24_May_ML_Assignment4

July 17, 2025

1. What is clustering in machine learning

Clustering is an unsupervised learning technique used to group similar data points into clusters based on features or patterns. The goal is to ensure data points in the same cluster are more similar to each other than to those in other clusters.

2. Explain the difference between supervised and unsupervised clustering

Supervised learning uses labeled data; clustering is not supervised.

Clustering is a form of unsupervised learning, meaning:

There are no predefined labels.

The model identifies natural groupings or structures in the data.

Note: "Supervised clustering" is a misleading term. Sometimes, semi-supervised approaches involve some labels, but traditional clustering is fully unsupervised.

3. What are the key applications of clustering algorithms

Customer segmentation (e.g., in marketing)

Image compression and segmentation

Document or news classification

Anomaly detection

Social network analysis

Recommendation systems

Genomic data analysis

4. Describe the K-means clustering algorithm

K-Means aims to partition n observations into k clusters by minimizing the within-cluster variance.

`Steps:`

Choose the number of clusters k.

Initialize k centroids randomly.

Assign each data point to the nearest centroid (using Euclidean distance).

Recompute centroids by averaging points in each cluster.

Repeat steps 3–4 until convergence (centroids no longer move or a max iteration is reached).

5. What are the main advantages and disadvantages of K-means clustering

`Advantages:`

Simple and easy to implement

Fast and efficient for large datasets

Works well with spherical, equally sized clusters

`Disadvantages:`

Need to specify k in advance

Sensitive to initial centroids (can lead to different results)

Poor performance with non-spherical or varying density clusters

Affected by outliers

6. How does hierarchical clustering work |

`Hierarchical clustering builds a tree of clusters (dendrogram). Two types:`

Agglomerative (bottom-up): Start with each data point as a cluster and merge the closest pair step-by-step.

Divisive (top-down): Start with one large cluster and divide it into smaller ones.

`No need to predefine the number of clusters-cutting the dendrogram at a desired level gives th`

7. What are the different linkage criteria used in hierarchical clustering?

`Linkage defines the distance between clusters:`

Single linkage: Minimum distance between points in the two clusters

Complete linkage: Maximum distance between points

Average linkage: Mean distance between all points in the two clusters

Ward's method: Minimizes total within-cluster variance (often preferred)

8. Explain the concept of DBSCAN clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that forms clusters based on density of points.

`Concepts:`

Core points: Have enough neighboring points within a radius

Border points: Near core points but don't have enough neighbors

Noise points: Do not belong to any cluster

`Clusters are formed by connecting density-reachable points.`

9. What are the parameters involved in DBSCAN clustering

`eps: Radius of neighborhood to search around a point`

`min_samples: Minimum number of points required to form a dense region`

These control how clusters are defined and how noise is handled.

10. Describe the process of evaluating clustering algorithms

Since clustering is unsupervised, evaluation doesn't rely on ground truth labels (though some datasets have labels for comparison).

`Internal Evaluation Metrics:`

Silhouette Score: Measures how similar a point is to its cluster vs. others.

Davies-Bouldin Index: Lower is better (compact & well-separated clusters).

Inertia (for K-Means): Sum of squared distances to nearest cluster center.

`External Metrics (if true labels are available):`

Adjusted Rand Index (ARI)

Normalized Mutual Information (NMI)

Fowlkes–Mallows Index

`Visual Tools:`

PCA or t-SNE for 2D visualization of high-dimensional data.

11. What is the silhouette score, and how is it calculated

The Silhouette Score measures how well a point fits within its cluster compared to other clusters.

`For a single point:`

a= average distance to all other points in the same cluster

b= average distance to all points in the nearest cluster

`Silhouette score for that point:`

s=b−a/max(a,b)

`Ranges from -1 to 1:`

Close to 1 → well clustered

Close to 0 → overlapping clusters

Negative → wrongly clustered

12. Discuss the challenges of clustering high-dimensional data

Curse of dimensionality: Distances become less meaningful as dimensions increase.

Sparsity: Data becomes sparse, reducing density-based algorithms' effectiveness.

Computational complexity: More features = more computation.

Visualization difficulty: Hard to interpret or evaluate clusters visually.

Irrelevant features: Can dominate clustering; feature selection or dimensionality reduction (PCA, t-SNE, UMAP) becomes essential.

13. Explain the concept of density-based clustering

Density-based clustering groups data based on regions of high point density separated by regions of low density.

