24_May_ML_Assignment1

July 11, 2025

1. Define Artificial Intelligence (AI)

Artificial Intelligence (AI) is the simulation of human intelligence in machines that are programmed to think, learn, and make decisions like humans. It includes capabilities such as reasoning, problem-solving, understanding language, and perception.

2. Explain the differences between Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and Data Science (DS)

Concept	Description
Artificial Intelligence (AI)	A broad field focused on creating machines capable of intelligent behavior.
Machine Learning (ML)	A subset of AI that enables systems to learn from data and improve over time without being explicitly programmed.
Deep Learning (DL)	A subset of ML that uses neural networks with many layers (deep networks) to analyze complex patterns in large datasets.
Data Science (DS)	A multidisciplinary field that uses statistical, mathematical, and computational methods to extract insights from data. It may use AI/ML as tools.

3. How does AI differ from traditional software development?

Aspect	Traditional Software	AI-based Systems
Logic	Explicitly programmed rules	Learns from data
Adaptability	Static behavior	Dynamic and improves over time
Input/Output	Predictable output for known input	Output may vary depending on learned model
Development Process	Rule-based coding	Model training and testing

4. Provide examples of AI, ML, DL, and DS applications.

AI: Chatbots (e.g., Siri, Alexa), autonomous vehicles

ML: Email spam detection, fraud detection

DL: Facial recognition, voice assistants

DS: Business intelligence dashboards, customer behavior analytics

5. Discuss the importance of AI, ML, DL, and DS in today's world

Automation: Reduces human effort in repetitive tasks.

Personalization: Powers personalized recommendations (e.g., Netflix, Amazon).

Healthcare: Enables early diagnosis through image analysis and data-driven treatment.

Finance: Detects fraudulent transactions, improves investment strategies.

Industry: Enhances efficiency, predictive maintenance, and quality control.

6. What is Supervised Learning

Supervised Learning is a type of machine learning where the model is trained on labeled data. The algorithm learns the mapping between input features and known output labels to make future predictions.

7. Provide examples of Supervised Learning algorithms

Linear Regression

Logistic Regression

Decision Trees

Random Forest

Support Vector Machines (SVM)

k-Nearest Neighbors (k-NN)

Neural Networks

8. Explain the process of Supervised Learning

1. Collect Data: Gather labeled training data.
2. Preprocess Data: Clean and prepare data for the model.
3. Split Data: Divide data into training and testing sets.
4. Train Model: Use training data to teach the algorithm.
5. Test Model: Evaluate performance using test data.
6. Deploy Model: Use the trained model for predictions in real-world scenarios.

9. What are the characteristics of Unsupervised Learning?

No labeled output data.

The model tries to find hidden patterns or structures.

Common for clustering, dimensionality reduction, and anomaly detection.

It discovers relationships in data without pre-defined labels.

10. Give examples of Unsupervised Learning algorithms

k-Means Clustering

Hierarchical Clustering

Principal Component Analysis (PCA)

DBSCAN (Density-Based Spatial Clustering)

Autoencoders

11. Describe Semi-Supervised Learning and its significance

Semi-Supervised Learning is a type of machine learning that uses a small amount of labeled data along with a large amount of unlabeled data. It falls between Supervised and Unsupervised learning.

Significance:

Reduces the cost and effort of labeling large datasets.

Often achieves higher accuracy than unsupervised learning when labeled data is scarce.

Useful in domains like speech recognition, medical imaging, and natural language processing.

12. Explain Reinforcement Learning and its applications

Reinforcement Learning (RL) is a learning paradigm where an agent learns to make decisions by interacting with an environment. It receives rewards or penalties based on its actions and aims to maximize cumulative reward.

Applications:

Game AI (e.g., AlphaGo)

Robotics (e.g., robot navigation)

Self-driving cars

Dynamic pricing

Personalized recommendations

13. How does Reinforcement Learning differ from Supervised and Unsupervised Learning?

Feature	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Data	Labeled	Unlabeled	Based on interaction (states, actions, rewards)
Goal	Predict output	Find hidden structure	Learn optimal strategy/policy
Feedback	Correct answers provided	No feedback	Delayed feedback (reward signals)
Learning	From examples	From patterns	From exploration and reward

14. What is the purpose of the Train-Test-Validation split in machine learning?

The Train-Test-Validation split is used to:

Train the model (training set)

Validate it during tuning (validation set)

Test final performance (testing set)

It helps prevent overfitting, ensures generalization, and improves model reliability.

