**Data Science Interview Question Part 1**

1. **How can you define Machine Learning?**  
   Machine Learning is a subset of artificial intelligence (AI) that focuses on creating systems that learn and improve from experience without explicit programming. It involves algorithms that identify patterns and make predictions or decisions based on data.
2. **What do you understand by a labeled training dataset?**  
   A labeled training dataset consists of input data paired with the corresponding correct output (label). For example, in an image classification task, each image (input) is associated with a label, like "cat" or "dog." These labels are used to train supervised learning models.
3. **What are the two most common supervised ML tasks you have performed so far?**

**Classification:** Predicting discrete labels, such as spam detection in emails or fraud detection in transactions.

**Regression:** Predicting continuous values, such as house price prediction based on features like area and location.

1. **What kind of Machine Learning algorithm would you use to make a robot walk in various unknown areas?**  
   Reinforcement Learning algorithms are best suited for this task. These algorithms learn by interacting with the environment and optimizing behavior based on feedback (rewards or penalties).
2. **What kind of ML algorithm can you use to segment your users into multiple groups?**  
   Clustering algorithms like K-Means, DBSCAN, or Hierarchical Clustering can be used for user segmentation in unsupervised learning tasks.
3. **What type of learning algorithm relies on similarity measures to make predictions?**  
   Instance-based learning algorithms like k-Nearest Neighbors (k-NN) use similarity measures (e.g., Euclidean distance) to predict outputs based on the most similar examples in the training data.
4. **What is an online learning system?**  
   An online learning system updates its model incrementally as new data becomes available, rather than training on the entire dataset at once. This is particularly useful for dynamic environments and streaming data.
5. **What is out-of-core learning?**  
   Out-of-core learning involves training machine learning models on datasets that are too large to fit into memory. The data is processed in smaller batches that are sequentially loaded into memory.
6. **Can you name a couple of ML challenges that you have faced?**
   1. **Data Imbalance:** Handling datasets where one class is significantly underrepresented compared to others.
   2. **Overfitting:** Preventing the model from performing well on training data but poorly on unseen data.
7. **Can you please give one example of hyperparameter tuning with respect to some classification algorithm?**  
   For a Support Vector Machine (SVM) classifier, tuning the **kernel type** (e.g., linear, polynomial, RBF) and **C** (regularization parameter) are examples of hyperparameter tuning. Grid search or random search can be used to find the optimal combination of these parameters
8. **What is out-of-bag evaluation?**  
   Out-of-bag (OOB) evaluation is a technique used in ensemble methods like Random Forest. During training, each tree is trained on a bootstrap sample (a random subset of the training data with replacement). The remaining data, not included in the bootstrap sample, is called the "out-of-bag" data. OOB evaluation tests the model on this data, providing an unbiased estimate of the model's performance without needing a separate validation set.
9. **What do you understand by hard and soft voting classifiers?**

**Hard Voting Classifier:** Aggregates the predictions of multiple models and selects the majority class as the final prediction.

**Soft Voting Classifier:** Uses the predicted probabilities from each model, averages them, and selects the class with the highest average probability.

1. **How will you bring down a training time of 5 minutes to 5 seconds?**  
   You can reduce training time by:
   1. Using **Distributed Computation:** Train the model across multiple machines or GPUs using frameworks like TensorFlow or PyTorch.
   2. **Dimensionality Reduction:** Reduce the number of features with PCA or feature selection techniques.
   3. **Smaller Dataset:** Use a subset of the data or implement mini-batch training.
   4. **Algorithm Optimization:** Choose a faster algorithm or tweak hyperparameters for quicker convergence.
2. **Can you combine 5 different models with 95% precision to get better results?**  
   Yes, combining models can improve results through ensemble methods. For example:
   1. **Bagging:** Combine models by averaging predictions (e.g., Random Forest).
   2. **Boosting:** Train models sequentially, focusing on errors made by previous ones (e.g., XGBoost).
   3. **Stacking:** Use predictions from these models as inputs to a meta-model.  
      If all models are highly correlated, however, improvement might be negligible.
3. **What is Gradient Descent? How will you explain Gradient Descent to a kid?**  
   Gradient Descent is an optimization algorithm used to minimize errors in a model by iteratively adjusting parameters.

**To a kid:** Imagine you're trying to roll a ball down a hill to reach the lowest point (goal). You take small steps in the direction where the ground slopes downward, adjusting your steps based on how steep the slope is.

1. **Can you explain the difference between regression and classification?**
   1. **Regression:** Predicts continuous values, e.g., predicting house prices.
   2. **Classification:** Predicts discrete labels, e.g., whether an email is spam or not.
2. **Explain a clustering algorithm of your choice.**  
   **K-Means Clustering:**
3. Assigns data points to KKK clusters by minimizing the distance between points and the cluster centroids.
4. Steps:
5. Initialize KKK centroids randomly.
6. Assign each point to the nearest centroid.
7. Update centroids as the mean of points in each cluster.
8. Repeat until convergence.
9. **How can you explain ML, DL, NLP, Computer Vision, and Reinforcement Learning with examples?**
   1. **Machine Learning (ML):** Using algorithms to identify patterns (e.g., predicting loan approvals).
   2. **Deep Learning (DL):** A subset of ML with neural networks (e.g., detecting objects in images).
   3. **Natural Language Processing (NLP):** Understanding and generating text (e.g., chatbots, translation).
   4. **Computer Vision:** Extracting information from images (e.g., face recognition).
   5. **Reinforcement Learning:** Learning through trial and error (e.g., training a robot to play chess).
10. **How can you explain semi-supervised ML with an example?**  
    Semi-supervised ML uses a small amount of labeled data and a large amount of unlabeled data.

**Example:** Imagine a teacher gives you a few labeled math problems and many unlabeled ones. You solve the labeled ones first, then use patterns to tackle the unlabeled ones.

1. **What is the difference between abstraction and generalization in your own words?**
2. **Abstraction:** Simplifying complex concepts by focusing on relevant details while hiding unnecessary specifics. (e.g., A car as a means of transport without mentioning its engine).
3. **Generalization:** Making broad assumptions or models that work for various scenarios. (e.g., Using past learning to predict outcomes in new situations).

**21. What steps did you follow in your last project to prepare the dataset?**

* **Data Collection:** Gathered raw data from multiple sources (e.g., APIs, databases, or CSV files).
* **Data Cleaning:** Handled missing values, removed duplicates, and fixed inconsistencies.
* **Feature Selection:** Identified relevant features using statistical tests or domain knowledge.
* **Encoding Categorical Variables:** Transformed categorical data into numeric form using one-hot encoding or label encoding.
* **Normalization/Standardization:** Scaled numeric features to ensure consistent ranges.
* **Data Splitting:** Divided the dataset into training, validation, and test sets.

**22. What steps were involved in the model selection procedure in your last project?**

* **Understand Problem Requirements:** Defined whether it was a classification, regression, or clustering task.
* **Baseline Model:** Started with simple models to establish a baseline (e.g., Logistic Regression for classification).
* **Algorithm Selection:** Experimented with various algorithms like Random Forest, SVM, and XGBoost.
* **Evaluation Metrics:** Selected metrics relevant to the problem (e.g., accuracy, F1-score, or RMSE).
* **Hyperparameter Tuning:** Used techniques like grid search or random search to optimize performance.
* **Cross-Validation:** Validated model consistency using k-fold cross-validation.
* **Ensemble Methods:** Combined models if necessary to improve performance.

**23. Steps to check the relationship between two columns:**

* **Visual Inspection:** Use scatterplots (for numerical data) or bar plots (for categorical data).
* **Correlation Analysis:** Compute correlation coefficients like Pearson or Spearman for numerical columns.
* **Chi-Square Test:** For categorical columns, test independence.
* **ANOVA or t-test:** Compare means if one column is numerical and the other categorical.
* **Regression Analysis:** Fit a simple regression model if looking for predictive relationships.

**24. Five strategies to handle missing values in a dataset:**

1. **Remove Rows/Columns:** Delete rows or columns with a high percentage of missing values.
2. **Impute with Mean/Median/Mode:** Replace missing values with statistical measures.
3. **Predict Missing Values:** Use a model to predict missing values based on other features.
4. **Use Domain Knowledge:** Replace missing values with relevant values from domain expertise.
5. **Add a Missing Indicator:** Add a binary feature indicating whether the value was missing.

**25. Five common issues with raw data:**

1. **Missing Values:** Unavailable entries in some features.
2. **Outliers:** Extreme values that distort model training.
3. **Inconsistent Data Formats:** Dates, currencies, or numerical values formatted differently.
4. **Duplicate Entries:** Repeated rows or records.
5. **Class Imbalance:** One class significantly outnumbers others in classification tasks.
6. **26. Strategy to handle categorical datasets:**

* **Label Encoding:** Convert categories to numeric labels for ordinal data.
* **One-Hot Encoding:** Create binary columns for each category (e.g., "red," "blue," "green" → [1, 0, 0]).
* **Frequency Encoding:** Replace categories with their frequency of occurrence.
* **Target Encoding:** Encode categories using the mean of the target variable.

**Example:** For a "Color" column with values ["Red", "Blue", "Green"], one-hot encoding would create three columns:

* Red: [1, 0, 0], Blue: [0, 1, 0], Green: [0, 0, 1].

**27. How do you define a model in machine learning in your own words?**  
A model is a mathematical representation of a real-world process that makes predictions or decisions based on input data. It learns patterns during training to generalize and apply them to unseen data.

**28. What is k-fold cross-validation, and when have you used it?**  
K-fold cross-validation divides the dataset into kkk equally sized subsets (folds). The model is trained on k−1k-1k−1 folds and tested on the remaining fold. This process is repeated kkk times, each time using a different fold for testing. It ensures the model's performance is robust across the dataset.

**Use case:** When dataset size is small and requires a reliable performance estimate.

**29. What is bootstrap sampling in your own words?**  
Bootstrap sampling involves creating multiple datasets by randomly sampling with replacement from the original dataset. Each bootstrap sample may contain duplicate records. This technique is used to estimate model stability and accuracy.

**30. What do you understand by underfitting and overfitting? Provide examples.**

* **Underfitting:** The model is too simple to capture patterns in the data, leading to poor performance on both training and test sets.
  + **Example:** Using a linear model to fit a dataset with non-linear relationships.
* **Overfitting:** The model is too complex and captures noise in the data, performing well on training data but poorly on test data.
  + **Example:** A deep decision tree that memorizes the training data instead of generalizing.

**31. What is the difference between cross-validation and bootstrapping?**

* **Cross-Validation:** Splits the dataset into folds for training and testing. Each fold is used once for testing, ensuring no data point is used for both simultaneously. Used to estimate the model’s generalization error.
* **Bootstrapping:** Creates multiple datasets by sampling with replacement. Some points may appear multiple times in a sample, while others may be excluded. Primarily used to estimate the stability or confidence intervals of statistics.

**32. What do you understand by the silhouette coefficient?**  
The silhouette coefficient measures how well a data point fits within its assigned cluster compared to other clusters.

* Values range from −1-1−1 to 111:
  + **1:** Perfectly assigned.
  + **0:** On or near the boundary between clusters.
  + **-1:** Incorrectly assigned to the wrong cluster.  
    It is computed as:  
    Silhouette Coefficient=b−amax⁡(a,b)\text{Silhouette Coefficient} = \frac{b - a}{\max(a, b)}Silhouette Coefficient=max(a,b)b−a​  
    Where aaa is the mean intra-cluster distance and bbb is the mean nearest-cluster distance.

**33. What is the advantage of using the ROC score?**  
The ROC score (Area Under the Curve - AUC) evaluates the performance of a binary classification model, considering both true positive and false positive rates across different thresholds.

**Advantages:**

* Provides a threshold-independent metric.
* Useful for comparing models, especially with imbalanced datasets.
* A higher AUC indicates better model performance.

**34. Complete approach to evaluate a regression model:**

1. **Split Data:** Use train-test split or cross-validation for unbiased evaluation.
2. **Metrics:** Choose evaluation metrics like RMSE, MAE, or R2R^2R2 based on the problem.
3. **Residual Analysis:** Plot residuals to check patterns and ensure errors are randomly distributed.
4. **Correlation Analysis:** Compare predicted and actual values using scatterplots.
5. **Validation Set:** Evaluate on a separate validation set to check overfitting.
6. **Error Analysis:** Investigate outliers or high-error cases for improvement.