`Key features:`

Can find arbitrarily shaped clusters

Can automatically detect noise/outliers

Popular algorithm: DBSCAN

`Idea: A cluster is formed if a core point has enough nearby points (within eps), and other poi`

14. How does Gaussian Mixture Model (GMM) clustering differ from K-means

Feature	K-Means	GMM Clustering
Cluster shape	Circular/Spherical	Elliptical / Arbitrary shape
Membership	Hard (1 point $\rightarrow$ 1 cluster)	Soft (probabilistic)
Based on	Euclidean distance	Probability density estimation
Model	Non-probabilistic	Probabilistic (Gaussian distributions)
Better for	Equal-sized clusters	Overlapping clusters

15. What are the limitations of traditional clustering algorithms

Require manual tuning of parameters (e.g., k in K-means)

Sensitive to outliers and scaling

Assume specific cluster shapes (e.g., spherical in K-means)

Poor performance on imbalanced or non-convex clusters

Don't work well with categorical data (unless encoded)

Limited by distance metrics in high dimensions

16. Discuss the applications of spectral clustering

Spectral clustering uses eigenvalues of the similarity matrix of the data to reduce dimensions before clustering.

`Applications:`

Image segmentation

Social network analysis

Speech and signal processing

Community detection

Non-convex cluster separation

Document and text clustering

17. Explain the concept of affinity propagation

Unlike K-means, Affinity Propagation does not require pre-defining the number of clusters.

`Key idea:`

Each point sends and receives messages to decide which point is best suited as a cluster center (exemplar).

Based on similarity matrix and message passing between points.

Automatically finds the number of clusters based on a "preference" value.

`Pros:`

Finds exemplars

Works well for non-spherical clusters

18. How do you handle categorical variables in clustering

`Options to handle categorical data:`

One-Hot Encoding: Converts categorical values to binary vectors (works with algorithms like K-means but increases dimensionality).

Frequency/Label Encoding: For ordinal data.

`Distance-based methods:`

Hamming distance for binary/categorical data

Gower distance: Handles mixed numerical and categorical types

`Use clustering algorithms designed for categorical data like:`

K-Modes

K-Prototypes (mixed data)

19. Describe the elbow method for determining the optimal number of clusters

Goal: Find the best number of clusters (k) for K-Means.

`Steps:`

1. Run K-Means for a range of k values (e.g., 1–10).
2. Plot k vs Inertia (sum of squared distances to cluster centers).
3. Look for the "elbow" point where the inertia stops decreasing significantly — that's the ideal k.

20. What are some emerging trends in clustering research

Deep Clustering: Combines neural networks with clustering (e.g., autoencoders + K-means).

Self-Supervised Clustering: Learning representations + clusters without labels.

Graph-based Clustering: Using Graph Neural Networks (GNNs) and community detection.

Clustering on streaming data: Online or real-time clustering (e.g., CluStream).

Explainable Clustering: Making cluster results interpretable and transparent.

Clustering for LLMs and embeddings: E.g., clustering dense text/image embeddings from transformers.

21. What is anomaly detection, and why is it important

Anomaly detection is the task of identifying rare items, events, or patterns that differ significantly from the majority of the data.

`Why important?`

Detect fraud (e.g., credit card fraud)

Identify failures in systems (e.g., machinery, networks)

Monitor cybersecurity threats (e.g., intrusions, phishing)

Ensure data quality by detecting corrupt records

22. Discuss the types of anomalies encountered in anomaly detection

    1. Point anomalies: A single instance is anomalous (e.g., very high transaction).
    2. Contextual anomalies: An anomaly only under specific context (e.g., a high temperature in winter).
    3. Collective anomalies: A group of data points are anomalous together (e.g., a sudden spike in network traffic).

23. Explain the difference between supervised and unsupervised anomaly detection techniques

Feature	Supervised	Unsupervised
Labels available	Yes (normal vs. anomaly)	No
Accuracy	Higher (if enough labels)	Varies (no ground truth)
Algorithms used	Classification models	Clustering, density, distance-based
Common examples	Logistic regression, SVM	Isolation Forest, LOF, DBSCAN

24. Describe the Isolation Forest algorithm for anomaly detection

Isolation Forest works on the idea that anomalies are easier to isolate from the rest of the data.

`How it works:`

Builds random decision trees by splitting data.

Shorter paths to isolate a point $\rightarrow$ likely an anomaly.

Aggregates isolation depths to calculate an anomaly score.

`Efficient for high-dimensional and large datasets.`

25. How does One-Class SVM work in anomaly detection

One-Class SVM tries to learn the boundary of the normal class.

It fits a hypersphere (or hyperplane) around the normal data points.

New data points falling outside this region are considered anomalies.

Good for:

High-dimensional data

When only normal data is available for training

26. Discuss the challenges of anomaly detection in high-dimensional data

Curse of dimensionality: Distances become less meaningful.