15. Explain the significance of the training set

The training set is the subset of data used to fit the model. It is crucial because:

The model learns patterns, relationships, and weights from it.

It directly influences how well the model can predict outcomes

16. How do you determine the size of the training, testing, and validation sets?

Commonly used split ratios:

70% Train / 15% Validation / 15% Test

80% Train / 10% Validation / 10% Test

Factors to consider:

Size of dataset

Model complexity

Need for reliable evaluation

Balance between training quality and evaluation robustness

17. What are the consequences of improper Train-Test-Validation splits?

Overfitting: Model performs well on training data but poorly on unseen data.

Underfitting: Not enough training data leads to poor learning.

Biased Evaluation: Using test data during training inflates performance metrics.

Poor Generalization: Model may not work well in real-world applications.

18. Discuss the trade-offs in selecting appropriate split ratios

More training data: Better learning but risk of weak evaluation.

More test/validation data: Better evaluation but may reduce learning capacity.

Small datasets: May need techniques like cross-validation instead of fixed splits.

Goal: Balance between learning accuracy and evaluation reliability.

19. Define model performance in machine learning

Model performance refers to how well a model makes predictions or classifies data. It is assessed based on its ability to generalize to unseen data.

20. How do you measure the performance of a machine learning model?

Depends on task:

1. Classification:

   Accuracy

   Precision, Recall, F1-Score

   Confusion Matrix

   ROC-AUC

2. Regression:

   Mean Squared Error (MSE)

   Root Mean Squared Error (RMSE)

   Mean Absolute Error (MAE)

   R^2 Score

3. General:

   Cross-validation scores

   Learning curves

   Overfitting/underfitting analysis

21. What is overfitting and why is it problematic?

Overfitting occurs when a machine learning model learns the training data too well, including its noise and outliers, and fails to generalize to new, unseen data.

Why it's problematic:

The model has high accuracy on training data but poor performance on test/real-world data.

It leads to poor generalization, which defeats the purpose of predictive modeling

22. Provide techniques to address overfitting

Simplify the model (reduce complexity)

Use more training data (if available)

Cross-validation (e.g., k-fold)

Regularization (e.g., L1/L2 penalties)

Early stopping (especially in neural networks)

Pruning (in decision trees)

Dropout (for neural networks)

Reduce training time or epoch count

23. Explain underfitting and its implications

Underfitting occurs when a model is too simple to capture the underlying patterns in the data.

Implications:

```
Poor performance on both training and test sets
```

```
Model fails to learn meaningful patterns
```

```
Results in high bias
```

24. How can you prevent underfitting in machine learning models?

Increase model complexity (e.g., deeper trees, more layers)

Train longer (more epochs or iterations)

Feature engineering (include more relevant features)

Reduce regularization (avoid over-penalizing complexity)

Tune hyperparameters (e.g., learning rate, depth, number of neurons)

25. Discuss the balance between bias and variance in model performance

This is known as the bias-variance trade-off:

Bias: Error due to overly simplistic models. High bias → Underfitting.

Variance: Error due to too much complexity. High variance → Overfitting.

Goal: Find a balance where both bias and variance are minimized:

Too simple → high bias, low variance

Too complex → low bias, high variance

Ideal → low bias and low variance → optimal generalization

26. What are the common techniques to handle missing data?

Remove records with missing values (if few and not critical)

Imputation:

Mean/Median/Mode substitution

Interpolation

Predictive models (e.g., KNN, regression)

Flag missing values as a separate category (for categorical data)

Use algorithms that handle missing data internally (e.g., XGBoost)

27. Explain the implications of ignoring missing data

Biased analysis and incorrect conclusions

Reduced accuracy

Loss of valuable information

Data inconsistency and model errors

Smaller dataset if rows/columns are dropped, reducing learning power

28. Discuss the pros and cons of imputation methods.

Method	Pros	Cons
Mean/Median/Mode	Simple, fast	Can distort data distribution; ignores feature relationships
KNN Imputation	Considers data similarity	Computationally expensive; may not scale
Regression Imputation	Captures relationships	Assumes linearity; can add bias
Multiple Imputation	Robust, statistically sound	Complex to implement; time-consuming
Dropping Missing Values	Easy to do	Risk of information loss; reduces dataset size

29. How does missing data affect model performance?

Reduces model accuracy

Increases uncertainty

Can lead to biased or invalid predictions if not properly handled

May cause models to fail to train if key features are missing

30. Define imbalanced data in the context of machine learning

Imbalanced data refers to datasets where classes are not represented equally, typically in classification problems (e.g., 95% negative, 5% positive).

