MACHINE LEARNING ASSIGNMENT-I

1. Define Artificial Intelligence (AI)
   Artificial intelligence (AI) is the theory and development of computer systems capable of performing tasks that historically required human intelligence, such as recognizing speech, making decisions, and identifying patterns

2. Explain the differences between Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and Data Science (DS).
   AI is the broader concept of machines being able to carry out tasks in a way that we would consider "smart."
   ML is a subset of AI that focuses on the ability of machines to receive data and learn for themselves without being explicitly programmed.
   DL is a subset of ML that uses neural networks with multiple layers (deep neural networks) to analyze various factors of data.
   DS is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data.

3. How does AI differ from traditional software development?
   Artificial intelligence (AI) represents a significant paradigm shift in the realm of software development. Unlike traditional software, which relies on static rules and explicit programming, AI systems, particularly those leveraging machine learning and neural networks, have the capability to adapt and learn from data over time.

4. Provide examples of AI, ML, DL, and DS applications
   AI: Siri or Google Assistant, Facebook
   ML: Spam filters, recommendation systems
   DL: Image recognition, speech recognition natural language processing
   DS: predictive modelling in finance, Weather Apps

5. Discuss the importance of AI, ML, DL, and DS in today's world.
   These technologies are driving innovation across industries, improving efficiency, enabling personalized experiences, and solving complex problems in fields like healthcare, finance, and environmental science.

6. What is Supervised Learning?
   Supervised Learning is a type of ML where the algorithm learns from trained on a labelled dataset, where each data point has a corresponding label or output value. The algorithm learns to map the input data to the desired output, allowing it to make predictions for new, unseen data

7. Provide examples of Supervised Learning algorithms
   1)Linear Regression 2) Logistic Regression 3) Support Vector Machines (SVM) 4) Decision Trees and Random Forests

8. Explain the process of Supervised Learning
   1. Collect labelled data
   2. Split data into training and testing sets
   3. Choose and initialize a model
   4. Train the model on the training data
   5. Make predictions on the test data
   6. Evaluate model performance
   7. Tune parameters and repeat if necessary
9. What are the characteristics of Unsupervised Learning?
   - Works with unlabelled data
   - Finds hidden patterns or structures in data
   - No predefined output variable
   - Used for clustering, dimensionality reduction, and association
10. Give examples of Unsupervised Learning algorithms
    - K-means clustering
    - Hierarchical clustering
    - Principal Component Analysis (PCA)

11. Describe Semi-Supervised Learning and its significance?
    Semi-Supervised Learning uses both labeled and unlabeled data for training. It is significant because it can improve learning accuracy while reducing the need for labeled data, which can be expensive or time-consuming to obtain.

12. Explain Reinforcement Learning and its applications.
    Reinforcement Learning is a type of ML where an agent learns to make decisions by taking actions in an environment to maximize a reward. Applications include game playing (e.g., AlphaGo), robotics, and autonomous systems.

13. How does Reinforcement Learning differ from Supervised and Unsupervised Learning?
    Reinforcement Learning learns through interaction with an environment, receiving feedback in the form of rewards or penalties. It does not require labeled data (like Supervised Learning) or find hidden patterns (like Unsupervised Learning), but instead learns optimal behavior through trial and error.

14. What is the purpose of the Train-Test-Validation split in machine learning?
    This split helps assess the model's performance on unseen data, prevent overfitting, and provide an unbiased evaluation of the final model.

15. Explain the significance of the training set.
    The training set is used to teach the model, allowing it to learn patterns and relationships in the data. It is crucial for the model to generalize well to new, unseen data.

16. How do you determine the size of the training, testing, and validation sets?
    Common splits are 60-20-20 or 70-15-15, but this can vary based on the size of the dataset and specific requirements. Larger datasets might use a smaller percentage for testing and validation.

17. What are the consequences of improper Train-Test-Validation splits?
    Improper splits can lead to overfitting, underfitting, or biased performance estimates. This can result in models that perform poorly on new data.