Feature redundancy or noise can mask anomalies.

Scalability: Algorithms may be computationally expensive.

Sparse anomalies: Few anomalous instances can make detection hard.

Visualization and interpretability are limited.

Solutions: Use PCA, autoencoders, or feature selection to reduce dimensions.

27. Explain the concept of novelty detection

Novelty detection focuses on detecting new patterns that differ from known data during prediction.

Trained only on normal data, and tested with unknown inputs.

Useful when anomalies are not present in the training set.

Example: A model trained on healthy machine readings detects faults it has never seen before.

28. What are some real-world applications of anomaly detection?

Finance: Credit card fraud detection

Healthcare: Disease outbreak or unusual health metrics

Network security: Intrusion and malware detection

Manufacturing: Predictive maintenance

E-commerce: Fake reviews or pricing errors

Banking: Money laundering and suspicious transactions

IoT systems: Detecting faulty sensors

29. Describe the Local Outlier Factor (LOF) algorithm

LOF detects anomalies based on local density.

Concept:

Compares the density of a point to its neighbors.

If a point is in a low-density region compared to neighbors, it's an outlier.

Key ideas:

k-nearest neighbors (k-NN) used

Score > 1 indicates an anomaly

Works well in datasets with varying density

30. How do you evaluate the performance of an anomaly detection model

`If labels are available:`

Precision, Recall, F1-score (important for imbalanced data)

ROC-AUC / PR-AUC (especially when anomalies are rare)

Confusion matrix (TP, FP, FN, TN)

`If labels are unavailable:`

Use domain knowledge

Visualizations (e.g., t-SNE + outlier highlighting)

Silhouette score or reconstruction error (e.g., autoencoders)

31. Discuss the role of feature engineering in anomaly detection

`Feature engineering plays a crucial role because:`

Relevant features improve anomaly detection accuracy.

Helps highlight patterns and separate normal from anomalous behavior.

Reduces noise and redundancy, especially in high-dimensional data.

Enables temporal, domain-specific, or derived features (e.g., time since last transaction, average value, etc.).

`Examples:`

In network intrusion detection: packet size, frequency, time gaps.

In credit card fraud: time since last transaction, amount deviation.

32. What are the limitations of traditional anomaly detection methods

Assume fixed data distribution (e.g., Gaussian) — not always true.

Poor performance on high-dimensional or noisy data.

Cannot handle concept drift (i.e., when normal behavior changes over time).

Many require manual thresholds or tuning.

Not robust against imbalanced datasets or unknown anomalies.

May not adapt well to real-time streaming scenarios.

33. Explain the concept of ensemble methods in anomaly detection

Ensemble methods combine multiple models to improve anomaly detection accuracy and robustness.

`Types:`

Bagging: Combines multiple detectors trained on different random subsamples (e.g., Isolation Forest).

Boosting: Sequentially trains models to correct previous errors.

Hybrid: Combines statistical + ML models (e.g., LOF + autoencoder).

Voting/Averaging: Aggregates anomaly scores from multiple models.

`Benefits:`

Reduces variance

Increases robustness

Handles diverse types of anomalies better

34. How does autoencoder-based anomaly detection work

Autoencoders are neural networks trained to reconstruct input data.

`Process:`

1. Train autoencoder on normal data only.
2. During inference:

   Normal data → low reconstruction error.

   Anomalies → high reconstruction error (as model has never seen them).
3. Set a threshold on reconstruction loss to detect anomalies.

   `Useful for high-dimensional, complex datasets (e.g., images, time series).`

35. What are some approaches for handling imbalanced data in anomaly detection

Resampling:

`Oversampling anomalies (e.g., SMOTE)`

`Undersampling normal data`

Anomaly score threshold tuning: Adjust to favor recall or precision

Cost-sensitive learning: Penalize false negatives more

Synthetic data generation: Create artificial anomalies

Use anomaly-specific evaluation metrics: e.g., AUC-PR, F1 for rare classes

Ensemble methods: Balance models to handle minority class

36. Describe the concept of semi-supervised anomaly detection

Assumes only normal class is labeled during training.

Learns patterns of normal behavior; anything different is an anomaly.

More realistic in many applications (e.g., cybersecurity, healthcare).

Models: One-Class SVM, autoencoders, semi-supervised deep learning.