Why it matters:

Models tend to favor the majority class

Accuracy becomes misleading

Critical in domains like fraud detection, disease diagnosis, etc.

31. Discuss the challenges posed by imbalanced data

Imbalanced data occurs when one class (usually the negative class) vastly outnumbers the other (positive class).

Challenges:

Biased models: Models may favor the majority class and ignore the minority.

Misleading accuracy: A model may show high accuracy while failing to detect minority class.

Poor recall/precision: Especially harmful in critical tasks like fraud detection or disease prediction.

Inadequate training: The model doesn't learn enough about the minority class.

32. What techniques can be used to address imbalanced data

1. Resampling:

   Up-sampling the minority class

   Down-sampling the majority class

2. Synthetic Data Generation:

   SMOTE (Synthetic Minority Over-sampling Technique)

3. Algorithmic Adjustments:

   Use models with class weighting

   Use cost-sensitive learning

4. Evaluation Metric Changes:

   Use F1-score, Precision, Recall, AUC instead of accuracy

33. Explain the process of up-sampling and down-sampling

Up-Sampling:

Increases the number of minority class examples by duplicating existing samples or generating synthetic ones.

Down-Sampling:

Reduces the number of majority class examples by removing random samples to balance the classes.

34. When would you use up-sampling versus down-sampling?

Method	When to Use
Up-Sampling	When minority class is very small and you have **sufficient computing resources**.
Down-Sampling	When the dataset is very large and you want to **reduce training time** or avoid overfitting.
Both	Sometimes used together for better balance without sacrificing too much data.

35. What is SMOTE and how does it work?

SMOTE (Synthetic Minority Over-sampling Technique) is a method to generate synthetic data points for the minority class.

How it works:

For each minority class sample, it finds its k-nearest neighbors.

It selects one neighbor randomly.

It creates a new synthetic sample along the line between the sample and neighbor.

36. Explain the role of SMOTE in handling imbalanced data

SMOTE helps to:

```
Balance the class distribution without duplicating data.

Improve model learning by exposing it to more representative minority class samples.

Avoid overfitting that can happen with simple duplication.
```

37. Discuss the advantages and limitations of SMOTE

Pros	Cons
Reduces overfitting	Can introduce noise if minority class is noisy
Generates more diverse examples	May create overlapping classes
Works better than random oversampling	Not effective if dataset is high-dimensional or has small minority clusters

38. Provide examples of scenarios where SMOTE is beneficial

Medical Diagnosis: Where disease cases are rare compared to healthy samples.

Fraud Detection: Fraudulent transactions are far fewer than legitimate ones.

Spam Filtering: Spam messages are typically outnumbered by normal emails.

Customer Churn Prediction: Fewer customers actually leave compared to those who stay

39. Define data interpolation and its purpose

Data interpolation is the process of estimating missing or unknown values within the range of known data points.

Purpose:

To fill in missing values in time series or numerical data

To ensure data continuity and smoothness

To support visualization or model training

40. What are the common methods of data interpolation?

Linear Interpolation: Connects two known values with a straight line

Polynomial Interpolation: Uses polynomial functions for curve fitting

Spline Interpolation: Fits smooth curves between points (cubic spline is common)

Nearest-Neighbor Interpolation: Uses the value of the nearest known point

Time-based Interpolation: Often used in time series data to infer missing timestamps

41. Discuss the implications of using data interpolation in machine learning

Data interpolation is used to estimate unknown values between known data points.

Implications:

Positive:

Helps fill in missing data, making models more robust.

Useful in time series to maintain continuity.

Reduces data loss and preserves trends in small datasets.

Negative:

Introduces bias if the interpolation does not represent real trends.

Can lead to overfitting if interpolated values are treated as true observations.

Might misrepresent variability in the data.

