ML Assignment -3

1. What are ensemble techniques in machine learning?
Ensemble techniques combine multiple models to improve accuracy.

2. Explain bagging and how it works in ensemble techniques.
Bagging reduces variance by training multiple models on random data subsets.

3. What is the purpose of bootstrapping in bagging?
Bootstrapping creates multiple random samples for training models.

4. Describe the random forest algorithm.
Random Forest is an ensemble of decision trees that improves accuracy through averaging.

5. How does randomization reduce overfitting in random forests?
Randomization prevents overfitting by introducing diversity in the trees.

6. Explain the concept of feature bagging in random forests.
Feature bagging involves selecting random subsets of features for each tree.

7. What is the role of decision trees in gradient boosting?
Decision trees are weak learners that improve iteratively in boosting.

8. Differentiate between bagging and boosting.
Bagging trains models independently, boosting trains sequentially, correcting previous mistakes.

9. What is the AdaBoost algorithm, and how does it work?
AdaBoost adjusts weights of misclassified data to improve weak learners.

10. Explain the concept of weak learners in boosting algorithms.
Weak learners are models that perform slightly better than random chance.

11. Describe the process of adaptive boosting.
Adaptive boosting adjusts weights based on misclassification to improve performance.

12. How does AdaBoost adjust weights for misclassified data points?
It increases weights for misclassified data to focus on hard-to-classify points.

13. Discuss the XGBoost algorithm and its advantages over traditional gradient boosting.

XGBoost uses regularization and optimization for faster, more accurate models.

14. Explain the concept of regularization in XGBoost.
Regularization in XGBoost controls model complexity to prevent overfitting.

15. What are the different types of ensemble techniques?
Types include bagging, boosting, and stacking.

16. Compare and contrast bagging and boosting.
Bagging reduces variance; boosting reduces bias.

17. Discuss the concept of ensemble diversity.
Diversity in ensemble models leads to better generalization.

18. How do ensemble techniques improve predictive performance?
By combining multiple models, they reduce overfitting and increase robustness.

19. Explain the concept of ensemble variance and bias.
Ensemble learning reduces variance and bias by combining multiple models.

20. Discuss the trade-off between bias and variance in ensemble learning.
Higher bias leads to underfitting, higher variance to overfitting.

21. What are some common applications of ensemble techniques?
Applications include fraud detection, image classification, and recommendation systems.

22. How does ensemble learning contribute to model interpretability?
Ensemble models are harder to interpret but can provide more reliable predictions.

23. Describe the process of stacking in ensemble learning.
Stacking combines predictions from multiple models using a meta-learner.

24. Discuss the role of meta-learners in stacking.
Meta-learners combine predictions from base models to improve final results.

25. What are some challenges associated with ensemble techniques?
Challenges include high computational cost and complexity.

26. What is boosting, and how does it differ from bagging?
Boosting trains models sequentially, bagging trains them independently.

27. Explain the intuition behind boosting.
Boosting focuses on improving weak learners by correcting errors.

28. Describe the concept of sequential training in boosting.
Each model in boosting trains on the mistakes of the previous one.

29. How does boosting handle misclassified data points?
Boosting adjusts weights to focus on misclassified data.

30. Discuss the role of weights in boosting algorithms.
Weights determine the influence of each data point in the learning process.

31. What is the difference between boosting and AdaBoost?
AdaBoost is a specific boosting algorithm that focuses on reweighting misclassified data.

32. How does AdaBoost adjust weights for misclassified samples?
It increases the weight of misclassified samples in subsequent models.

33. Explain the concept of weak learners in boosting algorithms.
Weak learners are simple models that improve iteratively.

34. Discuss the process of gradient boosting.
Gradient boosting builds trees sequentially, correcting the errors of previous trees.

35. What is the purpose of gradient descent in gradient boosting?
Gradient descent minimizes the loss function to improve model accuracy.

36. Describe the role of learning rate in gradient boosting.
The learning rate controls the size of updates to model parameters.

37. How does gradient boosting handle overfitting?
Gradient boosting uses regularization and early stopping to prevent overfitting.

38. Discuss the differences between gradient boosting and XGBoost.
XGBoost is faster and more efficient with additional regularization.

39. Explain the concept of regularized boosting.
Regularized boosting applies penalty terms to reduce model complexity.

40. What are the advantages of using XGBoost over traditional gradient boosting?
XGBoost is more scalable and efficient, with built-in regularization.

41. Describe the process of early stopping in boosting algorithms.
Early stopping halts training when model performance stops improving.

42. How does early stopping prevent overfitting in boosting?
It stops training before the model starts to overfit on the training data.

43. Discuss the role of hyperparameters in boosting algorithms.
Hyperparameters control the learning rate, tree depth, and regularization.

44. What are some common challenges associated with boosting?
Challenges include sensitivity to noisy data and long training times.

45. Explain the concept of boosting convergence.
Convergence occurs when boosting no longer improves the model's performance.

46. How does boosting improve the performance of weak learners?
Boosting iteratively improves weak learners by focusing on difficult samples.

47. Discuss the impact of data imbalance on boosting algorithms.
Imbalanced data can skew boosting results, emphasizing minority class errors.

48. What are some real-world applications of boosting?
Applications include predictive analytics, financial forecasting, and customer segmentation.

49. Describe the process of ensemble selection in boosting.
Ensemble selection selects the best models from the boosted ensemble.