**35. Example of lazy and eager learner algorithms:**

* **Lazy Learner:** k-Nearest Neighbors (k-NN), which stores the entire dataset and makes predictions during inference.
* **Eager Learner:** Decision Trees, Support Vector Machines (SVM), which build a model during training and use it for predictions.

**36. What do you understand by the holdout method?**  
The holdout method splits the dataset into separate subsets, typically training, validation, and test sets. The model is trained on the training set, tuned on the validation set, and evaluated on the test set. It is a simple method to estimate model performance.

**37. Difference between predictive modeling and descriptive modeling:**

* **Predictive Modeling:** Focuses on predicting future outcomes or values based on patterns in data. Example: Predicting sales for next month.
* **Descriptive Modeling:** Focuses on summarizing or understanding the patterns and relationships within the data. Example: Clustering customers based on buying behavior.

**38. How did you derive a feature for model building in your last project?**  
Derived new features using domain knowledge and data analysis. For example:

* **Feature Engineering:** Combined “transaction time” and “location” to create a “rush hour transaction” indicator for a fraud detection project.
* **Statistical Insights:** Used aggregate statistics like mean or count of features for grouping.

**39. Five different encoding techniques:**

1. **One-Hot Encoding:** Converts categories into binary columns.
2. **Label Encoding:** Assigns unique integers to each category.
3. **Frequency Encoding:** Replaces categories with their frequency of occurrence.
4. **Target Encoding:** Uses the mean of the target variable for each category.
5. **Binary Encoding:** Combines label and binary representations to reduce dimensionality.

**40. How do you define features as not important, and what strategy will you follow?**  
**Defining Unimportant Features:**

* Features with low correlation to the target variable.
* Features with high multicollinearity (highly correlated with other features).
* Features consistently assigned low importance scores by models (e.g., using feature importance in Random Forest).

**Strategies to Handle Unimportant Features:**

1. Use feature selection methods like Recursive Feature Elimination (RFE) or Lasso regression.
2. Drop low-variance features.
3. Analyze SHAP or permutation importance for deeper insights.

**41. What is the difference between Euclidean distance and Manhattan distance?**

* **Euclidean Distance:** Measures the straight-line distance between two points in space. It is based on the Pythagorean theorem.
  + Formula:  
    d=∑(xi−yi)2d = \sqrt{\sum (x\_i - y\_i)^2}d=∑(xi​−yi​)2​
  + Example: The distance "as the crow flies" between two locations on a map.
* **Manhattan Distance:** Measures the distance between two points by summing the absolute differences along each axis.
  + Formula:  
    d=∑∣xi−yi∣d = \sum |x\_i - y\_i|d=∑∣xi​−yi​∣
  + Example: The distance you would walk along city blocks (grid-like paths).

**42. What do you understand by feature selection, transformation, engineering, and EDA?**

1. **Feature Selection:** Identifying the most relevant features for a model to improve performance.
   * Example: Selecting "Age" and "Income" as predictors for loan approval.
   * Methods: Correlation analysis, Recursive Feature Elimination (RFE), or Lasso regression.
2. **Feature Transformation:** Modifying features to make them more useful for modeling.
   * Example: Scaling numerical values with StandardScaler or log-transforming skewed data.
3. **Feature Engineering:** Creating new features based on domain knowledge or existing data.
   * Example: Combining "Date" and "Time" into "IsWeekend" for fraud detection.
4. **Exploratory Data Analysis (EDA):** Understanding data by visualizing patterns, distributions, and relationships.
   * Example: Using histograms for data distribution or scatter plots for relationships.

**Steps in Each:**

* **Feature Selection:** Removed irrelevant columns (e.g., IDs).
* **Transformation:** Standardized numerical data, applied log transformations to highly skewed features.
* **Engineering:** Created derived features like "Total Transaction Amount" for financial data.
* **EDA:** Identified outliers and missing patterns using boxplots and heatmaps.

**43. Difference between SVD and PCA:**

* **SVD (Singular Value Decomposition):**
  + Decomposes a matrix into three matrices: A=UΣVTA = U \Sigma V^TA=UΣVT.
  + Works directly on any m×nm \times nm×n matrix.
  + General-purpose matrix factorization technique.
* **PCA (Principal Component Analysis):**
  + A dimensionality reduction technique that identifies principal components (new orthogonal axes).
  + Uses SVD under the hood to compute principal components.
  + Specifically designed for finding patterns in data.

**44. Feature transformations in your last project:**

* Normalized numerical features using Min-Max scaling.
* Applied log transformation to reduce the skewness of monetary transaction data.
* Encoded categorical variables using one-hot encoding and target encoding.
* Derived time-based features like "Hour of Day" from timestamp data.

**45. Have you used external features from 3rd-party data?**  
Yes. For example:

* In a fraud detection project, integrated location-based data from a third-party API to identify high-risk regions.
* Example: "Zip Code Risk Index" was fetched via API and merged with transaction records to enrich the dataset.

**46. If your model is overfitted, what will you do next?**

* Use regularization techniques like L1 (Lasso) or L2 (Ridge).
* Reduce model complexity (e.g., pruning decision trees, reducing layers in deep networks).
* Increase training data through augmentation or collection.
* Apply dropout in neural networks.
* Use cross-validation to monitor overfitting during training.

**47. Explain bias-variance trade-off:**

* **Bias:** Error due to overly simple models that underfit the data.
* **Variance:** Error due to overly complex models that overfit the data.  
  The trade-off involves finding a balance where the model generalizes well without underfitting or overfitting.

**48. Steps to improve model accuracy (5 approaches):**

1. **Feature Engineering:** Derive new, meaningful features to enhance the data’s predictive power.
   * Justification: Captures domain-specific insights.
2. **Hyperparameter Tuning:** Use grid search or Bayesian optimization.
   * Justification: Fine-tunes the model for better performance.
3. **Ensemble Methods:** Combine multiple models (e.g., bagging, boosting).
   * Justification: Reduces error by averaging predictions.
4. **Data Augmentation:** Expand the dataset using transformations.
   * Justification: Helps generalize in scenarios with limited data.
5. **Regularization:** Use techniques like L1, L2, or dropout.
   * Justification: Prevents overfitting.

**49. Feature engineering for text categorization:**

* Convert text into numerical form using techniques like TF-IDF or word embeddings.
* Remove stop words, perform stemming/lemmatization, and extract n-grams.
* Add metadata features like word count, sentence count, or average word length.

**50. Explain vectorization and Hamming distance:**

* **Vectorization:** Converting data into numerical vectors for machine learning. Example: Text data is vectorized using bag-of-words or word embeddings.
* **Hamming Distance:** Measures the number of positions at which corresponding elements differ in two binary strings or vectors.
  + Example:
    - String1: "10101"
    - String2: "10011"
    - Hamming Distance: 2 (differences at positions 2 and 4).

**PART - 2**

**1. Can you please explain chain rule and its use?**

* The **chain rule** is a fundamental concept in calculus used to compute the derivative of a composite function.
  + Formula: ddxf(g(x))=f′(g(x))⋅g′(x)\frac{d}{dx}f(g(x)) = f'(g(x)) \cdot g'(x)dxd​f(g(x))=f′(g(x))⋅g′(x)
* **Use in Machine Learning:**
  + Used in **backpropagation** for training neural networks.
  + Computes gradients of loss functions with respect to weights by chaining derivatives layer by layer.

**2. What is the difference between correlation and covariance?**

* **Covariance:** Measures how two variables vary together. It can take any value.
  + Example: If two variables increase together, covariance is positive.
* **Correlation:** Standardized version of covariance, bounded between −1-1−1 and 111.
  + Indicates both strength and direction of the linear relationship.

**3. Sampling techniques used in your project:**

1. **Random Sampling:** Selecting random data points.
2. **Stratified Sampling:** Ensuring class proportions are maintained (used for imbalanced datasets).
3. **Oversampling (SMOTE):** Generating synthetic data for minority classes.
4. **Undersampling:** Reducing data points from the majority class to balance datasets.

**4. Have you ever used hypothesis testing in your project?**  
Yes.

* Example: In a fraud detection project, we used hypothesis testing to verify if transaction amounts differ significantly between fraudulent and non-fraudulent transactions.
* Test used: **t-test** to compare means between the two groups.

**5. When to use Naïve Bayes and Decision Tree separately?**

* **Naïve Bayes:**
  + When data is categorical and features are independent.
  + Example: Spam classification.
* **Decision Tree:**
  + Handles both categorical and numerical data, even if features are dependent.
  + Example: Loan approval prediction.

**6. Advantages and disadvantages of Naïve Bayes:**

* **Advantages:**
  + Fast and efficient for high-dimensional data.
  + Works well with small datasets.
* **Disadvantages:**
  + Assumes feature independence, which is rarely true in real-world data.
  + Performs poorly with highly correlated features.

**7. Naïve Bayes classification equation for numerical data:**  
For continuous data, it uses the Gaussian distribution:

P(x∣Ck)=12πσk2e−(x−μk)22σk2P(x|C\_k) = \frac{1}{\sqrt{2\pi\sigma\_k^2}} e^{-\frac{(x - \mu\_k)^2}{2\sigma\_k^2}}P(x∣Ck​)=2πσk2​​1​e−2σk2​(x−μk​)2​

Where μk\mu\_kμk​ and σk\sigma\_kσk​ are the mean and variance of the feature for class CkC\_kCk​.

**8. Scenario for using a boosting classifier and regressor:**

* **Classifier:** Predicting fraudulent transactions in imbalanced datasets.
* **Regressor:** Predicting house prices where errors need to be minimized.

**9. What does Bayesian classifier try to learn?**

* Tries to learn the **posterior probability** P(C∣X)P(C|X)P(C∣X): the probability of a class CCC given data XXX.
* Procedure: Uses Bayes' theorem: P(C∣X)=P(X∣C)P(C)P(X)P(C|X) = \frac{P(X|C)P(C)}{P(X)}P(C∣X)=P(X)P(X∣C)P(C)​

**10. When to use SVM instead of Logistic Regression?**

* Use **SVM** when data is not linearly separable or requires a decision boundary with higher dimensions using kernels.
* Example: Classifying images with complex patterns.

**11. What do you understand by RBF kernel in SVM?**

* **Radial Basis Function (RBF):** Maps input data into higher dimensions to handle non-linear problems.
* Formula: K(x,x′)=e−γ∥x−x′∥2K(x, x') = e^{-\gamma \|x - x'\|^2}K(x,x′)=e−γ∥x−x′∥2

**12. Two scenarios where AI can increase revenue in travel industry:**

1. **Dynamic Pricing:** Using AI to adjust ticket prices based on demand and competition.
2. **Personalized Recommendations:** AI suggesting tailored travel packages to customers.

**13. What is a leaf node in a decision tree?**

* A leaf node represents the final output of a decision tree, containing the class label or predicted value.

**14. What is information gain and entropy in decision tree?**

* **Entropy:** Measures the impurity or randomness in data. H=−∑pilog⁡2piH = -\sum p\_i \log\_2 p\_iH=−∑pi​log2​pi​
* **Information Gain:** Reduction in entropy after a split. IG=Hparent−Weighted average of HchildrenIG = H\_{parent} - \text{Weighted average of } H\_{children}IG=Hparent​−Weighted average of Hchildren​

**15. Disadvantages of decision tree:**

* Prone to overfitting.
* Sensitive to noisy data.
* Creates biased trees with imbalanced datasets.

**16. Features of Random Forest:**

* Combines multiple decision trees for better accuracy.
* Reduces overfitting using bagging.
* Handles missing and imbalanced data effectively.

**17. How to avoid overfitting in decision trees?**

* Prune the tree.
* Set maximum depth.
* Use minimum samples for splits.
* Use ensemble methods like Random Forest.

**18. Explain polynomial regression in simple terms:**  
Polynomial regression fits a polynomial curve to data points instead of a straight line, capturing non-linear relationships.

**19. Learning mechanism of linear regression:**  
Linear regression learns coefficients θ\thetaθ by minimizing the cost function (Mean Squared Error) to best fit the line to data.

**20. Cost function in logistic regression:**  
The cost function is the **log-loss**:

J(θ)=−1m∑[ylog⁡(hθ(x))+(1−y)log⁡(1−hθ(x))]J(\theta) = -\frac{1}{m} \sum \left[ y \log(h\_\theta(x)) + (1-y) \log(1-h\_\theta(x)) \right]J(θ)=−m1​∑[ylog(hθ​(x))+(1−y)log(1−hθ​(x))]

**21. Error function in linear regression:**  
The error function is the **Mean Squared Error (MSE)**:

MSE=1n∑(yi−y^i)2MSE = \frac{1}{n} \sum (y\_i - \hat{y}\_i)^2MSE=n1​∑(yi​−y^​i​)2

**22. Use of OLS technique in dataset:**

* **OLS (Ordinary Least Squares):** Minimizes the sum of squared differences between observed and predicted values.
* Use: To determine the best-fitting line in regression tasks.