18. Discuss the trade-offs in selecting appropriate split ratios
    - Larger training sets provide more data for learning but may not leave enough for testing and validation.
    - Larger test/validation sets provide more reliable performance estimates but may limit the model's learning capacity.
    - The optimal ratio depends on the size of the dataset and the complexity of the problem.

19. Define model performance in machine learning
    Model performance refers to how well a model makes predictions or classifications on new, unseen data. It is typically measured using various metrics such as accuracy, precision, recall, F1 score, or mean squared error, depending on the type of problem.

20. How do you measure the performance of a machine learning model?
    Performance is measured using various metrics:
    - For classification: accuracy, precision, recall, F1 score, ROC-AUC
    - For regression: mean squared error, mean absolute error, R-squared
      Cross-validation techniques are often used to get a more robust estimate of performance

21. What is overfitting and why is it problematic?
    Overfitting occurs when a model learns the training data too well, including its noise and fluctuations. It is problematic because it leads to poor generalization on new, unseen data.

22. Provide techniques to address overfitting

    - Cross-validation
    - Regularization (L1, L2)
    - Dropout (for neural networks)
    - Early stopping
    - Data augmentation
    - Ensemble methods

23. Explain underfitting and its implications
    Underfitting occurs when a model is too simple to capture the underlying patterns in the data. It results in poor performance on both training and test data, indicating that the model has not learned enough from the available data.

24. How can you prevent underfitting in machine learning models?
    - Increase model complexity
    - Add more relevant features
    - Reduce regularization
    - Train for more epochs (in neural networks)
    - Ensure sufficient training data

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

    The bias-variance trade-off is about finding the sweet spot between underfitting (high bias) and overfitting (high variance). A good model should have low bias and low variance, accurately capturing the underlying patterns without being too sensitive to fluctuations in the training data.

26. What are the common techniques to handle missing data?
    - Deletion (listwise or pairwise)
    - Mean/median/mode imputation
    - Regression imputation
    - Multiple imputation
    - Using algorithms that handle missing values (e.g., some decision tree-based methods)

27. Explain the implications of ignoring missing data

    Ignoring missing data can lead to biased results, reduced statistical power, and inaccurate conclusions. It may also result in a loss of important information and potentially skew the model's performance.

28. Discuss the pros and cons of imputation methods

    Pros:
    - Preserves sample size
    - Can improve model accuracy
    - Allows use of complete-data methods

    Cons:
    - May introduce bias if not done carefully
    - Can underestimate variability
    - May lead to overfitting if not validated properly

29. How does missing data affect model performance?

    Missing data can reduce the statistical power of a model, introduce bias, and lead to inaccurate predictions. It may also limit the use of certain algorithms that require complete data.

30. Define imbalanced data in the context of machine learning

    Imbalanced data refers to a situation where the classes in a classification problem are not represented equally, with one class having significantly fewer samples than the others.

31. Discuss the challenges posed by imbalanced data
    - Biased model performance towards the majority class
    - Difficulty in learning the minority class
    - Misleading evaluation metrics (e.g., high accuracy but poor minority class prediction)
    - Risk of treating important minority cases as noise

32. What techniques can be used to address imbalanced data?
    - Oversampling minority class (e.g., SMOTE)
    - Under sampling majority class
    - Combination of over- and under-sampling
    - Adjusting class weights
    - Using appropriate evaluation metrics (e.g., F1-score, ROC-AUC)
    - Ensemble methods (e.g., Balanced Random Forest Classifier)

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

    Up-sampling: Increasing the number of minority class samples, often by duplication or synthetic data generation.

    Down-sampling: Reducing the number of majority class samples to match the minority class.

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

    - You have a small dataset and cannot afford to lose information
    - The minority class is very underrepresented Use down-sampling when:
    - You have a large dataset
    - The majority class has redundant information Often, a combination of both is used to find the optimal balance.

35. What is SMOTE and how does it work?

    SMOTE (Synthetic Minority Over-sampling Technique) is an algorithm for handling imbalanced datasets. It works by creating synthetic examples of the minority class by interpolating between existing minority instances.

36. Explain the role of SMOTE in handling imbalanced data

    SMOTE helps balance the dataset by increasing the number of minority class samples. This improves the model's ability to learn patterns in the minority class and reduces bias towards the majority class.

