1. Introduction to Machine Learning: Definition and Importance

- **ML is a subset of AI where systems learn from data.**
- **It automates decisions, predicts trends, and recognizes patterns.**

---

2. AI vs. ML vs. DL: Key Differences

- **AI: Machines mimicking human intelligence.**
- **ML: Data-driven learning under AI.**
- **DL: Deep learning with neural networks.**

---

3. Types of Machine Learning: Supervised, Unsupervised, Reinforcement Learning

- **Supervised: Learning from labeled data.**
- **Unsupervised: Learning from unlabeled data.**
- **Reinforcement: Learning via rewards and penalties.**

---

4. Challenges in Machine Learning

- **Issues with data quality, overfitting, and model complexity.**

---

5. Applications of Machine Learning

- **Used in healthcare, finance, marketing, and self-driving cars.**

---

6. Data Types: Ordinal, Nominal, Ratio, Interval

- **Ordinal: Ranked data.**
- **Nominal: Categories without order.**
- **Ratio: Data with a true zero.**
- **Interval: Data with equal intervals but no true zero.**

---

7. Structured, Semi-structured, and Unstructured Data

- **Structured: Organized data (e.g., databases).**
- **Semi-structured: Partially organized (e.g., JSON, XML).**
- **Unstructured: Raw data (e.g., images, text).**

---

8. Machine Learning Development Life Cycle

- **Steps include problem definition, data collection, model building, testing, and deployment.**

---

9. Preliminary Project Planning

- **Define the problem, collect data, choose algorithms, and set timelines.**

---

10. Bias and Variance Tradeoff

- **Bias: Error due to overly simple models.**
- **Variance: Error from complex models overfitting data.**

---

11. Overfitting and Underfitting

- **Overfitting: Model is too complex and learns noise.**
- **Underfitting: Model is too simple and cannot capture patterns.**

---

12. Understanding Datasets: Features Selection, Train/Test/Validation Sets

- **Feature Selection: Choosing relevant data attributes.**
- **Train/Test/Validation: Dividing data for training, testing, and validation.**

---

13. Preprocessing Techniques in ML

- **Techniques like normalization, encoding, and imputation to clean data.**

---

14. Cross Validation Techniques

- **Splitting data into multiple subsets for training and validation to improve model accuracy.**

---

15. Hyperparameter Tuning and Optimization

- **Adjusting parameters to enhance model performance.**

---

16. Confusion Matrix: Understanding True Positives and False Positives

- **A matrix showing true positives, false positives, true negatives, and false negatives.**

---

17. Precision, Recall, F1 Score

- **Precision: Accuracy of positive predictions.**
- **Recall: Ability to find all positive instances.**
- **F1 Score: Harmonic mean of precision and recall.**

---

18. Technical Seminar - 1

- **Presentation on a machine learning topic with analysis.**

---

19. Type-1 Error vs. Type-2 Error

- **Type-1 Error: False positive (incorrectly rejecting a true null hypothesis).**
- **Type-2 Error: False negative (failing to reject a false null hypothesis).**

---

20. Error Metrics: MAE, MSE, RMSE

- **MAE: Mean absolute error.**
- **MSE: Mean squared error.**
- **RMSE: Root mean squared error.**

---

21. Introduction to Regression Models: Linear Regression

- **Predicts a continuous output using linear relationships.**

---

22. Cost Function and Gradient Descent for Linear Regression

- **Cost Function: Measures prediction error.**
- **Gradient Descent: Optimization algorithm for minimizing the cost function.**

---

23. Multiple Linear Regression: Concept and Implementation

- **Extends linear regression to predict outcomes based on multiple features.**

---

24. Introduction to Classification Algorithms

- **Algorithms used for classifying data into categories (e.g., decision trees, k-NN).**

---

25. Logistic Regression in Detail

- **A classification algorithm for binary outcomes using logistic function.**

---

26. Decision Trees: Working Mechanism and Applications

- **A tree-like structure for decision-making, widely used in classification tasks.**

---

27. Naive Bayes

- **A probabilistic classifier based on Bayes' theorem assuming independence of features.**

---

28. k-Nearest Neighbors and Support Vector Machines

- **k-NN: Classification based on closest neighbors.**
- **SVM: Classifies data by finding the optimal hyperplane.**

---

29. Introduction to Clustering and Its Importance

- **Grouping similar data points together in unsupervised learning.**

---

30. k-Means Clustering Algorithm: Steps and Implementation

- **A popular clustering algorithm that partitions data into k clusters.**

---

31. Hierarchical Clustering: Agglomerative and Divisive Methods

- **Builds a hierarchy of clusters either by merging (agglomerative) or splitting (divisive).**

---

32. Probabilistic Clustering: Gaussian Mixture Models

- **A probabilistic approach that assumes data points are generated from a mixture of Gaussian distributions.**

---

33. Introduction to Dimensionality Reduction Techniques

- **Techniques like PCA to reduce the number of features while retaining important information.**

---

34. Principal Component Analysis (PCA) – Theory and Implementation

- **A technique for reducing data dimensions by transforming features into principal components.**

---

35. Applications of PCA in Machine Learning

- **PCA is used in image compression, noise reduction, and data visualization.**

---

36. Basic Design of Neural Networks and Architecture Terminology

- **Neural networks consist of layers of nodes (neurons) for pattern recognition tasks.**

---

37. Multilayer Perceptrons (MLP) – Understanding Deep Learning Networks

- **A type of neural network with multiple layers, used in deep learning.**

---

38. Activation Functions: Sigmoid, ReLU, Tanh, Softmax

- **Functions that introduce non-linearity in neural networks:**
    - **Sigmoid: Outputs values between 0 and 1.**
    - **ReLU: Outputs 0 for negative inputs and the input itself for positive inputs.**
    - **Tanh: Outputs values between -1 and 1.**
    - **Softmax: Converts output into probability distribution.**