**23. Explain dendrogram in your own way:**

* A **dendrogram** is a tree-like diagram that visualizes the hierarchy of clusters formed during hierarchical clustering.
* Nodes represent clusters, and the height of the branches represents the distance or dissimilarity between clusters.
* It's used to decide the optimal number of clusters by "cutting" the tree at a certain level.

**24. How do you measure the quality of clusters in DBSCAN?**

* **Silhouette Coefficient:** Measures how well-separated the clusters are.
* **Core Points Ratio:** Checks the proportion of points classified as core points.
* **Cluster Density:** Analyzes the density of points within clusters.

**25. How do you evaluate DBSCAN algorithm?**

* Evaluate using metrics like **Adjusted Rand Index (ARI)**, **Homogeneity**, or **Completeness Score**.
* Visualize the clusters if the data is 2D or 3D.
* Compare results to ground truth (if available) or assess how well it handles noise and outliers.

**26. What do you understand by market basket analysis?**

* Market Basket Analysis uses **association rule mining** to discover relationships between items purchased together.
* Example: If a customer buys bread, they are likely to buy butter (rule: bread → butter).

**27. Explain centroid formation technique in K-Means algorithm:**

* Initial centroids are randomly chosen or selected using **k-means++**.
* The algorithm assigns points to the nearest centroid and recalculates centroids as the mean of points in each cluster.
* This process repeats until convergence.

**28. Have you ever used SVM regression in any project? If yes, why?**  
Yes, for projects requiring robust performance on small datasets or when dealing with outliers, as **SVM regression** minimizes the error margin (epsilon).

**29. Explain the concept of GINI Impurity:**

* **GINI Impurity** measures the likelihood of incorrect classification at a node in a decision tree:

G=1−∑(pi)2G = 1 - \sum (p\_i)^2G=1−∑(pi​)2

* Lower values indicate purer nodes.

**30. How to control the growth of a decision tree with 100 columns?**

* Use **max\_depth** to limit tree depth.
* Set **min\_samples\_split** and **min\_samples\_leaf** to avoid over-splitting.
* Perform feature selection to reduce the number of input features.

**31. AdaBoost underfitting hyperparameter tuning:**

* Increase the number of weak learners (**n\_estimators**).
* Decrease the regularization parameter (**learning\_rate**).
* Switch to more complex base learners, e.g., decision trees with more depth.

**32. Explain gradient boosting algorithm:**

* Gradient Boosting is an ensemble method that builds weak learners sequentially to minimize the loss function.
* Each new model corrects the errors of the previous ones by focusing on poorly predicted instances.

**33. Can we use PCA to reduce dimensionality of highly non-linear data?**

* PCA assumes linear relationships. For non-linear data, use **Kernel PCA** or **t-SNE**.

**34. How do you evaluate performance of PCA?**

* Use **explained variance ratio** to determine how much variance is captured by principal components.
* Rebuild models with reduced features and compare performance metrics.

**35. Have you used multiple dimensionality techniques?**

* Yes, PCA for reducing features and **t-SNE** for visualizing high-dimensional data.

**36. Curse of dimensionality explained with example:**

* With many features, data points become sparse, and distances lose meaning.
* Example: In high-dimensional space, nearest neighbors may not be "close" anymore, affecting algorithms like k-NN.

**37. Difference between anomaly detection and novelty detection:**

* **Anomaly detection:** Identifies unusual patterns in seen data.
* **Novelty detection:** Identifies completely unseen or new patterns.

**38. Explain Gaussian Mixture Model (GMM):**

* GMM is a probabilistic clustering algorithm that models data as a combination of Gaussian distributions.
* Each cluster is represented by a Gaussian with mean and variance.

**39. List of 10 activation functions with explanations:**

1. **Sigmoid:** Maps values to (0, 1). Used for binary classification.
2. **Tanh:** Maps values to (-1, 1). Better for zero-centered data.
3. **ReLU:** Outputs positive values; zero otherwise. Efficient and widely used.
4. **Leaky ReLU:** Handles negative values with a small slope.
5. **Softmax:** Converts outputs into probabilities for multiclass classification.
6. **Swish:** Combines ReLU and Sigmoid for smooth transitions.
7. **ELU:** Similar to ReLU but smooth for negative inputs.
8. **Maxout:** Adapts to data shape by selecting the maximum value.
9. **Linear:** Outputs input as-is; used in regression tasks.
10. **GELU:** Smooth curve for better gradient flow.

**40. Explain neural network in terms of mathematical function:**

* A neural network is a composite function:

f(x)=f(L)(f(L−1)(...f(1)(x)))f(x) = f^{(L)}(f^{(L-1)}(...f^{(1)}(x)))f(x)=f(L)(f(L−1)(...f(1)(x)))

* Each layer applies a transformation:

z(l)=W(l)a(l−1)+b(l)z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}z(l)=W(l)a(l−1)+b(l)

**41. Correlate biological and artificial neuron:**

* Biological neuron: Dendrites (inputs), Cell body (processing), Axon (output).
* Artificial neuron: Weights (inputs), Activation function (processing), Output layer (output).

**42. List of cost functions:**

1. **Mean Squared Error (MSE):** Regression.
2. **Mean Absolute Error (MAE):** Regression.
3. **Log Loss:** Classification.
4. **Hinge Loss:** SVMs.
5. **Cross-Entropy Loss:** Multiclass classification.

**43. Can I solve classification problems with tabular data in neural networks?**

* Yes, by preprocessing data and using dense layers.

**44. What is backpropagation?**

* A method for updating neural network weights by propagating errors backward to minimize the cost function.

**45. Why neural networks instead of equations?**

* Neural networks can model complex non-linear relationships.

**46. Weight initialization techniques used:**

1. **Random Initialization.**
2. **Xavier Initialization.**
3. **He Initialization.**

**47. Visualizing a neural network:**

* Use tools like **TensorBoard** or **Netron**.

**48. Explain training of neural networks:**

* Forward pass: Compute outputs.
* Backward pass: Compute gradients.
* Update weights using optimizers.

**49. Difference between sigmoid and tanh functions:**

* **Sigmoid:** Outputs between (0, 1).
* **Tanh:** Outputs between (-1, 1), better for zero-centered data.

**50. Disadvantage of ReLU function:**

* Can cause **dead neurons** when gradients become zero for negative inputs.

**PART-3**

**1. How do you select the number of layers and neurons in a neural network?**

* **Start small, experiment, and tune based on performance.**
* **Rules of thumb:**
  + **For simple problems, use 1-2 layers.**
  + **For complex data, increase layers gradually.**
  + **Number of neurons: Equal to or slightly more than input features in hidden layers.**

**2. Have you ever designed a Neural Network architecture by yourself?  
Yes, by defining the number of layers, types of layers (e.g., convolutional, dense), activation functions, and optimizers tailored to the dataset and problem.**

**3. Can you please explain SWISS Function?  
The Swish function is an activation function defined as:**

**f(x)=x⋅sigmoid(x)=x⋅11+e−xf(x) = x \cdot \text{sigmoid}(x) = x \cdot \frac{1}{1 + e^{-x}}f(x)=x⋅sigmoid(x)=x⋅1+e−x1​**

**It provides smooth, non-monotonic behavior, often outperforming ReLU in deep networks.**

**4. What is the learning rate in layman's terms? How do you control it?**

* **Learning rate determines how much to adjust weights in each step.**
* **Too high → unstable training. Too low → slow convergence.**
* **Control it with learning rate schedules (e.g., step decay, exponential decay) or optimizers like Adam that adapt the rate.**

**5. Difference between batch, minibatch, and stochastic gradient descent (SGD):**

* **Batch Gradient Descent: Uses the entire dataset for each step.**
* **Mini-batch Gradient Descent: Uses a subset of data. Faster, balances memory and speed.**
* **Stochastic Gradient Descent (SGD): Uses one sample at a time, noisy but fast.**

**6. What is batch size while training Neural Network with example?**

* **The batch size is the number of training samples processed before updating weights.**
* **Example: Dataset = 1000 samples, Batch size = 100 → 10 updates per epoch.**

**7. Five best optimizers with mathematical explanation:**

1. **SGD: wt+1=wt−η∇L(wt)w\_{t+1} = w\_t - \eta \nabla L(w\_t)wt+1​=wt​−η∇L(wt​)**
2. **Momentum: Adds velocity to updates: vt=γvt−1+η∇L(wt)v\_t = \gamma v\_{t-1} + \eta \nabla L(w\_t)vt​=γvt−1​+η∇L(wt​)**
3. **Adam: Combines momentum and RMSProp: wt+1=wt−ηm^tv^t+ϵw\_{t+1} = w\_t - \eta \frac{\hat{m}\_t}{\sqrt{\hat{v}\_t} + \epsilon}wt+1​=wt​−ηv^t​​+ϵm^t​​**
4. **Adagrad: Adapts learning rate per parameter.**
5. **RMSProp: Normalizes gradients by recent magnitudes.**

**8. Can you build Neural Network without using any library? If yes, prove it.  
Yes, by implementing matrix multiplication, activation functions, and gradient descent using Python or similar tools.**

**9. What is the use of biases in a neural network?  
Bias allows shifting the activation function, enabling the model to fit data more flexibly.**

**10. Hyperparameter tuning for neural networks:**

* **Use Grid Search, Random Search, or Bayesian Optimization.**
* **Tune parameters like learning rate, batch size, layers, neurons, dropout rates.**

**11. Regularization used in neural networks:**

1. **L1/L2 Regularization (Weight Decay).**
2. **Dropout: Randomly disables neurons during training.**
3. **Early Stopping: Stops training when validation performance degrades.**

**12. Libraries used for neural network implementation:**

* **TensorFlow, PyTorch, Keras, MXNet, Theano, ONNX.**

**13. What is a custom layer and a custom model?**

* **Custom Layer: A user-defined layer implementing specific behavior.**
* **Custom Model: Combines layers in a novel architecture.**

**14. Differentiation using TensorFlow/PyTorch:**

* **TensorFlow: Use tf.GradientTape for automatic differentiation.**
* **PyTorch: Use torch.autograd with backward() for gradients.**

**15. What is an epoch in simple terms?  
An epoch is one complete pass through the entire training dataset.**

**16. What is a TensorFlow record?  
A TFRecord is a file format optimized for storing and loading large datasets efficiently.**

**17. Data augmentation in deep learning:**

* **Techniques: Flipping, rotation, scaling, cropping, adding noise, color jittering.**
* **Tools: TensorFlow ImageDataGenerator, PyTorch transforms.**

**18. List of CNN networks heard of:**

* **LeNet, AlexNet, VGGNet, ResNet, DenseNet, MobileNet, EfficientNet.**

**19. Object detection algorithms known:**

* **YOLO, SSD, Faster R-CNN, RetinaNet, Mask R-CNN.**

**20. Difference between object detection and classification:**

* **Object Detection: Identifies and locates multiple objects in an image (bounding boxes + labels).**
* **Classification: Assigns a single label to an entire image.**

**21. Major tasks performed in CNN:**

1. **Feature Extraction: Automatically extracts features (e.g., edges, textures) using convolutional layers.**
2. **Object Detection: Identifies and localizes objects in images.**
3. **Image Classification: Assigns labels to images.**
4. **Segmentation: Divides an image into meaningful regions.**
5. **Feature Map Generation: Represents intermediate data in CNN layers.**

**22. Algorithms for segmentation:**

1. **U-Net**
2. **Mask R-CNN**
3. **DeepLab**
4. **PSPNet (Pyramid Scene Parsing Network)**
5. **FCN (Fully Convolutional Networks)**

**23. Algorithm for tracking a football in a match:**

* **YOLO (You Only Look Once) or Deep SORT (Simple Online and Realtime Tracking) for real-time tracking.**

**24. Algorithm for identifying objects in satellite image data:**

* **UNet or DeepLab for segmentation.**
* **ResNet or Inception for classification.**

**25. Algorithm for PCB fault detection:**

* **Convolutional Neural Networks (CNNs) tailored for defect detection.**
* **Models like YOLO or Mask R-CNN can be adapted for fault localization.**

**26. What is a pretrained model?  
A pretrained model is a model trained on a large dataset (e.g., ImageNet) and used for similar tasks on smaller datasets via transfer learning.**