42. What are outliers in a dataset?

Outliers are data points that deviate significantly from other observations in the dataset. They may be due to:

Measurement errors

Data entry errors

True variability in the data (e.g., rare events)

43. Explain the impact of outliers on machine learning models

Skew model predictions (especially in regression and mean-based algorithms).

Increase error rate in models sensitive to distance (e.g., KNN, SVM).

Affect convergence in optimization algorithms like gradient descent.

However, in some domains (fraud detection), outliers are meaningful and should not be removed.

44. Discuss techniques for identifying outliers

Statistical methods:

Z-score ($|Z| > 3$ often indicates an outlier)

IQR method (values outside 1.5 * IQR are outliers)

Visualization:

Box plots

Scatter plots

Histogram

Model-based methods:

Isolation Forest

One-Class SVM

DBSCAN clustering

45. How can outliers be handled in a dataset?

```
Remove: If caused by error and not representative.

Cap/Floor (Winsorization): Limit values to a certain range.

Transform: Use log/square root to reduce skewness.

Impute: Replace with mean/median if appropriate.

Model Adjustment: Use models robust to outliers (e.g., tree-based models).
```

46. Compare and contrast Filter, Wrapper, and Embedded methods for feature selection

Method	Description	Evaluation Criterion
Filter	Select features based on statistical measures	Independent of ML model
Wrapper	Select features by evaluating model performance	Uses ML model to test subsets
Embedded	Feature selection during model training	Built into model algorithm

47. Provide examples of algorithms associated with each method

`Filter:`

Chi-square test

Mutual information

Pearson correlation

`Wrapper:`

Recursive Feature Elimination (RFE)

Forward/Backward Selection

`Embedded:`

LASSO (L1 Regularization)

Decision Trees (e.g., feature importance from Random Forest)

48. Discuss the advantages and disadvantages of each feature selection method

Method	Advantages	Disadvantages
Filter	Fast, model-agnostic, simple	Ignores feature interaction with model
Wrapper	Better performance, considers feature interaction	Computationally expensive, prone to overfitting
Embedded	Efficient, model-aware, balances accuracy/speed	Model-specific, less generalizable

49. Explain the concept of feature scaling

Feature scaling is the process of normalizing or standardizing independent variables to a common scale. It's crucial for algorithms that are sensitive to the scale of data, such as:

KNN, SVM, Gradient Descent-based models, PCA

50. Describe the process of standardization

Standardization transforms data to have:

Mean = 0

Standard Deviation = 1

Formula:

= (−)/

Where:

x is the original value

is the mean

is the standard deviation

This process helps models treat all features equally and improves convergence in optimization.

51. How does mean normalization differ from standardization?

Mean Normalization:

Scales data so that its mean becomes 0 and the values are in the range [-1, 1]. Formula:

x = x− / max−min

Standardization (Z-score normalization):

Scales data to have mean = 0 and standard deviation = 1. Formula:

x = x− /

Key difference:

Mean normalization uses the data range.

Standardization uses standard deviation.

52. Discuss the advantages and disadvantages of Min-Max scaling

Advantages:

Preserves relationships and relative distances between values.

Keeps values within a bounded range [0, 1], making it suitable for algorithms like neural networks and gradient descent.

Disadvantages:

Sensitive to outliers – a single large or small value can skew the scaling.

Not robust; requires all data to be known in advance.

53. What is the purpose of unit vector scaling

Unit vector scaling (also called normalization) rescales a feature vector so that its length (or norm) is 1. Formula (L2 Norm):

x = x/ x 2

Purpose:

Ensures equal contribution of each feature.

Useful in text classification, KNN, and cosine similarity-based models.

54. Define Principle Component Analysis (PCA)

PCA is a dimensionality reduction technique that transforms a high-dimensional dataset into a lower-dimensional space by finding new orthogonal axes (principal components) that capture maximum variance in the data

55. Explain the steps involved in PCA

    1. Standardize the dataset.
    2. Compute covariance matrix.
    3. Calculate eigenvalues and eigenvectors of the covariance matrix.
    4. Sort eigenvectors by decreasing eigenvalues.
    5. Select top k eigenvectors (principal components).
    6. Transform the data into the new lower-dimensional space.

56. Discuss the significance of eigenvalues and eigenvectors in PCA

Eigenvectors: Define the direction of new axes (principal components).

Eigenvalues: Indicate the amount of variance captured by each principal component.