`Useful when:`

Labeled anomalies are rare

Manual labeling is expensive

37. Discuss the trade-offs between false positives and false negatives in anomaly detection

`False Positive (FP): Normal instance marked as anomaly`

Can lead to alert fatigue, wasted resources

`False Negative (FN): Anomaly missed as normal`

Can be critical in applications like fraud or disease detection

`Trade-off depends on domain:`

In fraud detection → FN is worse

In system alerts → Too many FP reduces trust

`Balance using:`

Precision-Recall tuning

Cost-based loss functions

ROC/PR curve analysis

38. How do you interpret the results of an anomaly detection model

`Interpretation involves:`

Visualizing outliers using t-SNE, PCA, or UMAP

Ranking by anomaly scores (e.g., top 5% most anomalous)

Comparing results with domain knowledge

Understanding model output (e.g., reconstruction error in autoencoder)

Using feature attribution tools (e.g., SHAP, LIME) for black-box models

39. What are some open research challenges in anomaly detection

Label scarcity: Real anomalies are rare and hard to label

Concept drift: Changes in normal behavior over time

Explainability: Why a point is anomalous

Dynamic & streaming data: Real-time detection is complex

Multi-modal data: Text, image, audio combinations

Data privacy and federated anomaly detection

Adversarial robustness: Attackers may mask anomalies

40. Explain the concept of contextual anomaly detection

Contextual anomalies are data points that are only anomalous within a specific context.

`Examples:`

High heart rate during exercise is normal, but not while sleeping.

30°C temperature is normal in summer, but anomalous in winter.

`Key components:`

Contextual attributes: Time, location, etc.

Behavioral attributes: Values being analyzed

`Models must consider context + behavior simultaneously. Often solved with:`

LSTM (for time series)

Context-aware autoencoders

Rule-based systems

41. What is time series analysis, and what are its key components

Time series analysis involves methods to analyze data points collected or recorded over time (e.g., stock prices, weather, sales).

`Key components:`

Trend: Long-term direction (upward/downward).

Seasonality: Repeating patterns (e.g., daily, weekly, yearly).

Cyclic behavior: Non-fixed, longer-term fluctuations.

Noise (or residual): Random variations not explained by the model.

42. Discuss the difference between univariate and multivariate time series analysis

Aspect	Univariate	Multivariate
Input variables	Only one time-dependent variable	Two or more time-dependent variables
Example	Temperature over time	Temperature + humidity over time
Complexity	Simpler models (ARIMA, ETS)	More complex (VAR, LSTM multivariate)
Use cases	Forecasting a single metric	Capturing inter-variable relationships

43. Describe the process of time series decomposition

Time series decomposition breaks down the original series into separate interpretable components:

1. Trend extraction (e.g., via moving average)
2. Seasonality detection
3. Residual analysis (random error/noise)

   `Helps in understanding the data and building better forecasting models.`

44. What are the main components of a time series decomposition

1. Trend (T): Long-term increase or decrease.
2. Seasonality (S): Repeating short-term cycles.
3. Residual (R): Irregular, random noise.
4. Cycle (optional): Long-term fluctuations that aren't seasonal.

   `Two models:`

Additive: $Y(t)=T(t)+S(t)+R(t)$

Multiplicative: $Y(t)=T(t)\times S(t)\times R(t)$

45. Explain the concept of stationarity in time series data

`A time series is stationary if its:`

Mean, variance, and covariance are constant over time.

`Why important?`

Most forecasting models (like ARIMA) require stationarity for validity.

46. How do you test for stationarity in a time series

    1. Visual inspection: Plot to check for trends or changing variance.
    2. Rolling statistics: Mean & variance over sliding windows.
    3. Statistical tests:

       ADF (Augmented Dickey-Fuller): Null hypothesis = non-stationary

       KPSS (Kwiatkowski-Phillips-Schmidt-Shin): Null = stationary

If not stationary:

Apply differencing, log transforms, or seasonal adjustments.

47. Discuss the autoregressive integrated moving average (ARIMA) model

`ARIMA is a forecasting model combining:`

AR (Autoregression): Uses past values

I (Integrated): Differencing to make data stationary

MA (Moving Average): Uses past forecast errors

`Used for non-seasonal time series prediction.`

48. What are the parameters of the ARIMA model

`ARIMA is defined by three parameters: ARIMA(p, d, q)`

p: Order of AR (lag of past values)

d: Order of differencing (to make data stationary)

q: Order of MA (lag of past forecast errors)

`Example:`

ARIMA(1,1,2) $\rightarrow$ 1 AR term, 1 differencing, 2 MA terms

49. Describe the seasonal autoregressive integrated moving average (SARIMA) model

`SARIMA extends ARIMA by adding seasonality:`

SARIMA(p, d, q)(P, D, Q, s)

p, d, q: ARIMA terms

P, D, Q: Seasonal components

s: Season length (e.g., 12 for monthly data with yearly seasonality)

`Captures both short-term and seasonal patterns in time series.`

50. How do you choose the appropriate lag order in an ARIMA mode
    1. ACF (Autocorrelation Function): Helps select q (MA terms)
    2. PACF (Partial Autocorrelation Function): Helps select p (AR terms)
    3. Differencing (d): Use until series is stationary
    4. Grid search or auto_arima (pmdarima): Automatically finds best (p,d,q)
    5. Information criteria:

       AIC (Akaike Information Criterion)

       BIC (Bayesian Information Criterion)

Lower AIC/BIC = better model.