**27. Types of transfer learning:**

1. **Feature Extraction: Use pretrained layers as fixed feature extractors.**
2. **Fine-Tuning: Adjust pretrained weights for the target task.**
3. **Domain Adaptation: Modify the model for different but related domains.**

**28. CNN vs. RNN Failure Cases:**

* **CNN fails with sequential or temporal data (e.g., speech, text).**
* **RNNs succeed in tasks like time series prediction, language modeling, or speech-to-text conversion.**

**29. GPU for training object detection models:  
Common GPUs: NVIDIA RTX 3080, RTX 3090, or Tesla V100.**

**30. Dataset size, epochs, time, and accuracy:  
This depends on the problem. Example:**

* **Dataset: 10,000 images**
* **Epochs: 50**
* **Time: 10 hours on GPU**
* **Accuracy: ~92%**

**31. Optimization for object detection model:**

1. **Use pretrained weights for faster convergence.**
2. **Adjust learning rates dynamically.**
3. **Use data augmentation to improve generalization.**
4. **Reduce model size using quantization or pruning.**

**32. Evaluate object detection models:**

* **Metrics:**
  + **mAP (mean Average Precision)**
  + **IoU (Intersection over Union)**
* **Tools: TensorBoard, COCO evaluation scripts.**

**33. Algorithms for object tracking:**

1. **Deep SORT**
2. **ByteTrack**
3. **SiamMask**
4. **CenterTrack**
5. **Kalman Filter**

**34. What is FPS (frames per second)?  
FPS measures the speed of a system to process frames, indicating real-time capability.**

**35. 2D vs. 3D Convolution:**

* **2D Convolution: Processes 2D spatial data (images).**
* **3D Convolution: Processes 3D data (videos, volumetric data).**

**36. What is batch normalization?  
A technique to normalize layer inputs for stable training and faster convergence.**

**37. Algorithm for handwriting detection:**

* **RNNs (e.g., LSTM) combined with CNNs.**
* **CRNN (Convolutional Recurrent Neural Network).**

**38. Explain SoftMax function:**

**SoftMax(xi)=exi∑j=1nexj\text{SoftMax}(x\_i) = \frac{e^{x\_i}}{\sum\_{j=1}^{n} e^{x\_j}}SoftMax(xi​)=∑j=1n​exj​exi​​**

**It converts logits into probabilities.**

**39. Disadvantage of RNN:**

1. **Vanishing/exploding gradients.**
2. **Difficulty in capturing long-term dependencies.**
3. **Computational inefficiency.**

**40. Examples of RNNs:**

1. **Vanilla RNN**
2. **LSTM**
3. **GRU**
4. **BiLSTM**
5. **Sequence-to-Sequence RNN**

**41. LSTM architecture, advantages, and disadvantages:**

* **Advantages: Handles long-term dependencies.**
* **Disadvantages: Computationally expensive.**

**42. BiLSTM architecture, advantages, and disadvantages:**

* **Advantages: Accesses past and future context.**
* **Disadvantages: Computationally intensive.**

**43. Stacked LSTM architecture, advantages, and disadvantages:**

* **Advantages: Captures hierarchical features.**
* **Disadvantages: Increased complexity.**

**44. What is TF-IDF?  
Term Frequency-Inverse Document Frequency is a statistic to evaluate the importance of words in a document relative to a collection.**

**45. Creating Word2Vec:  
Train using Gensim library or TensorFlow.**

**46. Vectorization techniques:**

1. **Bag of Words**
2. **TF-IDF**
3. **Word2Vec**
4. **GloVe**
5. **FastText**

**47. Difference between RNN and Encoder-Decoder:**

* **RNN: Processes sequential data.**
* **Encoder-Decoder: Transforms input sequence to another format (e.g., translation).**

**48. Attention mechanism:  
Highlights important parts of the input for better focus, widely used in NLP tasks.**

**49. "Attention is All You Need":  
It introduces the Transformer model, focusing entirely on attention without recurrence.**

**50. Multi-Headed Attention:  
Uses multiple attention mechanisms to learn different relationships in the data. It improves context capture in Transformers.**

**PART-4**

**1. How can you define Machine Learning?  
Answer:**  
Machine Learning (ML) is a field of artificial intelligence (AI) that enables systems to learn from data and improve their performance over time without being explicitly programmed. It involves using algorithms and statistical models to analyze and make predictions or decisions based on data.

**2. What do you understand by a Labelled Training Dataset?  
Answer:**  
A Labelled Training Dataset is a dataset where each input (or instance) is paired with a corresponding label or output. This label helps the model learn the relationship between the input and the desired output, enabling supervised learning tasks like classification and regression.

**3. What are the 2 most common supervised ML tasks you have performed so far?  
Answer:**  
The two most common supervised ML tasks are:

* Classification: Where the goal is to predict discrete labels (e.g., classifying emails as spam or not spam).
* Regression: Where the goal is to predict a continuous output (e.g., predicting house prices based on various features).

**4. What kind of Machine Learning algorithm would you use to walk a robot in various unknown areas?  
Answer:**  
Reinforcement Learning is the most suitable algorithm for this task. The robot can explore the environment, learn from its interactions, and improve its walking behavior by receiving rewards or penalties based on its actions.

**5. What kind of ML algorithm can you use to segment your user into multiple groups?  
Answer:**  
To segment users into multiple groups, Clustering algorithms such as K-means or Hierarchical Clustering can be used. These unsupervised algorithms group similar data points together based on their features.

**6. What type of learning algorithm relies on similarity measures to make a prediction?  
Answer:**  
The k-Nearest Neighbors (k-NN) algorithm relies on similarity measures (such as distance metrics) to classify or predict an output based on how similar a new input is to the training data.

**7. What is an online learning system?  
Answer:**  
An Online Learning System is a type of machine learning where the model is trained incrementally as new data becomes available, rather than being trained on the entire dataset at once. This is particularly useful for scenarios where data is constantly changing or arriving in real time.

**8. What is out-of-core learning?  
Answer:**  
Out-of-Core Learning refers to a method where the machine learning algorithm processes datasets that are too large to fit into memory by handling the data in smaller, manageable chunks or batches.

**9. Can you name a couple of ML challenges that you have faced?  
Answer:**Two common ML challenges are:

* Overfitting: Where the model performs well on training data but poorly on unseen data.
* Data Imbalance: When the dataset contains an unequal distribution of classes, leading to biased predictions.

**10. Can you provide an example of hyperparameter tuning with respect to a classification algorithm?  
Answer:**An example of hyperparameter tuning in a classification algorithm is tuning the C parameter in Support Vector Machines (SVM). The C parameter controls the trade-off between maximizing the margin and minimizing classification errors. A lower C value allows for a larger margin but may result in more misclassifications, while a higher C value tries to fit the training data more precisely but could lead to overfitting.

**11. What is out of bag evaluation?**  
**Answer:**  
Out of bag (OOB) evaluation is a method used in ensemble learning, especially in Random Forests, where each decision tree is trained on a bootstrap sample (random sampling with replacement) from the data. The samples that are not selected in the bootstrap sample are referred to as "out-of-bag." These out-of-bag samples are used to evaluate the performance of the model, providing an estimate of the model's accuracy without the need for a separate validation set.

**12. What do you understand by hard & soft voting classifier?**  
**Answer:**

* **Hard Voting Classifier**: In hard voting, each classifier in an ensemble makes a prediction, and the final prediction is made based on the majority vote. The class that gets the most votes is chosen as the final prediction.
* **Soft Voting Classifier**: In soft voting, each classifier in the ensemble outputs a probability for each class, and the final prediction is based on the class that has the highest average probability across all classifiers.

**13. Let’s suppose your ML algorithm is taking 5 min to train, how will you bring down time to 5 seconds for training? (Hint: Distributed Computation)**  
**Answer:**  
To bring down training time from 5 minutes to 5 seconds, you can use **distributed computation**. This involves running the training process on multiple machines or processors simultaneously, enabling the workload to be divided and processed in parallel. Tools like **Apache Spark**, **Hadoop**, or using **cloud-based services** like AWS or Google Cloud can help with distributed training by scaling out the computation across several nodes.

**14. Let’s Suppose I have trained 5 different models with the same training dataset & all of them have achieved 95% precision. Is there any chance that you can combine all these models to get better results? If yes, How? If no, Why?**  
**Answer:**  
Yes, it is possible to combine these models to potentially get better results. This can be done using an **ensemble learning** technique such as **stacking**, **bagging**, or **boosting**. By combining multiple models, you can reduce variance, bias, or overfitting, depending on the method used. For example, using **stacking**, you can train a meta-model on the predictions of the 5 models, which might improve accuracy by learning which models perform best under different circumstances.

**15. What do you understand by Gradient Descent? How will you explain Gradient Descent to a kid?**  
**Answer:**  
Gradient Descent is an optimization algorithm used to minimize the error in machine learning models by adjusting the parameters (like weights in a neural network). It works by computing the gradient (slope) of the error with respect to each parameter, then moving the parameters in the opposite direction of the gradient to reduce the error.

To explain it to a kid: Imagine you are at the top of a hill, and you want to find the lowest point in the valley. You can’t see the valley, but you can feel which way the ground slopes. You take small steps downhill, and by feeling the slope, you keep going until you reach the lowest point.

**16. Can you please explain the difference between regression & classification?**  
**Answer:**

* **Regression** is used when the output is a continuous value (e.g., predicting house prices, temperatures, or sales numbers). The goal is to predict a numerical value.
* **Classification** is used when the output is a category or class label (e.g., classifying emails as spam or not spam, diagnosing diseases). The goal is to assign an input to one of several predefined categories.

**17. Explain a clustering algorithm of your choice.**  
**Answer:**  
One common clustering algorithm is **K-means clustering**. It works by:

1. Choosing **K** initial cluster centroids (randomly or using a smarter initialization).
2. Assigning each data point to the nearest centroid based on Euclidean distance.
3. Recalculating the centroids by averaging the points assigned to each cluster.
4. Repeating the process until the centroids no longer change significantly.

The goal is to minimize the distance between the data points and their assigned centroids, leading to compact and well-separated clusters.

**18. How can you explain ML, DL, NLP, Computer Vision & Reinforcement Learning with examples in your own terms?**  
**Answer:**

* **Machine Learning (ML)**: ML is about teaching a machine to make decisions based on data. Example: A spam filter that classifies emails as spam or not based on patterns it has learned from previous emails.
* **Deep Learning (DL)**: DL is a subset of ML that uses neural networks with many layers (deep networks) to solve more complex problems. Example: Recognizing faces in photos using a deep neural network.
* **Natural Language Processing (NLP)**: NLP is a field of AI that focuses on how machines understand and process human language. Example: A chatbot that understands and responds to your questions in natural language.
* **Computer Vision**: This is about teaching machines to understand and interpret visual information from the world, like images or videos. Example: Self-driving cars that recognize traffic signs and pedestrians.
* **Reinforcement Learning**: This is when an agent learns to make decisions by interacting with an environment and receiving rewards or penalties. Example: A robot learning how to play a game by trial and error.

**19. How can you explain semi-supervised ML in your own way with an example?**  
**Answer:**  
**Semi-supervised Learning** is a type of machine learning where the model is trained on a small amount of labelled data and a large amount of unlabelled data. It’s like a teacher helping a student who already knows some things but still needs help with the rest.  
Example: You have a few labelled images of cats and dogs, but a lot of unlabelled images. A semi-supervised learning algorithm can use the labelled images to infer patterns and then apply them to the unlabelled images to classify them.

**20. What is the difference between abstraction & generalization in your own words?**  
**Answer:**

* **Abstraction** is the process of simplifying complex systems by focusing on the essential features and ignoring unnecessary details. It allows you to represent something in a more general or simplified way.  
  Example: A map of a city that only shows major roads, ignoring minor details like small shops.
* **Generalization** is the ability of a model to apply learned patterns to unseen data. It means that the model has learned the broader patterns and can make predictions beyond the specific examples it was trained on.  
  Example: A student learning to solve math problems who can apply the same technique to solve new, unseen problems.

**21. What are the steps that you have followed in your last project to prepare the dataset?**  
**Answer:**  
In my last project, the steps I followed to prepare the dataset were:

1. **Data Collection**: Gathering data from different sources (e.g., APIs, databases, CSV files).
2. **Data Cleaning**: Handling missing values, correcting inconsistencies, and removing duplicates.
3. **Data Transformation**: Converting categorical variables to numerical values (e.g., one-hot encoding) and normalizing or scaling features.
4. **Feature Selection**: Identifying and selecting relevant features for model training.
5. **Splitting Data**: Dividing the dataset into training, validation, and test sets.