→ Larger eigenvalue = more important component → They help in ranking components to choose top k dimensions.

57. How does PCA help in dimensionality reduction

PCA projects data onto a smaller set of orthogonal axes (principal components) that capture the most variance.

It removes redundant features, reduces noise, and speeds up model training while retaining most of the important information.

58. Define data encoding and its importance in machine learning

Data encoding is the process of converting categorical data into numerical format so that it can be used in machine learning algorithms.

Importance:

Many ML models (e.g., linear regression, SVM) require numerical input.

Helps models interpret and learn from categorical variables.

59. Explain Nominal Encoding and provide an example.

Nominal Encoding (a.k.a. Label Encoding) assigns an integer to each category in a feature. Used for non-ordinal categorical data.

Example:

```
Color: Red, Green, Blue → Red=0, Green=1, Blue=2
```

60. Discuss the process of One Hot Encoding

One Hot Encoding creates binary columns for each category in a feature. Each row has a 1 in the column corresponding to its category, and 0 elsewhere.

Example:

```
Color: Red, Green, Blue
→ Red  → [1, 0, 0]
→ Green→ [0, 1, 0]
→ Blue → [0, 0, 1]
```

Prevents models from interpreting categories as ordinal.

Increases dimensionality (especially with many categories).

61. How do you handle multiple categories in One Hot Encoding

When a feature has many categories, One Hot Encoding can create a large number of columns, leading to:

High dimensionality

Increased memory usage

Sparsity of data

Solutions:

Limit top categories and group rare ones as “Other”

Feature hashing to reduce dimensionality

Embedding techniques for high-cardinality data (especially in deep learning)

62. Explain Mean Encoding and its advantages

Mean Encoding replaces each category with the mean of the target variable for that category.

Example:

City	Avg Purchase (Target)
Delhi	2000
Mumbai	1500

Encoded:

City: Delhi → 2000, Mumbai → 1500

Advantages:

Captures target-category relationship

Low-dimensional, unlike One Hot Encoding

Useful for tree-based models

Caution: Can lead to target leakage → use techniques like K-fold mean encoding to avoid overfi

63. Provide examples of Ordinal Encoding and Label Encoding

Ordinal Encoding: Used for ordered categories

Education: High School = 1, Bachelor = 2, Master = 3, PhD = 4

Label Encoding: Used for nominal data (no order), but still assigns integers

Color: Red = 0, Green = 1, Blue = 2

64. What is Target Guided Ordinal Encoding and how is it used

In Target Guided Ordinal Encoding, categories are ranked based on the mean of the target variable, and then assigned ordered integers.

Steps:

Calculate mean target per category.

Sort categories by this mean.

Assign integers accordingly.

Example:

Category	Avg Target
C	10
A	20
B	30

Encoded:

C = 0, A = 1, B = 2

Useful for models that benefit from ordinal information
Needs to be handled carefully to prevent overfitting

65. Define covariance and its significance in statistics

Covariance measures the directional relationship between two variables.

Formula:

$\mathrm{Cov}(X,Y) = \ (xi - \ x\ )(yi - \ y\ )/n-1$

Positive covariance → variables increase together

Negative covariance → one increases while the other decreases

Significance:

Helps identify linear relationships

Used in PCA and portfolio risk analysis

66. Explain the process of correlation check

A correlation check helps find relationships between features, typically using Pearson or Spearman methods.

Steps:

Choose correlation method (e.g., Pearson, Spearman)

Compute correlation matrix

Visualize using heatmap (optional)

Drop one of two features with high correlation (e.g., > 0.8)

Helps avoid multicollinearity

67. What is the Pearson Correlation Coefficient

The Pearson Correlation Coefficient (r) measures the linear relationship between two variables.

Formula:

= Cov(X,Y)/ X Y

Range: -1 to +1

+1 → perfect positive linear correlation

-1 → perfect negative linear correlation

0 → no linear correlation

68. How does Spearman's Rank Correlation differ from Pearson's Correlation

Feature	Pearson	Spearman
Type	Measures **linear** relationships	Measures **monotonic** relationships
Assumptions	Requires **normal distribution**	**No assumption** of distribution
Data Handling	Uses raw values	Uses **ranks**
Outlier Sensitivity	Sensitive	Less sensitive

69. Discuss the importance of Variance Inflation Factor (VIF) in feature selection

VIF quantifies how much a feature is correlated with other features (i.e., multicollinearity).

