ML Assignment-1

Section-1

1. **AI Definition:** Simulation of human intelligence in machines to perform tasks like reasoning and learning.
2. **AI vs ML vs DL vs DS:**
    - AI: Broad field (robotics, NLP)
    - ML: Subset where systems learn from data
    - DL: Neural networks for complex patterns
    - DS: Data analysis + ML
3. **AI vs Traditional Software:**
    - AI adapts; software follows fixed rules
4. **Examples:**
    - AI: Self-driving cars
    - ML: Spam filters
    - DL: Face recognition
    - DS: Customer analytics
5. **Importance:** Automation, predictive analytics, etc.
6. **Supervised Learning:** Training on labeled data (input-output pairs).
7. **Examples:** Linear Regression, SVM.
8. **Process:** Data → Train → Validate → Test.
9. **Unsupervised Learning:** Finds patterns in unlabeled data (clustering).
10. **Examples:** K-Means, PCA.
11. **Semi-Supervised:** Uses both labeled/unlabeled data.
12. **Reinforcement Learning:** Learns via rewards (e.g., game AI).
13. **RL vs Others:** No labels; goal-oriented learning.

Section-2

14. **Train-Test-Val Split:** Prevents overfitting (70-15-15).
15. **Training Set:** Teaches model patterns.
16. **Split Sizes:** Large data: 70-15-15; Small: 60-20-20.
17. **Bad Splits Cause:** Overfitting/underfitting.

18. **Trade-offs:** More training = better learning but less testing.
19. **Model Performance:** Accuracy on unseen data.
20. **Metrics:** Accuracy, RMSE, F1-Score.
21. **Overfitting:** Model memorizes noise → poor generalization.
22. **Solutions:** Dropout, regularization.
23. **Underfitting:** Too simple → high bias.
24. **Fix:** Add features, reduce regularization.
25. **Bias-Variance Tradeoff:** Balance simplicity/complexity.
26. **Missing Data Handling:** Deletion, imputation (mean/median).
27. **Ignoring Missing Data:** Biases results.
28. **Imputation Pros/Cons:**

- Mean: Simple but reduces variance
- KNN: Accurate but slow

29. **Impact on Performance:** Skews predictions.
30. **Imbalanced Data:** One class dominates (e.g., 95% non-fraud).
31. **Challenges:** Model ignores minority class.
32. **Solutions:** SMOTE, class weights.
33. **Up/Down-Sampling:**

- Up: Duplicate minority
- Down: Reduce majority

34. **SMOTE:** Generates synthetic minority samples.
35. **Interpolation:** Estimates missing values (linear/polynomial).
36. **Outliers:** Extreme values skewing results.
37. **Handling:** Remove, transform, or use robust models.

Section-3

46. **Feature Selection Methods:**

- Filter: Statistical tests (Pearson)
- Wrapper: Uses model performance (RFE)
- Embedded: Built-in (Lasso)

47. **Examples:**

- Filter: Chi-Square

- Wrapper: Backward Elimination

48. **Pros/Cons:**

- Filter: Fast but ignores feature interactions
- Wrapper: Accurate but computationally heavy

49. **Feature Scaling:** Normalizes data (e.g., Min-Max).
50. **Standardization:** $(X - \mu)/\sigma \rightarrow$ mean=0, std=1.
51. **Mean Normalization:** Scales to [-1,1] vs. standardization.
52. **Min-Max Scaling:** [0,1] range; sensitive to outliers.
53. **Unit Vector Scaling:** Normalizes to length=1.
54. **PCA:** Reduces dimensions while preserving variance.
55. **PCA Steps:**
56. Standardize
57. Covariance matrix
58. Eigenvectors/values
59. Select top components
60. **Eigenvalues:** Indicate variance explained.
61. **Dimensionality Reduction:** Projects data to lower dimensions.
62. **Data Encoding:** Converts categories to numbers.
63. **Nominal Encoding:** No order (One-Hot).
64. **One-Hot Encoding:** Binary columns per category.
65. **Many Categories:** Use hashing or embedding.
66. **Mean Encoding:** Replaces categories with target mean.
67. **Ordinal Encoding:** Ordered labels (e.g., small=1, medium=2).
68. **Target Guided Encoding:** Uses target relationship.
69. **Covariance:** Measures joint variability.
70. **Correlation Check:** Pearson/Spearman tests.
71. **Pearson:** Linear relationships (-1 to 1).
72. **Spearman:** Monotonic relationships.
73. **VIF:** Detects multicollinearity (>5 = problematic).
74. **Feature Selection Purpose:** Improves model efficiency.
75. **RFE:** Recursively removes least important features.

76. **Backward Elimination:** Starts with all features, removes one by one.
77. **Forward Elimination:** Starts empty, adds best features.
78. **Feature Engineering:** Creates new features (e.g., log transforms).
79. **Steps:**
80. Domain research
81. Transformations
82. Validation
83. **Examples:** Binning, polynomial features.
84. **Feature Selection vs Engineering:**

- Selection: Chooses best existing features
- Engineering: Creates new features

78. **Importance:** Reduces noise, improves accuracy.
79. **Impact on Performance:** Better features → better models.
80. **Choosing Features:** Use domain knowledge + statistical tests.