**22. In your last project, what steps were involved in the model selection procedure?**  
**Answer:**  
In my last project, the steps involved in model selection were:

1. **Problem Definition**: Clearly understanding the problem (e.g., classification, regression) to determine the type of model required.
2. **Model Selection Criteria**: Defining performance metrics (e.g., accuracy, precision, recall) to evaluate models.
3. **Choosing Algorithms**: Testing several algorithms based on the problem type (e.g., logistic regression, decision trees, random forests, etc.).
4. **Cross-validation**: Using techniques like k-fold cross-validation to evaluate models on different subsets of data.
5. **Model Tuning**: Hyperparameter tuning to optimize model performance.
6. **Model Evaluation**: Evaluating models using validation and test data to ensure the model generalizes well.

**23. If I give you 2 columns of any dataset, what will be the steps involved to check the relationship between those 2 columns?**  
**Answer:**  
To check the relationship between two columns, the steps involved would be:

1. **Data Exploration**: Visualizing the data using plots like scatter plots (for continuous data) or bar plots (for categorical data).
2. **Correlation Analysis**: Calculating the correlation coefficient (e.g., Pearson or Spearman) if both columns are numerical.
3. **Statistical Test**: Applying statistical tests like chi-square test (for categorical columns) or t-tests (for numerical columns) to check for significance.
4. **Modeling**: Building a simple model (e.g., linear regression for continuous data) to quantify the relationship between the columns.

**24. Can you please explain 5 different kinds of strategies at least to handle missing values in a dataset?**  
**Answer:**  
Five strategies to handle missing values in a dataset include:

1. **Deletion**: Removing rows or columns with missing values, though this may lead to loss of information.
2. **Imputation**: Filling missing values with a statistical measure like mean, median, or mode.
3. **Forward Fill**: Using the last observed value to fill missing data (typically used in time-series data).
4. **Interpolation**: Estimating missing values based on the surrounding data points (linear or spline interpolation).
5. **Using Predictive Models**: Creating a model to predict missing values based on other features (e.g., regression, KNN imputation).

**25. What kind of different issues have you faced with respect to your raw data? At least mention 5 issues.**  
**Answer:**  
Five issues I have faced with raw data include:

1. **Missing Values**: Some columns or rows had missing values.
2. **Data Inconsistencies**: Inconsistent formatting, such as different date formats or text encoding issues.
3. **Outliers**: Extreme values that could distort the analysis or model performance.
4. **Duplicate Entries**: Some rows were repeated, leading to biased results.
5. **Data Type Mismatch**: Some columns had incorrect data types (e.g., numeric columns stored as strings).

**26. What is your strategy to handle categorical datasets? Explain with an example.**  
**Answer:**  
To handle categorical datasets, my strategy includes:

1. **Encoding Categorical Data**: Using **One-Hot Encoding** to convert categorical variables into binary format (e.g., "Red", "Blue", and "Green" become three separate columns with 0s and 1s).
2. **Label Encoding**: Assigning an integer value to each category (e.g., "Red" = 1, "Blue" = 2, "Green" = 3) when there is an ordinal relationship.
3. **Handling Rare Categories**: Grouping less frequent categories into an "Other" category to avoid sparsity.
4. **Feature Engineering**: Creating new features that might help the model better understand categorical data relationships.

**Example**: If a dataset has a column with colors like "Red", "Blue", and "Green", one-hot encoding would create three new columns (Red, Blue, Green) with binary values.

**27. How do you define a model in terms of machine learning or in your own words?**  
**Answer:**  
A **model** in machine learning is a mathematical representation or an algorithm that learns patterns and relationships from data to make predictions or decisions. The model is trained on data to identify these patterns and then applies what it has learned to new, unseen data to make informed predictions or classifications.

**28. What do you understand by k-fold validation & in what situation you have used k-fold cross-validation?**  
**Answer:**  
**K-fold Cross-Validation** is a technique used to assess the performance of a model by dividing the data into K equally sized subsets (or "folds"). The model is trained on K-1 folds and tested on the remaining fold, and this process is repeated K times, each time using a different fold as the test set. This helps evaluate the model’s performance more robustly and prevents overfitting.

I have used k-fold cross-validation when tuning a model and trying to select the best hyperparameters, as it helps ensure the model performs well across different subsets of the data.

**29. What is the meaning of bootstrap sampling? Explain me in your own words.**  
**Answer:**  
**Bootstrap Sampling** is a technique in which multiple subsets of data are generated by randomly sampling from the original dataset with replacement. This means that some data points can appear multiple times in a subset, while others may not appear at all. It's commonly used in ensemble methods like **Bagging** to improve model performance by generating diverse subsets of data.

**30. What do you understand by underfitting & overfitting of a model with an example?**  
**Answer:**

* **Underfitting** occurs when a model is too simple and cannot capture the underlying patterns of the data. This typically happens when the model has high bias and low variance, leading to poor performance on both training and testing data.  
  **Example**: Using a linear model to predict house prices where the data has a complex, non-linear relationship will likely lead to underfitting.
* **Overfitting** occurs when a model is too complex and learns the noise in the training data, leading to high variance and poor generalization to new data. The model performs well on the training data but fails to generalize.  
  **Example**: A decision tree with too many branches might perform well on training data but poorly on unseen test data, as it memorizes specific details that do not generalize well.

Here are the questions with their corresponding answers in a Q&A format:

**31. What is the difference between cross-validation and bootstrapping?**  
**Answer:**

* **Cross-validation**: In cross-validation (specifically k-fold cross-validation), the dataset is divided into **k** subsets or "folds." The model is trained on **k-1** folds and tested on the remaining fold. This process is repeated **k** times, with each fold serving as the test set once.
* **Bootstrapping**: Bootstrapping involves generating multiple subsets of data by randomly sampling from the original dataset with **replacement**. This means that some data points may appear more than once in a subset, while others may not appear at all. It is often used in ensemble methods like bagging.

The key difference is that cross-validation splits the dataset into distinct training and testing sets for each fold, whereas bootstrapping allows repeated sampling of the same data points.

**32. What do you understand by silhouette coefficient?**  
**Answer:**  
The **silhouette coefficient** is a metric used to evaluate the quality of a clustering solution. It measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette score ranges from -1 to 1:

* A score near 1 indicates that the object is well clustered.
* A score near 0 indicates that the object lies between two clusters.
* A negative score indicates that the object may have been incorrectly assigned to the wrong cluster.

It is useful for assessing the effectiveness of clustering algorithms like K-means.

**33. What is the advantage of using ROC Score?**  
**Answer:**  
The **ROC (Receiver Operating Characteristic) score** is useful for evaluating the performance of binary classification models. It plots the **True Positive Rate (sensitivity)** against the **False Positive Rate (1-specificity)** at various thresholds. The main advantages are:

* It helps compare models across different thresholds, giving a broader view of performance.
* The **AUC (Area Under the Curve)** value derived from ROC represents the model’s ability to distinguish between positive and negative classes, with a higher value indicating better performance.
* It is particularly useful when dealing with imbalanced datasets, where accuracy may not be a reliable performance metric.

**34. Explain the complete approach to evaluate your regression model.**  
**Answer:**  
To evaluate a regression model, the following steps are typically followed:

1. **Split the Data**: Divide the dataset into training and testing sets (typically 80/20 or 70/30).
2. **Train the Model**: Fit the regression model on the training data.
3. **Prediction**: Use the trained model to make predictions on the testing data.
4. **Evaluate Performance**: Use various metrics to assess the model's performance:
   * **Mean Absolute Error (MAE)**: Average of the absolute differences between actual and predicted values.
   * **Mean Squared Error (MSE)**: Average of the squared differences between actual and predicted values.
   * **Root Mean Squared Error (RMSE)**: Square root of the MSE, which gives an error metric in the same unit as the target variable.
   * **R-squared (R²)**: Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.
5. **Residual Analysis**: Plot residuals (differences between actual and predicted values) to check for patterns. Ideally, residuals should be randomly scattered around zero, indicating that the model is capturing all the trends in the data.
6. **Cross-validation**: Use cross-validation to further assess the model's ability to generalize to unseen data.

**35. Give me an example of lazy learner and eager learner algorithms.**  
**Answer:**

* **Lazy Learner**: A lazy learner is a type of machine learning algorithm that delays the learning process until it is required to make a prediction. These algorithms do not build a model during the training phase; instead, they store the entire dataset and make predictions based on the dataset when required.
  + **Example**: **K-Nearest Neighbors (KNN)**.
* **Eager Learner**: An eager learner builds a model during the training phase, which can be used directly to make predictions. The learning process happens before predictions are needed.
  + **Example**: **Decision Trees**, **Linear Regression**.

**36. What do you understand by the holdout method?**  
**Answer:**  
The **holdout method** is a simple technique used to evaluate a machine learning model. In this method, the dataset is split into two parts: one for training the model and the other for testing its performance. Typically, the dataset is divided into **training (e.g., 80%)** and **testing (e.g., 20%)** sets. The model is trained on the training set and then evaluated on the testing set to check its performance.

**37. What is the difference between predictive modeling and descriptive modeling?**  
**Answer:**

* **Predictive Modeling**: Predictive modeling is used to make predictions about future events or outcomes based on historical data. It focuses on predicting the value of a target variable.
  + Example: Predicting house prices based on features like location, size, and number of rooms.
* **Descriptive Modeling**: Descriptive modeling is used to describe and understand the patterns, relationships, and characteristics in a dataset. It is often used for summarization or grouping.
  + Example: Segmenting customers into different groups based on their purchasing behavior (clustering).

**38. How have you derived a feature for model building in your last project?**  
**Answer:**  
In my last project, feature engineering was a critical step in model building. The approach included:

1. **Data Exploration**: Identifying potential features by analyzing the dataset, domain knowledge, and the problem requirements.
2. **Transforming Raw Data**: Creating new features such as aggregating data (e.g., sum, mean, or count of certain variables), normalizing or scaling features, or handling categorical variables with one-hot encoding.
3. **Feature Interaction**: Combining existing features to create new ones, such as calculating the ratio of two numerical features or creating interaction terms.
4. **Feature Selection**: Using techniques like **correlation analysis**, **Recursive Feature Elimination (RFE)**, and **L1 regularization** to select the most important features.
5. **Domain Expertise**: Leveraging knowledge from the domain (e.g., banking, healthcare) to create meaningful features that are likely to improve model performance.

**39. Explain 5 different encoding techniques.**  
**Answer:**

1. **One-Hot Encoding**: Converts categorical variables into binary vectors (e.g., "Red", "Blue", "Green" becomes three columns with 1s and 0s).
2. **Label Encoding**: Converts categorical variables into numerical labels (e.g., "Red" = 0, "Blue" = 1, "Green" = 2).
3. **Ordinal Encoding**: Similar to label encoding but used for ordinal data where there is an inherent order (e.g., "Low", "Medium", "High" could be encoded as 1, 2, 3).
4. **Binary Encoding**: Combines hash encoding and one-hot encoding, mapping categories into binary digits (used for high cardinality categorical features).
5. **Frequency Encoding**: Encodes categorical variables based on the frequency of their occurrence in the dataset (e.g., if "Red" appears 100 times, it is encoded as 100).

**40. How do you define some features as not important for an ML model? What strategy will you follow?**  
**Answer:**  
To identify unimportant features, I would look for:

1. **Correlation**: Features that are highly correlated with others may be redundant and can be removed.
2. **Feature Importance**: Using algorithms like **Decision Trees** or **Random Forests** that provide feature importance scores.
3. **Univariate Selection**: Statistical tests (e.g., chi-square or ANOVA) to check the relationship between features and the target variable.
4. **Recursive Feature Elimination (RFE)**: Iteratively removing the least important features and checking the model's performance.
5. **Domain Knowledge**: Some features may be deemed irrelevant based on the understanding of the problem domain.

The strategy I would follow includes removing or transforming unimportant features using feature selection techniques or dimensionality reduction methods like **PCA** to improve model performance and reduce overfitting.

