and scalability are critical.
- Works well with sparse, high-dimensional data like text or user logs.