37. Discuss the advantages and limitations of SMOTE

    Advantages:
    - Improves classifier performance
    - Avoids overfitting caused by simple oversampling
    - Generates realistic synthetic samples

    Limitations:
    - May increase noise if minority class is already noisy
    - Can lead to overgeneralization
    - Does not consider the majority class distribution

38. Provide examples of scenarios where SMOTE is beneficial

- Fraud detection in financial transactions
- Rare disease diagnosis in medical datasets
- Churn prediction with few churned customers
- Anomaly detection in manufacturing processes

39. Define data interpolation and its purpose

Data interpolation is the process of estimating unknown values within the range of a discrete set of known data points. Its purpose is to fill gaps in data or to create a continuous function from discrete data points.

40. What are the common methods of data interpolation?

- Linear interpolation
- Polynomial interpolation
- Spline interpolation
- Cubic interpolation

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

- Can help fill missing data points
- May introduce bias if the interpolation method does not match the true underlying distribution
- Can smooth out noise, potentially removing important variations
- Useful for time series data and continuous variables
- May not be suitable for categorical data or when the relationship between variables is complex

42. What are outliers in a dataset?

Outliers are data points that significantly differ from other observations in the dataset. They can be caused by variability in the measurement, experimental errors, or indicate genuine extreme values in the population.

43. Explain the impact of outliers on machine learning models

- Can significantly skew statistical measures (mean, standard deviation)
- May lead to biased model parameters
- Can affect model performance, especially in linear models
- May result in overfitting if the model tries to accommodate outliers
- Can provide valuable insights in some cases (e.g., fraud detection)

44. Discuss techniques for identifying outliers

- Statistical methods (e.g., Z-score, IQR method)
- Visualization techniques (box plots, scatter plots)
- Distance-based methods (e.g., Local Outlier Factor)
- Density-based methods (e.g., DBSCAN)
- Machine learning-based methods (Isolation Forest, One-Class SVM)

45. How can outliers be handled in a dataset?
- Removal (if they are confirmed errors)
- Transformation (e.g., log transformation)
- Capping
- Treating as a separate category
- Using robust statistical methods or algorithms less sensitive to outliers

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

Filter methods:
- Select features independently of the model
- Fast and computationally efficient
- May not consider feature interactions

Wrapper methods:
- Use a predictive model to score feature subsets
- Can detect feature interactions
- Computationally intensive

Embedded methods:
- Perform feature selection as part of the model training process
- Balance between filter and wrapper methods in terms of computational cost
- Specific to a given learning algorithm

47. Provide examples of algorithms associated with each method

Filter: Chi-squared test, correlation coefficient, variance threshold

Wrapper: Recursive feature elimination, forward/backward selection

Embedded: Lasso, Ridge regression, Decision trees

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

Filter:
Advantages: Fast, scalable, independent of the model
Disadvantages: Ignores feature dependencies, may select redundant features
Wrapper:
Advantages: Considers feature interactions, usually provides the best feature subset for that model
Disadvantages: Computationally expensive, risk of overfitting
Embedded:
Advantages: Interacts with the model during training, less computationally intensive than wrapper methods
Disadvantages: Tied to a specific model, can be complex to implement

49. Explain the concept of feature scaling

Feature scaling is the process of normalizing the range of independent variables or features of data. It has done to standardize the range of features in a dataset, which can be especially important for algorithms that calculate distances between data points.

50. Describe the process of standardization
Standardization (or Z-score normalization) transforms features to have a mean of 0 and a standard deviation of 1. The formula is: $z = (x - \mu) / \sigma$, where x is the original value, μ is the mean, and σ is the standard deviation.

51. How does mean normalization differ from standardization?
Mean normalization scales feature to be between -1 and 1, with a mean of 0.
The formula is: x_new = (x - mean(x)) / (max(x) - min(x)).
Unlike standardization, it does not ensure a specific standard deviation.