**41. What is the difference between Euclidean distance and Manhattan distance? Explain in simple words.**  
**Answer:**

* **Euclidean Distance** is the straight-line distance between two points in a space. It’s calculated using the Pythagorean theorem and represents the shortest path between two points.  
  **Formula**: Euclidean Distance=(x2−x1)2+(y2−y1)2\text{Euclidean Distance} = \sqrt{(x\_2 - x\_1)^2 + (y\_2 - y\_1)^2}Euclidean Distance=(x2​−x1​)2+(y2​−y1​)2​.
* **Manhattan Distance** is the sum of the absolute differences of their coordinates. It represents the total distance traveled along grid-like streets (like the streets of Manhattan), where you can only move horizontally and vertically.  
  **Formula**: Manhattan Distance=∣x2−x1∣+∣y2−y1∣\text{Manhattan Distance} = |x\_2 - x\_1| + |y\_2 - y\_1|Manhattan Distance=∣x2​−x1​∣+∣y2​−y1​∣.

In simple terms: Euclidean distance is like flying directly from one point to another, while Manhattan distance is like driving along city streets, following the grid layout.

**42. What do you understand by feature selection, transformation, engineering, and EDA? What are the steps that you have performed in each of these in detail with examples?**  
**Answer:**

* **Feature Selection**: This process involves selecting the most relevant features for model training while eliminating unnecessary or redundant ones.
  + **Example**: Using **Recursive Feature Elimination (RFE)** to select the most important features for a classification model.
* **Feature Transformation**: This involves changing the scale or format of the features to make them suitable for modeling.
  + **Example**: Applying **Min-Max Scaling** to scale features to a range of 0 to 1 or **Log Transformation** to handle skewed data.
* **Feature Engineering**: This is the process of creating new features from the existing ones to improve model performance.
  + **Example**: Creating a **"total\_transaction\_amount"** feature by summing up transaction amounts for a user in a fraud detection system.
* **Exploratory Data Analysis (EDA)**: This involves analyzing and visualizing the data to understand patterns, relationships, and distributions before model building.
  + **Example**: Using **pair plots** to visualize correlations between different features or **box plots** to check for outliers.

**43. What is the difference between Singular Value Decomposition (SVD) and PCA? (Hint: SVD is one of the ways to do PCA.)**  
**Answer:**

* **Singular Value Decomposition (SVD)** is a matrix factorization technique that decomposes a matrix into three smaller matrices: UUU, Σ\SigmaΣ, and VTV^TVT. It’s widely used in various machine learning tasks like dimensionality reduction and matrix approximation.
* **Principal Component Analysis (PCA)** is a technique used for reducing the dimensionality of data by transforming it into a new set of uncorrelated variables called principal components, ordered by the variance they explain. PCA can be computed using SVD as one of the methods.

In simple terms, **SVD** is a general-purpose matrix factorization technique, and **PCA** is a specific application of SVD used for dimensionality reduction.

**44. What kind of feature transformations have you done in your last project?**  
**Answer:**  
In my last project, I performed the following feature transformations:

1. **Normalization**: I applied **Min-Max scaling** to features like transaction amount to bring them within a range of 0 to 1.
2. **Log Transformation**: I transformed skewed numerical data using **logarithmic transformation** to make the distribution more normal.
3. **One-Hot Encoding**: I used **one-hot encoding** for categorical variables like transaction type to convert them into binary features.
4. **Binning**: I created new features by **binning** continuous features into discrete intervals, such as grouping ages into bins (20-30, 30-40).

**45. Have you taken any external features in any of the projects from third-party data? If yes, explain that scenario.**  
**Answer:**  
Yes, in a past project involving credit card fraud detection, I integrated **external data** from a third-party source to enhance the model. Specifically, I used publicly available **geolocation data** to determine whether transactions were occurring in a foreign country. This feature helped the model distinguish between domestic and international transactions, improving its ability to identify potentially fraudulent activity.

**46. If your model is overfitted, what will you do next?**  
**Answer:**  
If the model is overfitted, I would consider the following actions:

1. **Simplify the Model**: Reduce the complexity by using simpler algorithms (e.g., reducing the depth of decision trees).
2. **Cross-Validation**: Use **k-fold cross-validation** to better assess the model's ability to generalize.
3. **Regularization**: Apply **L1 (Lasso)** or **L2 (Ridge)** regularization to penalize large coefficients and reduce model complexity.
4. **Increase Training Data**: Gather more data to allow the model to generalize better.
5. **Early Stopping**: In neural networks, use **early stopping** to halt training when performance on the validation set starts to degrade.

**47. Explain the bias-variance trade-off.**  
**Answer:**  
The **bias-variance trade-off** refers to the balance between two sources of errors in a model:

* **Bias**: The error introduced by making strong assumptions about the data (e.g., assuming a linear relationship when it's actually non-linear). High bias leads to **underfitting**.
* **Variance**: The error introduced by the model’s sensitivity to small fluctuations in the training data. High variance leads to **overfitting**.

The trade-off occurs because increasing model complexity tends to reduce bias but increase variance, and vice versa. The goal is to find a balance that minimizes both bias and variance, leading to better generalization.

**48. What steps would you take to improve the accuracy of your model? At least mention 5 approaches, and justify why you would choose those approaches.**  
**Answer:**

1. **Feature Engineering**: Creating new meaningful features (e.g., interaction terms) can significantly improve the model's predictive power.
2. **Hyperparameter Tuning**: Using techniques like **Grid Search** or **Random Search** to find the optimal parameters for the model.
3. **Cross-Validation**: Using **k-fold cross-validation** to ensure the model is evaluated on different subsets of data, reducing the risk of overfitting.
4. **Ensemble Methods**: Combining multiple models using techniques like **Random Forests** or **XGBoost** to improve accuracy through the aggregation of diverse models.
5. **Increasing Training Data**: More data helps the model generalize better and reduces overfitting, leading to higher accuracy.

**49. Explain the process of feature engineering in the context of text categorization.**  
**Answer:**  
Feature engineering for text categorization typically involves the following steps:

1. **Tokenization**: Splitting text into smaller units like words or sentences.
2. **Stop Word Removal**: Eliminating common words (e.g., "the", "and", "is") that do not carry meaningful information.
3. **Vectorization**: Converting text data into numerical representations, such as **TF-IDF** or **Word2Vec**.
4. **Stemming/Lemmatization**: Reducing words to their root form (e.g., "running" to "run").
5. **Feature Extraction**: Extracting relevant features, such as word counts, term frequency, or sentence length, to capture the key characteristics of the text.

**50. Explain vectorization and Hamming distance.**  
**Answer:**

* **Vectorization**: This is the process of converting text or categorical data into numerical representations, typically as vectors (arrays of numbers). Common techniques include **TF-IDF (Term Frequency-Inverse Document Frequency)** and **Word2Vec**, which represent text data in a way that machine learning algorithms can process.
* **Hamming Distance**: Hamming distance is a metric used to measure the difference between two strings of equal length by counting the number of positions at which the corresponding elements are different. It is used primarily for comparing binary data or strings. **Example**: The Hamming distance between "karolin" and "kathrin" is 3 (since the three different characters are "r" vs. "t", "o" vs. "h", and "l" vs. "r").

**PART - 5**

**1. Can you please explain the chain rule and its use?**  
**Answer:**  
The **chain rule** is a fundamental concept in calculus, used to compute the derivative of a composite function. If we have two functions, say f(x)f(x)f(x) and g(x)g(x)g(x), the chain rule allows us to differentiate the composite function f(g(x))f(g(x))f(g(x)) by multiplying the derivative of fff with respect to g(x)g(x)g(x) and the derivative of g(x)g(x)g(x) with respect to xxx.  
**Formula**:  
ddx[f(g(x))]=dfdg⋅dgdx\frac{d}{dx}[f(g(x))] = \frac{df}{dg} \cdot \frac{dg}{dx}dxd​[f(g(x))]=dgdf​⋅dxdg​.

In machine learning, the chain rule is especially useful in backpropagation algorithms for neural networks, where the gradients are propagated backward through the network layer by layer to optimize the model.

**2. What is the difference between correlation and covariance?**  
**Answer:**

* **Correlation** is a normalized measure of the relationship between two 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. Correlation is scale-independent.
* **Covariance** is a measure of the joint variability of two variables. It shows whether two variables increase or decrease together but is not normalized and can take any value, making it sensitive to the scale of the variables.  
  **Formula** for covariance:  
  cov(X,Y)=1n∑(Xi−Xˉ)(Yi−Yˉ)\text{cov}(X, Y) = \frac{1}{n} \sum (X\_i - \bar{X})(Y\_i - \bar{Y})cov(X,Y)=n1​∑(Xi​−Xˉ)(Yi​−Yˉ).

**3. What are the sampling techniques you have used in your project?**  
**Answer:**  
In my projects, I have used the following sampling techniques:

1. **Random Sampling**: I randomly selected subsets of data from the dataset to train models or for cross-validation.
2. **Stratified Sampling**: When the dataset is imbalanced (e.g., fraud detection), I ensured that each class was proportionally represented in the training and test sets.
3. **Bootstrap Sampling**: Used for creating multiple training sets by randomly sampling with replacement, particularly useful for ensemble methods like bagging.
4. **Systematic Sampling**: Used when the data has an inherent order; I select every nth data point from the dataset.

**4. Have you ever used Hypothesis testing in your last project, if yes, explain how?**  
**Answer:**  
Yes, in my last project, I used **hypothesis testing** to assess whether certain features significantly influenced the likelihood of credit card fraud. For example, I used a **t-test** to compare the average transaction amounts between fraudulent and non-fraudulent transactions. The null hypothesis was that there is no difference in the transaction amounts between the two groups, and I rejected the null hypothesis if the p-value was below a significance threshold (usually 0.05).

**5. In which case would you use a Naive Bayes classifier and a Decision Tree separately?**  
**Answer:**

* **Naive Bayes** is useful when the features are conditionally independent, or when you have a large number of categorical features, as it assumes independence between them. It is especially good for text classification tasks such as spam detection or sentiment analysis.  
  **Scenario**: Use Naive Bayes in a situation where you are classifying emails as spam or not spam based on a large number of word frequencies.
* **Decision Trees** work well when the data has non-linear relationships between features. They are more flexible than Naive Bayes and do not require the features to be independent. Decision trees can be used for both classification and regression.  
  **Scenario**: Use a Decision Tree for predicting whether a loan applicant will default or not based on features like income, age, and credit score.

**6. What is the advantage and disadvantage of the Naive Bayes classifier?**  
**Answer:**  
**Advantages**:

1. **Simple and fast**: Naive Bayes is easy to implement and computationally efficient, especially for large datasets.
2. **Works well with small datasets**: It can perform well even when the amount of data is limited.
3. **Works well with categorical data**: Particularly effective for problems like text classification and spam detection.

**Disadvantages**:

1. **Assumption of independence**: Naive Bayes assumes that all features are independent, which is rarely true in real-world data. This can lead to poor performance when this assumption is violated.
2. **Sensitivity to irrelevant features**: Naive Bayes may perform poorly if there are many irrelevant or noisy features.

**7. In case of numerical data, what is the Naive Bayes classification equation you will use?**  
**Answer:**  
For numerical data, Naive Bayes assumes that the features are normally distributed. The classification equation is based on **Bayes' Theorem**:

P(C∣X)=P(C)∏P(Xi∣C)P(X)P(C|X) = \frac{P(C) \prod P(X\_i|C)}{P(X)}P(C∣X)=P(X)P(C)∏P(Xi​∣C)​

Where:

* P(C∣X)P(C|X)P(C∣X) is the posterior probability of class CCC given the features XXX.
* P(C)P(C)P(C) is the prior probability of the class.
* P(Xi∣C)P(X\_i|C)P(Xi​∣C) is the likelihood of each feature XiX\_iXi​ given the class, which is modeled using a Gaussian distribution for each feature.

For each feature XiX\_iXi​, we compute the likelihood P(Xi∣C)P(X\_i|C)P(Xi​∣C) using the Gaussian probability density function:

P(Xi∣C)=12πσ2exp⁡(−(Xi−μ)22σ2)P(X\_i|C) = \frac{1}{\sqrt{2 \pi \sigma^2}} \exp\left(-\frac{(X\_i - \mu)^2}{2\sigma^2}\right)P(Xi​∣C)=2πσ2​1​exp(−2σ2(Xi​−μ)2​)

Where μ\muμ and σ\sigmaσ are the mean and standard deviation of the feature XiX\_iXi​ for class CCC.

**8. Give me a scenario where I will be able to use a boosting classifier and regressor?**  
**Answer:**  
**Boosting Classifier**: Boosting can be used when you have a binary classification problem and need to improve model accuracy by combining multiple weak models (like decision trees) into a strong model.  
**Scenario**: Predicting whether a customer will churn (yes/no) based on customer activity and demographic data.

**Boosting Regressor**: Boosting can also be used for regression tasks, where the goal is to predict a continuous output variable.  
**Scenario**: Predicting house prices based on features like location, square footage, and number of bedrooms.