50. How does boosting contribute to model interpretability?
Boosting models are often less interpretable due to their complexity.

51. Explain the curse of dimensionality and its impact on KNN.
Higher dimensions make distance metrics less meaningful, reducing KNN performance.

52. What are the applications of KNN in real-world scenarios?
KNN is used in recommendation systems, image classification, and fraud detection.

53. Discuss the concept of weighted KNN.
Weighted KNN assigns higher weights to closer neighbors for prediction.

54. How do you handle missing values in KNN?
Missing values can be handled by using imputation or ignoring the feature.

55. Explain the difference between lazy learning and eager learning algorithms, and where does KNN fit in?
Lazy learning, like KNN, delays computation until prediction; eager learning precomputes models.

56. What are some methods to improve the performance of KNN?
Improvement methods include feature scaling, dimensionality reduction, and selecting the optimal K.

57. Can KNN be used for regression tasks? If yes, how?
Yes, KNN can predict continuous values by averaging the values of K neighbors.

58. Describe the boundary decision made by the KNN algorithm.
KNN classifies based on the majority label of the nearest neighbors.

59. How do you choose the optimal value of K in KNN?
K is chosen by cross-validation to minimize error.

60. Discuss the trade-offs between using a small and large value of K in KNN.
Small K leads to high variance, large K to high bias.

61. Explain the process of feature scaling in the context of KNN.
Feature scaling normalizes features to prevent distance bias in KNN.

62. Compare and contrast KNN with other classification algorithms like SVM and Decision Trees.
KNN is simple and non-parametric; SVM and Decision Trees model complex relationships.

63. How does the choice of distance metric affect the performance of KNN?
The distance metric determines how similarity is measured, impacting classification accuracy.

64. What are some techniques to deal with imbalanced datasets in KNN?
Techniques include resampling, adjusting class weights, and using distance-weighted voting.

65. Explain the concept of cross-validation in the context of tuning KNN parameters.

Cross-validation evaluates model performance by partitioning data into training and test sets.

66. What is the difference between uniform and distance-weighted voting in KNN?
Uniform voting gives equal weight to all neighbors, distance-weighted gives more weight to closer neighbors.

67. Discuss the computational complexity of KNN.
KNN has high computational complexity, especially with large datasets.

68. How does the choice of distance metric impact the sensitivity of KNN to outliers?
Outliers affect KNN more with distance metrics like Euclidean, as they distort distances.

69. Explain the process of selecting an appropriate value for K using the elbow method.
The elbow method identifies the optimal K by plotting error rates against K values.

70. Can KNN be used for text classification tasks? If yes, how?
Yes, KNN can classify text by converting text into feature vectors (e.g., TF-IDF).

71. How do you decide the number of principal components to retain in PCA?
The number is chosen based on the cumulative explained variance.

72. Explain the reconstruction error in the context of PCA.
Reconstruction error measures the difference between the original and compressed data.

73. What are the applications of PCA in real-world scenarios?
PCA is used in image compression, data visualization, and noise reduction.

74. Discuss the limitations of PCA.
PCA may not work well with nonlinear data and is sensitive to outliers.

75. What is Singular Value Decomposition (SVD), and how is it related to PCA?
SVD is a matrix decomposition method used in PCA for dimensionality reduction.

76. Explain the concept of latent semantic analysis (LSA) and its application in natural language processing.
LSA reduces dimensionality in text data to find latent topics and meanings.

77. What are some alternatives to PCA for dimensionality reduction?
Alternatives include t-SNE, ICA, and autoencoders.

78. Describe t-distributed Stochastic Neighbor Embedding (t-SNE) and its advantages over PCA.
t-SNE preserves local structure better than PCA but is slower and harder to scale.

79. How does t-SNE preserve local structure compared to PCA?
t-SNE minimizes the divergence between pairwise similarities, preserving local relationships.

80. Discuss the limitations of t-SNE.
t-SNE is computationally expensive and sensitive to parameters.

81. What is the difference between PCA and Independent Component Analysis (ICA)?
PCA finds uncorrelated components, while ICA seeks statistically independent components.

82. Explain the concept of manifold learning and its significance in dimensionality reduction.
Manifold learning assumes data lies on a lower-dimensional manifold and aims to preserve this structure during reduction.

83. What are autoencoders, and how are they used for dimensionality reduction?
Autoencoders are neural networks that compress data into a lower-dimensional representation.

84. Discuss the challenges of using nonlinear dimensionality reduction techniques.
Challenges include computational cost, difficulty in interpretation, and sensitivity to hyperparameters.

85. How does the choice of distance metric impact the performance of dimensionality reduction techniques?
The distance metric affects how similarities between data points are measured, influencing reduction quality.

86. What are some techniques to visualize high-dimensional data after dimensionality reduction?
Techniques include t-SNE, PCA, and UMAP for 2D or 3D visualization.

87. Explain the concept of feature hashing and its role in dimensionality reduction.
Feature hashing maps high-dimensional features into a lower-dimensional space using a hash function.

88. What is the difference between global and local feature extraction methods?
Global methods extract overall data patterns, while local methods focus on specific, localized features.

89. How does feature sparsity affect the performance of dimensionality reduction techniques?
Sparsity can make dimensionality reduction more challenging, requiring more sophisticated methods.

90. Discuss the impact of outliers on dimensionality reduction algorithms.
Outliers can distort the data structure, leading to poor performance in dimensionality reduction.