51. Explain the concept of differencing in time series analysis

`Differencing is a technique used to make a time series stationary by removing trends or seasona`

First-order differencing: yt =yt−yt−1

Second-order differencing: Apply differencing again on the result.

`It helps in stabilizing the mean of the series, which is essential for models like ARIMA.`

52. What is the Box-Jenkins methodology

The Box-Jenkins methodology is a 4-step iterative approach to time series modeling using ARIMA/SARIMA models.

`Steps:`

1. Identification: Make data stationary; identify p, d, q using ACF/PACF.
2. Estimation: Estimate model parameters (using MLE or least squares).
3. Diagnostic checking: Examine residuals for white noise.
4. Forecasting: Use the fitted model to predict future values.

   `This is the foundation for ARIMA-based forecasting.`

53. Discuss the role of ACF and PACF plots in identifying ARIMA parameters

    `ACF (Autocorrelation Function):`

Measures correlation between a time series and its lags.

Helps determine q (MA order).

    `PACF (Partial Autocorrelation Function):`

Removes intermediate correlations.

Helps determine p (AR order).

`Pattern-based decisions:`

AR(p): Sharp cutoff in PACF

MA(q): Sharp cutoff in ACF

ARMA(p,q): Gradual decay in both

54. How do you handle missing values in time series data

`Methods:`

Forward fill / Backward fill (ffill/bfill)

Linear interpolation

Moving average smoothing

Seasonal decomposition and imputation

Model-based imputation (e.g., Kalman filters, ARIMA)

Delete if very few missing points and not critical

`Important to maintain temporal consistency when filling.`

55. Describe the concept of exponential smoothing

Exponential smoothing assigns exponentially decreasing weights to past observations.

`Formula:`

$^{yt= yt-1+(1-)}yt-1$

[0, 1]: smoothing factor

`Types:`

Simple: For data without trend/seasonality

Double: Accounts for trend

Triple (Holt-Winters): Handles trend + seasonality

56. What is the Holt-Winters method, and when is it used?

Holt-Winters is an extension of exponential smoothing for seasonal time series.

`Types:`

Additive: For constant seasonal variation

Multiplicative: For seasonality that scales with level

`Components:`

Level (L)

Trend (T)

Seasonality (S)

`Used when data exhibits both trend and seasonal patterns (e.g., retail sales, energy usage).`

57. Discuss the challenges of forecasting long-term trends in time series data

Unpredictable changes: Market shifts, policy changes, natural events

Accumulated error: Forecast error grows over time

Model overfitting: Too complex models may perform poorly long-term

Concept drift: Underlying pattern may change

Seasonal shifts: Future seasons may not match past patterns

`Mitigation: Use ensemble models, confidence intervals, update models regularly.`

58. Explain the concept of seasonality in time series analysis

Seasonality is a repeating pattern or cycle in data at regular intervals.

`Examples:`

Daily: Website traffic peaks at certain hours

Weekly: Shopping habits on weekends

Yearly: Weather or sales patterns

`Handled using:`

Seasonal decomposition

SARIMA

Holt-Winters

Fourier terms in ML models

59. How do you evaluate the performance of a time series forecasting model

`Common metrics:`

MAE (Mean Absolute Error)

RMSE (Root Mean Squared Error)

MAPE (Mean Absolute Percentage Error) – beware of division by zero

SMAPE (Symmetric MAPE)

R^2 Score

Cross-validation: Time series split (not random split!)

`Visual tools:`

Forecast vs. actual plot

Residual plot to detect bias

60. What are some advanced techniques for time series forecasting?

Facebook Prophet: Robust, interpretable, handles holidays and missing data

LSTM/GRU (Recurrent Neural Networks): Learn long-term dependencies

Temporal Convolutional Networks (TCN)

Transformer models: For multivariate or high-dimensional forecasting (e.g., Informer, TimeGPT)

AutoML models: AutoTS, H2O-AutoML, GluonTS

Hybrid models: Combine statistical + deep learning (e.g., ARIMA + LSTM)