**9. In the case of a Bayesian classifier, what exactly does it try to learn? Define its learning procedure.**  
**Answer:**  
A **Bayesian classifier** learns the **conditional probability distribution** of the classes given the features. Specifically, it tries to compute the probability of each class (class posterior) given the observed features using **Bayes' Theorem**.  
**Learning Procedure**:

1. **Estimate priors**: Calculate the prior probability for each class, P(C)P(C)P(C).
2. **Estimate likelihoods**: Calculate the likelihood of each feature for each class, P(Xi∣C)P(X\_i|C)P(Xi​∣C), assuming a particular distribution (e.g., Gaussian).
3. **Apply Bayes' Theorem**: For a new data point, calculate the posterior probability of each class given the features and choose the class with the highest posterior probability.

**10. Give me a situation where I will be able to use SVM instead of Logistic regression.**  
**Answer:**  
Support Vector Machines (SVM) are often used when the data has a **non-linear relationship** between features or when there is a need for a **high-dimensional feature space**. Logistic regression, while effective in simpler cases, can struggle when there is complex boundary separation.  
**Scenario**: If you're classifying handwritten digits (such as in the MNIST dataset), SVM is likely to perform better because the data has non-linear decision boundaries that logistic regression may not capture well. SVM uses kernel tricks to transform the feature space, making it better for complex, non-linear separations.

**11. What do you understand by RBF kernel in SVM?**  
**Answer:**  
The **Radial Basis Function (RBF) kernel** is a popular kernel function used in Support Vector Machines (SVM) for non-linear classification problems. The RBF kernel computes the similarity between two data points in a higher-dimensional space, transforming the original feature space into a space where a hyperplane can separate the data points linearly.  
The formula for the RBF kernel is:

K(x,y)=exp⁡(−∥x−y∥22σ2)K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)K(x,y)=exp(−2σ2∥x−y∥2​)

Where ∥x−y∥\|x - y\|∥x−y∥ is the Euclidean distance between two points, and σ\sigmaσ is a parameter that controls the width of the Gaussian function. RBF kernel is especially useful for problems where the decision boundary is highly non-linear.

**12. Give me 2 scenarios where AI can be used to increase revenue in the travel industry.**  
**Answer:**

1. **Dynamic Pricing:** AI can be used to optimize pricing for travel services like flights, hotels, and car rentals. By analyzing customer behavior, demand patterns, and market trends, AI models can predict the best time to adjust prices to maximize revenue, ensuring competitive pricing while maintaining high demand.
2. **Personalized Recommendations:** AI can improve customer experience by providing personalized travel recommendations. By analyzing user preferences, past behaviors, and trends, AI-powered systems can suggest personalized travel packages, tours, and accommodations, thereby increasing the likelihood of additional bookings and enhancing customer satisfaction.

**13. What do you understand by a leaf node in a decision tree?**  
**Answer:**  
In a **decision tree**, a **leaf node** represents the final output or class prediction for a given set of features. It is a terminal node that does not have any further branches or decisions. Each path from the root node to a leaf node corresponds to a series of decision rules applied to the data, leading to a prediction or classification. For example, in a classification problem, each leaf node will contain the majority class of the samples that reached it.

**14. What is information gain and entropy in a decision tree?**  
**Answer:**

* **Entropy** is a measure of uncertainty or impurity in the data. It quantifies the disorder or randomness in the dataset. The formula for entropy is:

H(S)=−∑p(x)log⁡2p(x)H(S) = - \sum p(x) \log\_2 p(x)H(S)=−∑p(x)log2​p(x)

where p(x)p(x)p(x) is the probability of a class in the dataset. If all the instances belong to one class, entropy is 0 (perfectly pure), and if the classes are evenly distributed, entropy is 1 (completely impure).

* **Information Gain** is used to measure the effectiveness of a feature in classifying the data. It is the reduction in entropy achieved by partitioning the dataset based on a specific attribute. Information gain is calculated as the difference between the entropy of the parent node and the weighted average entropy of the child nodes:

IG(S,A)=H(S)−∑∣Si∣∣S∣H(Si)IG(S, A) = H(S) - \sum \frac{|S\_i|}{|S|} H(S\_i)IG(S,A)=H(S)−∑∣S∣∣Si​∣​H(Si​)

where SiS\_iSi​ represents the subsets after splitting by attribute AAA, and H(Si)H(S\_i)H(Si​) is the entropy of the subsets.

**15. Give disadvantages of using Decision Tree.**  
**Answer:**

1. **Overfitting**: Decision trees tend to overfit the data, especially if the tree is deep, capturing noise or irrelevant patterns in the training data.
2. **Instability**: Small changes in the data can result in a completely different tree structure, making decision trees sensitive to fluctuations in the dataset.
3. **Bias towards features with more levels**: Decision trees may favor features with many possible values, leading to biased decisions when choosing the best feature for a split.
4. **Poor performance on complex patterns**: Decision trees may struggle with capturing complex relationships in the data, particularly when the decision boundaries are not axis-aligned.
5. **High variance**: A deep decision tree may have high variance and might not generalize well to unseen data.

**16. List some of the features of Random Forest.**  
**Answer:**

1. **Ensemble Method**: Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve accuracy.
2. **Bagging**: It uses bootstrap aggregating (bagging) to create different subsets of data and builds a tree for each subset, reducing variance.
3. **Random Feature Selection**: At each split, a random subset of features is chosen, ensuring diversity among the trees and helping prevent overfitting.
4. **Feature Importance**: Random Forest can evaluate feature importance by assessing how much each feature contributes to reducing the impurity at each split.
5. **Handles Missing Data**: Random Forest can handle missing data by splitting the data based on available features and using surrogates for missing values.
6. **Works with Classification and Regression**: It can be used for both classification and regression problems.

**17. How can you avoid overfitting in a decision tree?**  
**Answer:**  
To avoid overfitting in decision trees:

1. **Pruning**: Limit the depth of the tree or prune branches that have little importance or don't significantly reduce the impurity.
2. **Minimum Samples per Leaf**: Set a minimum number of samples required in a leaf node, ensuring that nodes with too few samples are not split further.
3. **Maximum Depth**: Limit the maximum depth of the tree, preventing it from growing too deep and capturing noise.
4. **Maximum Features**: Limit the number of features considered for splitting at each node, reducing complexity and overfitting.
5. **Cross-Validation**: Use cross-validation to assess the model's performance on different data splits and detect overfitting early.

**18. Explain polynomial regression in your own way.**  
**Answer:**  
**Polynomial regression** is an extension of linear regression where we fit a polynomial equation to the data instead of a straight line. It allows the model to capture non-linear relationships between the independent variable(s) and the dependent variable. For example, if a dataset shows a curving trend, polynomial regression can provide a curve that fits better than a straight line.

The equation for a polynomial regression model of degree nnn is:

y=β0+β1x+β2x2+⋯+βnxny = \beta\_0 + \beta\_1x + \beta\_2x^2 + \cdots + \beta\_nx^ny=β0​+β1​x+β2​x2+⋯+βn​xn

where β0,β1,…,βn\beta\_0, \beta\_1, \dots, \beta\_nβ0​,β1​,…,βn​ are the coefficients to be learned, and xxx is the independent variable raised to the power of the degree.

**19. Explain the learning mechanism of linear regression.**  
**Answer:**  
Linear regression works by finding the best-fitting line through the data points, which minimizes the **mean squared error (MSE)** between the predicted and actual values. The model assumes a linear relationship between the dependent variable yyy and the independent variable(s) xxx. The learning mechanism involves finding the optimal values of the coefficients (weights) β0,β1,…\beta\_0, \beta\_1, \dotsβ0​,β1​,… that minimize the error.

The cost function for linear regression is:

MSE=1n∑i=1n(yi−(β0+β1xi))2MSE = \frac{1}{n} \sum\_{i=1}^{n} (y\_i - (\beta\_0 + \beta\_1x\_i))^2MSE=n1​i=1∑n​(yi​−(β0​+β1​xi​))2

Linear regression uses optimization algorithms (like gradient descent) to minimize the cost function and determine the best-fitting line.

**20. What is the cost function in logistic regression?**  
**Answer:**  
In **logistic regression**, the cost function is the **logistic loss** or **binary cross-entropy** function, which measures how well the model's predictions align with the actual labels. For binary classification, the cost function is:

J(θ)=−1m∑i=1m[yilog⁡(hθ(xi))+(1−yi)log⁡(1−hθ(xi))]J(\theta) = - \frac{1}{m} \sum\_{i=1}^{m} \left[ y\_i \log(h\_\theta(x\_i)) + (1 - y\_i) \log(1 - h\_\theta(x\_i)) \right]J(θ)=−m1​i=1∑m​[yi​log(hθ​(xi​))+(1−yi​)log(1−hθ​(xi​))]

Where:

* hθ(x)=11+e−θTxh\_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}hθ​(x)=1+e−θTx1​ is the hypothesis function (sigmoid).
* yiy\_iyi​ is the actual label (0 or 1).
* hθ(xi)h\_\theta(x\_i)hθ​(xi​) is the predicted probability of the positive class for the iii-th sample.
* mmm is the total number of training examples.

The goal is to minimize the cost function to improve the model's accuracy in predicting the probability of class membership.

**21. What is the error function in linear regression?**  
**Answer:**  
In linear regression, the error function is typically the **Mean Squared Error (MSE)**, which measures the average squared difference between the observed actual outcomes and the predicted values. The formula for MSE is:

MSE=1m∑i=1m(yi−y^i)2MSE = \frac{1}{m} \sum\_{i=1}^{m} (y\_i - \hat{y}\_i)^2MSE=m1​i=1∑m​(yi​−y^​i​)2

Where:

* yiy\_iyi​ is the actual value,
* y^i\hat{y}\_iy^​i​ is the predicted value,
* mmm is the total number of data points.

Minimizing the MSE ensures that the model's predictions are as close as possible to the actual values.

**22. What is the use of implementing OLS technique with respect to a dataset?**  
**Answer:**  
The **Ordinary Least Squares (OLS)** technique is a method for estimating the parameters of a linear regression model. It minimizes the sum of the squared residuals (the differences between observed and predicted values) to find the best-fitting line. OLS is used in regression tasks to estimate the relationship between the independent variables and the dependent variable, providing a way to predict outcomes and understand underlying patterns in the data.

OLS is particularly useful when the assumption of linearity holds and when you are looking for a simple yet effective method to model the relationship between variables.

**23. Explain dendrogram in your own way.**  
**Answer:**  
A **dendrogram** is a tree-like diagram that is used to represent hierarchical clustering. It shows the arrangement of clusters formed during the clustering process and how they are merged or split at different levels. Each node in a dendrogram represents a cluster, and the branches show the merging of clusters at various levels of similarity. The height of the branches indicates the distance or dissimilarity between clusters. Dendrograms are helpful for visualizing the hierarchy of clusters and determining the optimal number of clusters.

**24. How do you measure the quality of clusters in DBSCAN?**  
**Answer:**  
In **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**, the quality of clusters is typically measured using the following methods:

1. **Silhouette Score**: It measures how similar an object is to its own cluster compared to other clusters. A higher silhouette score indicates better-defined clusters.
2. **Cluster Density**: The quality of DBSCAN clusters can also be evaluated based on the density of points within a cluster. DBSCAN groups points that are close together and has a noise (outlier) component, so clusters with dense regions are preferred.
3. **Visual Inspection**: In some cases, a simple visual inspection of the clusters (using scatter plots, for example) can provide insights into the quality of clustering.

**25. How do you evaluate DBSCAN algorithm?**  
**Answer:**  
DBSCAN can be evaluated based on:

1. **Silhouette Score**: This score provides an indication of how well each point lies within its cluster and whether it is appropriately grouped or placed in the wrong cluster.
2. **Cluster Validity Indices**: These indices include measures like **Davies-Bouldin Index** and **Dunn Index**, which evaluate how well-separated and compact the clusters are.
3. **Handling of Noise**: Since DBSCAN can classify some points as noise (outliers), evaluating how well it identifies and handles outliers is an important factor.
4. **Visualizations**: Visualizing the data (e.g., using scatter plots) and checking whether the clusters formed by DBSCAN align with expected groupings or patterns can help in evaluating the algorithm.

**26. What do you understand by market basket analysis?**  
**Answer:**  
**Market Basket Analysis** is a data mining technique used to discover associations or patterns between products that frequently co-occur in transactions. It is commonly used in retail to analyze consumer behavior and identify product associations (e.g., "customers who bought X also bought Y"). The most common method for market basket analysis is the **Apriori algorithm**, which finds frequent itemsets and generates association rules. This helps businesses in cross-selling, promotions, and inventory management.