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

Advantages:
- Preserves zero entries in sparse data
- Preserves the shape of the original distribution
- Brings all values into a fixed range (usually [0,1])

Disadvantages:
- Does not handle outliers well
- Does not centre the data around zero, which might be required for some algorithms

53. What is the purpose of unit vector scaling?
Unit vector scaling (or normalization) scales the features of a data point so that their Euclidean length (magnitude) equals 1. It is useful when the direction of the data matters more than the magnitude, such as in text classification or clustering.

54. Define Principal Component Analysis (PCA)
PCA is a dimensionality reduction technique that transforms a dataset into a new coordinate system, where the new axes (principal components) represent the directions of maximum variance in the data.

55. Explain the steps involved in PCA:
- Standardize the data
- Compute the covariance matrix
- Calculate eigenvectors and eigenvalues of the covariance matrix
- Sort eigenvectors by decreasing eigenvalues
- Choose k eigenvectors as the new k dimensions
- Transform the original dataset to get k-dimensional feature subspace

56. Discuss the significance of eigenvalues and eigenvectors in PCA
Eigenvectors represent the directions of maximum variance in the data (principal components), while eigenvalues represent the amount of variance explained by each eigenvector. They help in determining which principal components are most important for representing the data.

57. How does PCA help in dimensionality reduction?

    PCA reduces dimensionality by projecting the data onto a lower-dimensional subspace while preserving as much variance as possible.

    This helps in:

    - Reducing computational complexity
    - Mitigating the curse of dimensionality
    - Visualizing high-dimensional data
    - Removing noise and redundant features

58. Define data encoding and its importance in machine learning:
    Data encoding is the process of converting categorical variables into a numerical format that can be used by machine learning algorithms. It is important because most ML algorithms work with numerical data, and proper encoding can significantly impact model performance.

59. Explain Nominal Encoding and provide an example
    Nominal Encoding assigns a unique integer to each category in a categorical variable.
    For example, for colours: Red -> 1 Blue -> 2 Green -> 3 This encoding does not imply any ordinal relationship between categories.

60. Discuss the process of One Hot Encoding:
    One Hot Encoding creates binary columns for each category in a categorical variable. Each column represents a category and contains 1 if the data point belongs to that category, and 0 otherwise.
    For example: Colour | Red | Blue | Green
    Red | 1 | 0 | 0
    Blue | 0 | 1 | 0
    Green | 0 | 0 | 1
61. How do you handle multiple categories in One Hot Encoding?
    For multiple categories, create a binary column for each category. If there are many categories, consider:
    - Grouping less frequent categories into an “Other” category
    - Using dimension reduction techniques after encoding
    - Employing alternative encoding methods like feature hashing

62. Explain Mean Encoding and its advantages
    Mean Encoding replaces categorical variables with the mean of the target variable for each category.
    Advantages
    - Captures information about the target variable
    - Can handle high cardinality features efficiently
    - Often improves model performance for certain algorithms

63. Provide examples of Ordinal Encoding and Label Encoding
    Ordinal Encoding: Assigns integers based on the order of categories.
    Example (Education): High School -> 1, Bachelor’s -> 2, Master’s -> 3, PhD -> 4
    Label Encoding: Assigns unique integers to each category without implying order.
    Example (Colors): Red -> 0, Blue -> 1, Green -> 2

64. What is Target Guided Ordinal Encoding and how is it used?
Target Guided Ordinal Encoding orders categories based on their relationship with the target variable. It is used by:

- Calculating the mean of the target variable for each category
- Ordering categories based on these means
- Assigning ordinal numbers based on this order This preserves information about the target variable while creating an ordinal representation.

65. Define covariance and its significance in statistics
Covariance measures how two variables change together.
It is significant is:

- It indicates the direction of the linear relationship between variables
- It is used in calculating correlation
- It is crucial in techniques like Principal Component Analysis

66. Explain the process of correlation check

A correlation check involves:

- Calculating correlation coefficients between variables
- Creating a correlation matrix or heatmap
- Identifying strong correlations (positive or negative)
- Analysing the relationships for potential feature selection or multicollinearity issues

67. What is the Pearson Correlation Coefficient?
The Pearson Correlation Coefficient measures the strength and direction of the linear relationship between two continuous variables. It ranges from -1 to 1, where:

- 1 indicates a perfect positive linear relationship
- -1 indicates a perfect negative linear relationship
- 0 indicates no linear relationship

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

- Measures monotonic relationships (not just linear)
- Works with ordinal data
- Is less sensitive to outliers
- Uses ranked values instead

Pearson Correlation

- Measures linear relationships only
- Works best with continuous data
- Is sensitive to outliers
- Uses raw data values

69. Discuss the importance of Variance Inflation Factor (VIF) in feature selection
VIF quantifies multicollinearity in regression analysis.
It is important because:

- High VIF indicates that a feature is highly correlated with others

- It helps identify redundant features that can be removed
- Reducing multicollinearity improves model stability and interpretability

70. Define feature selection and its purpose
    Feature selection is the process of selecting a subset of relevant features for use in model construction. Its purpose is to:
    - Simplify models for easier interpretation
    - Reduce training time
    - Enhance generalization by reducing overfitting
    - Improve model performance by removing irrelevant or redundant features

71. Explain the process of Recursive Feature Elimination
    Recursive Feature Elimination (RFE) is a feature selection method that works by recursively removing attributes and building a model on those attributes that remain. It involves:
    - Training the model on all features
    - Calculating feature importance
    - Removing the least important feature(s)
    - Repeating steps 1-3 until the desired number of features is reached

72. How does Backward Elimination work?

    Backward Elimination is a feature selection technique that starts with all features and iteratively removes the least significant ones:
    - Start with all features
    - Fit a model and compute the significance of each feature
    - Remove the least significant feature
    - Repeat steps 2-3 until all remaining features are significant or a stopping criterion is met

73. Discuss the advantages and limitations of Forward Elimination

    Advantages:
    - Can be computationally efficient, especially with many features
    - May find a good feature subset quickly

    Limitations:

    - May not find the optimal feature subset
    - Can miss feature interactions
    - Prone to overfitting if not properly validated

74. What is feature engineering and why is it important?

    Feature engineering is the process of using domain knowledge to create new features or transform existing ones. It is important because:
    - It can uncover hidden patterns in data
    - It often leads to improved model performance
    - It allows incorporation of domain expertise into the modeling process
    - It can make certain patterns more explicit for the model to learn

75. Discuss the steps involved in feature engineering
    - Understand the problem and domain
    - Gather and explore the data
    - Brainstorm potential new features
    - Create new features (e.g., combining existing features, extracting from text/dates)
    - Validate the usefulness of new features
    - Iterate and refine

76. Provide examples of feature engineering techniques
    - Binning continuous variables
    - Creating interaction terms
    - Extracting components from dates (day of week, month, etc.)
    - Text feature extraction (TF-IDF, word embeddings)
    - Domain-specific calculations (e.g., BMI from height and weight)
    - Aggregating time series data

77. How does feature selection differ from feature engineering?
    Feature selection chooses a subset of existing features, while feature engineering creates new features or transforms existing ones. Feature selection reduces dimensionality, while feature engineering can increase it by adding new informative features.

78. Explain the importance of feature selection in machine learning pipelines
    Feature selection in ML pipelines is important because it:
    - Reduces overfitting by removing irrelevant features
    - Improves model performance and generalization
    - Reduces training time and computational requirements
    - Enhances model interpretability
    - Helps in dealing with the curse of dimensionality

79. Discuss the impact of feature selection on model performance

    Feature selection can impact model performance by:
    - Improving accuracy by focusing on the most relevant features
    - Reducing noise in the data
    - Preventing overfitting, leading to better generalization
    - Potentially worsening performance if important features are incorrectly removed

80. How do you determine which features to include in a machine-learning model?
    To determine which features to include:
    - Use domain knowledge to identify potentially important features
    - Perform exploratory data analysis to understand feature distributions and relationships
    - Apply feature selection techniques (filter, wrapper, or embedded methods)
    - Use feature importance scores from tree-based models
    - Conduct correlation analysis to identify redundant features
    - Perform cross-validation to assess the impact of different feature subsets
    - Consider the interpretability requirements of the model
    - Iterate and refine based on model performance