Formula:

VIF= 1/1−R^2

VIF > 5 or 10 → high multicollinearity → consider removing the feature.

Helps improve model interpretability
Mostly used for linear regression models

70. Define feature selection and its purpose

Feature selection is the process of selecting the most relevant features from a dataset for use in model building.

`Purpose:`

Improve model performance and accuracy

Reduce overfitting

Shorten training time

Enhance interpretability

`Feature selection can be done via:`

Filter methods (e.g., correlation)

Wrapper methods (e.g., RFE)

Embedded methods (e.g., LASSO)

71. Explain the process of Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a wrapper-based feature selection technique that recursively removes the least important features.

`Process:`

1. Choose a base model (e.g., linear regression, decision tree).
2. Train the model on the full feature set.
3. Rank features by importance (e.g., coefficients or impurity).
4. Remove the least important feature(s).
5. Repeat until the desired number of features is reached.

```
Finds optimal subset of features
Computationally expensive for large datasets
```

72. How does Backward Elimination work

Backward Elimination starts with all features, and iteratively removes the least significant one based on a metric (e.g., p-value, feature importance).

`Steps:`

1. Train model on all features.
2. Evaluate significance of each feature.
3. Remove the least significant one.
4. Repeat until all remaining features are significant.

```
Simple, interpretable

May miss combinations of weakly significant but useful features
```

73. Discuss the advantages and limitations of Forward Elimination

`Advantages:`

Starts from nothing → prevents unnecessary complexity.

Faster than backward elimination with many features.

Useful when model training is expensive.

`Limitations:`

Greedy: might miss optimal feature combinations.

Sensitive to the initial selection.

May stop too early due to local optima.

74. What is feature engineering and why is it important

Feature Engineering is the process of creating, transforming, or selecting features to improve model performance.

`Importance:`

Increases accuracy and efficiency

Helps models learn patterns more easily

Bridges gap between raw data and machine learning models

75. Discuss the steps involved in feature engineering

1. Understand the data: Explore data types, distributions, and domain context.
2. Handle missing values: Imputation or removal.
3. Encode categorical variables: One-Hot, Label, Target encoding.
4. Create new features: Ratios, differences, time-based features.
5. Scale/normalize: StandardScaler, MinMaxScaler, etc.
6. Transform: Log, square root, Box-Cox, etc.
7. Reduce dimensions (if needed): PCA, t-SNE.
8. Select features: Use filter/wrapper/embedded methods.

76. Provide examples of feature engineering techniques

Binning: Grouping numerical values (e.g., age → age groups)

Date decomposition: Extracting year/month/day from datetime

Interaction terms: feature1 * feature2

Aggregations: Mean, sum, count per group

Log transforms: To reduce skewness

Text vectorization: TF-IDF, word embeddings for text data

Polynomial features: Adding squared or interaction terms

77. How does feature selection differ from feature engineering

Aspect	Feature Selection	Feature Engineering
Definition	Choosing the best subset of features	Creating/modifying new features
Purpose	Reduce redundancy/noise	Extract meaningful information
Output	Subset of existing features	Enhanced/new feature set
Examples	RFE, LASSO, Chi-Square	Log transform, datetime split, NLP TF-IDF

78. Explain the importance of feature selection in machine learning pipelines

Reduces overfitting

Enhances model accuracy

Decreases training time

Improves interpretability

Reduces dimensionality, helping in visualization and debugging

`Essential step before modeling in automated ML pipelines (like sklearn's Pipeline, FeatureUnion`

79. Discuss the impact of feature selection on model performance

Improves generalization by removing noise

Speeds up training and inference

Helps models focus on relevant patterns

Too aggressive selection may remove useful information, hurting performance

→ Balance is key: avoid both under- and over-selection.

80. How do you determine which features to include in a machine-learning model?

`Approaches:`

1. Statistical techniques: Correlation, ANOVA, Chi-square
2. Model-based importance: Tree-based models, LASSO
3. Domain knowledge: Expert insight
4. Automated methods:

   Recursive Feature Elimination (RFE)

   SHAP or Permutation Importance
5. Validation performance: Compare model scores (accuracy, RMSE, F1) with different feature subsets using cross-validation.