**27. Explain centroid formation technique in K-Means algorithm.**  
**Answer:**  
In the **K-Means** algorithm, the **centroid** is the central point of a cluster, representing the average location of all the points within that cluster. The process of centroid formation is as follows:

1. **Initialization**: Initially, K centroids are chosen randomly from the data points.
2. **Assignment Step**: Each data point is assigned to the nearest centroid, forming K clusters.
3. **Update Step**: The centroid of each cluster is recalculated as the mean of all data points within that cluster.
4. **Repeat**: Steps 2 and 3 are repeated iteratively until the centroids stop changing (convergence).

The centroid serves as the "representative" of the cluster and is used to minimize the within-cluster sum of squared distances (also called inertia).

**28. Have you ever used SVM regression in any of your projects? If yes, Why?**  
**Answer:**  
Yes, **SVM regression** (Support Vector Machine Regression) can be very useful when the relationship between the features and the target variable is non-linear, and a linear model like **linear regression** might not be appropriate. SVM regression is beneficial because:

1. It works well with high-dimensional data.
2. It is effective when the data is noisy and has outliers because SVM can fit the best hyperplane that provides a margin of tolerance (ε-insensitive tube).
3. It helps when you need to predict a continuous target variable, and you want a robust model that generalizes well.

For example, I might have used SVM regression in a project where I predicted house prices based on various factors (e.g., square footage, location, amenities) and needed a model that handles non-linearities and outliers.

**29. Explain the concept of GINI Impurity.**  
**Answer:**  
**GINI Impurity** is a metric used in decision trees to measure the impurity or disorder of a dataset. It is used to evaluate how well a feature splits the data into different classes. The Gini Impurity for a set of classes is defined as:

GINI=1−∑i=1Cpi2GINI = 1 - \sum\_{i=1}^{C} p\_i^2GINI=1−i=1∑C​pi2​

Where:

* CCC is the number of classes,
* pip\_ipi​ is the proportion of instances in class iii.

The Gini impurity ranges from 0 (perfectly pure, all data points are of the same class) to 1 (completely impure, data points are uniformly distributed across all classes). The decision tree algorithm aims to minimize the Gini impurity when making splits at each node.

**30. Let’s suppose I have given you a dataset with 100 columns. How will you be able to control the growth of a decision tree?**  
**Answer:**  
To control the growth of a decision tree when faced with a dataset that has many features (like 100 columns), I would take the following steps:

1. **Limit Maximum Depth**: Set a maximum depth for the tree to prevent it from growing too deep and overfitting the data.
2. **Minimum Samples per Split**: Set a minimum number of samples required to split a node, ensuring that the tree doesn’t split based on too few data points.
3. **Minimum Samples per Leaf**: Set a minimum number of samples that must be in a leaf node to avoid creating overly specific branches.
4. **Pruning**: Apply post-pruning methods, such as **cost-complexity pruning**, to trim branches that add little value.
5. **Feature Selection**: Use feature selection techniques (like **Random Forest** or **L1 regularization**) to reduce the number of irrelevant or redundant features, thus simplifying the tree-building process.
6. **Use Ensemble Methods**: Consider using ensemble methods like **Random Forest** or **Gradient Boosting**, which aggregate multiple smaller trees and help prevent overfitting.

These techniques help in controlling the growth of the decision tree and ensure that it generalizes well to unseen data.

**31. If you are using AdaBoost algorithm & if it is giving you underfitted result, what is the hyperparameter tuning you will do?**  
**Answer:**  
If AdaBoost is giving underfitting results, the following hyperparameters can be tuned to improve the performance:

1. **Increase the number of estimators (n\_estimators)**: By increasing the number of weak learners (usually decision trees), AdaBoost can learn more complex patterns and improve its performance.
2. **Increase the base learner complexity (max\_depth)**: If you're using decision trees as base learners, increase their depth or use more complex models as base learners to capture more complex patterns.
3. **Adjust the learning rate (learning\_rate)**: A higher learning rate may help the algorithm learn faster and better fit the data, but it might lead to overfitting if too high. Experiment with values like 0.1 or 0.5.
4. **Change the base estimator**: Consider using more complex base estimators such as **Random Forest** or **Support Vector Machines (SVM)** as base learners for AdaBoost if the current ones are too simple.

**32. Explain gradient boosting algorithm.**  
**Answer:**  
**Gradient Boosting** is an ensemble learning technique that builds a series of decision trees in a sequential manner, where each tree tries to correct the errors (residuals) made by the previous tree. The algorithm focuses on fitting new models to the residual errors of the previous models, optimizing a loss function through gradient descent.

Steps involved in **Gradient Boosting**:

1. Train an initial model (usually a decision tree).
2. Compute the residual errors from the model’s predictions.
3. Fit a new model to these residual errors.
4. Add the new model’s predictions to the original model.
5. Repeat this process iteratively, gradually improving the model.

It typically leads to better performance than other algorithms, especially in regression tasks and classification problems.

**33. Can we use PCA to reduce dimensionality of highly non-linear data?**  
**Answer:**  
No, **Principal Component Analysis (PCA)** is a **linear** technique, which means it works by identifying the directions (principal components) that capture the most variance in the data. PCA assumes linear relationships between features and might not perform well on **highly non-linear** data. For non-linear data, techniques like **t-SNE (t-Distributed Stochastic Neighbor Embedding)** or **autoencoders** would be better suited to reduce dimensionality while preserving the non-linear structure.

**34. How do you evaluate performance of PCA?**  
**Answer:**  
The performance of **Principal Component Analysis (PCA)** can be evaluated using the following methods:

1. **Explained Variance Ratio**: This measures the proportion of the dataset's variance captured by each principal component. A high explained variance indicates that PCA has effectively reduced dimensionality while retaining most of the important information.
2. **Reconstruction Error**: After dimensionality reduction, you can reconstruct the original data from the reduced data and calculate the error between the reconstructed data and the original data. A smaller reconstruction error means PCA has effectively captured the important features.
3. **Visualization**: Visualizing the data in a 2D or 3D space using the first few principal components can help determine if the reduction makes sense and if clusters or patterns in the data are preserved.

**35. Have you ever used multiple dimensionality techniques in any project? If yes, give reason. If no, where can we use it?**  
**Answer:**  
Yes, I have used multiple dimensionality reduction techniques in a project, particularly **PCA** for initial reduction and then **t-SNE** for further visual exploration. The reason for using both is:

* **PCA**: Helps in reducing dimensions while preserving the most important variance. It is faster and works well with linear data.
* **t-SNE**: Is used after PCA to visualize high-dimensional data in 2D or 3D space, especially for clustering or classification tasks where non-linear relationships are important.

Using multiple techniques ensures that the dimensionality reduction process captures both linear and non-linear structures in the data.

**36. What do you understand by curse of dimensionality? Explain with an example.**  
**Answer:**  
The **curse of dimensionality** refers to the problems that arise when analyzing and organizing data in high-dimensional spaces (with many features). As the number of dimensions increases, the volume of the space grows exponentially, leading to sparsity in the data. This sparsity makes it harder to model the data and can lead to:

1. **Overfitting**: High-dimensional data may result in models that fit the noise in the data rather than the underlying pattern.
2. **Increased computational cost**: The more dimensions there are, the more complex and computationally expensive the model becomes.

**Example**:  
Consider a dataset with 2 dimensions (height and weight) where clustering is relatively simple. If we add hundreds of dimensions (age, income, education level, etc.), the data becomes sparse, and traditional clustering algorithms may not work effectively, as the data points will be far apart in high-dimensional space.

**37. What is the difference between anomaly detection and novelty detection?**  
**Answer:**

* **Anomaly Detection**: Refers to detecting data points that deviate significantly from the norm in a dataset. It focuses on finding outliers in a known dataset, where both normal and anomalous instances are present. It's often used in fraud detection, network security, etc.
* **Novelty Detection**: Refers to identifying previously unseen data points that do not fit the model based on the training data. It is typically used in scenarios where the model is trained only on normal data and needs to detect novel, unseen instances that might be anomalies.

In short, **anomaly detection** works with both normal and anomalous data, while **novelty detection** focuses only on detecting new or unseen instances.

**38. Explain Gaussian Mixture Model.**  
**Answer:**  
A **Gaussian Mixture Model (GMM)** is a probabilistic model that represents a mixture of multiple **Gaussian distributions**. It assumes that the data is generated from a combination of several Gaussian distributions, each with its own mean and variance. GMM is often used in clustering and density estimation tasks, where the goal is to model the data distribution as a combination of Gaussian distributions.

The model uses the **Expectation-Maximization (EM)** algorithm to fit the model to the data by alternating between:

1. **E-step (Expectation)**: Estimating the probabilities that each data point belongs to each Gaussian component.
2. **M-step (Maximization)**: Updating the parameters (mean, variance, and weights) of each Gaussian component.

**39. Give me a list of 10 activation functions with explanation.**

1. **Sigmoid**: A smooth function that maps input values between 0 and 1. It's used in binary classification. σ(x)=11+e−x\sigma(x) = \frac{1}{1 + e^{-x}}σ(x)=1+e−x1​
2. **Tanh**: Similar to the sigmoid but outputs values between -1 and 1. It's often preferred over the sigmoid due to its zero-centered output. tanh⁡(x)=ex−e−xex+e−x\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}tanh(x)=ex+e−xex−e−x​
3. **ReLU (Rectified Linear Unit)**: Outputs the input directly if it is positive, otherwise, it outputs zero. It’s widely used due to its simplicity and efficiency. ReLU(x)=max⁡(0,x)\text{ReLU}(x) = \max(0, x)ReLU(x)=max(0,x)
4. **Leaky ReLU**: A variant of ReLU that allows small negative values for the input. Leaky ReLU(x)=max⁡(αx,x)(α is a small constant)\text{Leaky ReLU}(x) = \max(\alpha x, x) \quad (\alpha \text{ is a small constant})Leaky ReLU(x)=max(αx,x)(α is a small constant)
5. **Softmax**: Used for multi-class classification. It converts logits (raw predictions) into probabilities. Softmax(xi)=exi∑jexj\text{Softmax}(x\_i) = \frac{e^{x\_i}}{\sum\_{j} e^{x\_j}}Softmax(xi​)=∑j​exj​exi​​
6. **Swish**: A self-gated activation function that is a combination of sigmoid and ReLU, yielding better results for deeper networks. Swish(x)=x⋅σ(x)\text{Swish}(x) = x \cdot \sigma(x)Swish(x)=x⋅σ(x)
7. **ELU (Exponential Linear Unit)**: Similar to ReLU but it has an exponential part for negative values to avoid dead neurons. ELU(x)={xif x>0α(ex−1)if x≤0\text{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha (e^x - 1) & \text{if } x \leq 0 \end{cases}ELU(x)={xα(ex−1)​if x>0if x≤0​
8. **SELU (Scaled ELU)**: A scaled version of ELU that self-normalizes the output.
9. **Hard Sigmoid**: An approximation of the sigmoid function, faster to compute but less accurate. Hard Sigmoid(x)=max⁡(0,min⁡(1,0.2x+0.5))\text{Hard Sigmoid}(x) = \max(0, \min(1, 0.2x + 0.5))Hard Sigmoid(x)=max(0,min(1,0.2x+0.5))
10. **Softplus**: A smooth approximation of the ReLU function.

Softplus(x)=ln⁡(1+ex)\text{Softplus}(x) = \ln(1 + e^x)Softplus(x)=ln(1+ex)

**40. Explain neural network in terms of mathematical function.**  
**Answer:**  
A **neural network** can be viewed as a **composition of mathematical functions**. In its simplest form, a neural network consists of layers of nodes (neurons), where each neuron applies a mathematical transformation to its inputs.

For a single neuron in a fully connected layer, the output yyy can be described by:

y=f(w1x1+w2x2+⋯+wnxn+b)y = f(w\_1x\_1 + w\_2x\_2 + \dots + w\_nx\_n + b)y=f(w1​x1​+w2​x2​+⋯+wn​xn​+b)

Where:

* x1,x2,…,xnx\_1, x\_2, \dots, x\_nx1​,x2​,…,xn​ are the inputs,
* w1,w2,…,wnw\_1, w\_2, \dots, w\_nw1​,w2​,…,wn​ are the weights,
* bbb is the bias,
* fff is the activation function.

Each neuron’s output becomes the input to the next layer, and this process continues until the output layer is reached.