DAA - Fibonacci
Q1. What is the time complexity of the Fibonacci algorithm (recursive and iterative)?
Answer:

- **Recursive approach:**
  Time Complexity = **O(2^n)**
  Because each function call branches into two more recursive calls, leading to exponential growth.
- **Iterative approach:**
  Time Complexity = **O(n)**
  Because the function iterates once per Fibonacci term up to *n*.

Q2. How do you calculate the step count in your code?
Answer:

- Step count represents the **number of basic operations** executed during program execution (like additions, assignments, comparisons, etc.).

Q3. What is the difference between dynamic programming and recursion?
Answer:

Recursion	Dynamic Programming
Solves problems by calling itself repeatedly	Solves problems by **storing subproblem results**

Q4. What are the base conditions for Fibonacci?
Answer:
F(0)=0,F(1)=1

Q7. If n = 10, how many recursive calls are made?
Answer:

- In a naive recursive Fibonacci, number of calls ≈ 2^n - 1.
- For `n = 10` → 2^{10} - 1 = 1023 calls approximately.

Q8. What is the Big-O notation used for?
A: To express the **upper bound of time complexity** (worst-case performance) of an algorithm.

Q9. What is the advantage of using dynamic programming for Fibonacci?
A: Saves computation time. Avoids redundant calls. Converts exponential time to linear time.
DAA - KNAPSACK GREEDY ALGORITHM
Q1. What is a greedy algorithm?
A: A greedy algorithm builds up a solution step-by-step, **choosing the best option at each step** with the hope of finding the global optimum.

2. What is the time complexity of your program?
A: Overall ≈ **O(n log n + n × m)** → For small deadlines, roughly **O(n log n)**.
3. What are real-life applications of job sequencing?
A: CPU job scheduling, Resource allocation , Project/task management systems , Manufacturing and machine scheduling
4. What is the 0–1 Knapsack Problem?

- The 0–1 Knapsack problem is an optimization problem where:
  - Each item has a **weight** and a **value**.
  - You must select a subset of items to **maximize the total value**.
  - The total weight must not exceed the knapsack capacity.
  - You **cannot take fractions** of items — only 0 or 1 (take or don't take).

2. What is the difference between 0–1 Knapsack and Fractional Knapsack?

0–1 Knapsack	Fractional Knapsack
Either take or leave the item	Can take fractions
Dynamic Programming / Branch & Bound	Greedy Algorithm

3. What is Dynamic Programming?

- A method to solve problems by breaking them into **overlapping subproblems** and **storing intermediate results** to avoid recomputation.

4. What are the limitations of the 0–1 Knapsack problem?

- The algorithm becomes slow for **large capacity values (W)** because complexity depends on it.
- Only handles integer capacities efficiently.

12. Is 0–1 Knapsack NP-hard?

- Yes, the **0–1 Knapsack** is **NP-hard**, meaning no polynomial-time algorithm exists for all cases

DAA - DYNAMIC KNAPSACK

ML - UBER

Q1. What is the aim of your practical?

To predict the fare amount of an Uber ride from pickup and drop-off locations using machine learning algorithms like Linear Regression and Random Forest Regression.

Q2. What type of learning problem is this — regression or classification?

It's a regression problem because the output (fare amount) is a continuous numeric value.

Q3. What is the dependent (target) variable?

`fare_amount`

Q4. What are the independent variables (features)?

`pickup_datetime`, `pickup_latitude`, `pickup_longitude`, `dropoff_latitude`, `dropoff_longitude`, `passenger_count`, and derived `distance`.

Q5. Why is data preprocessing important?

To clean and prepare data for training — removing null values, outliers, and converting datetime or coordinates into usable numeric features.

Q6. How did you detect outliers?

Using boxplots and statistical methods like the **Interquartile Range (IQR)**.

Q7. How did you handle them?

By removing data points outside the acceptable range.

Q8. What is correlation analysis used for?

To identify the strength and direction of relationships between features and the target variable.

Q9. What is Random Forest Regression?

An **ensemble** of multiple decision trees where each tree predicts independently and results are averaged.

Q10. Why is Random Forest better than Linear Regression?

It can handle non-linear data, missing values, and outliers better and reduces overfitting.

Q.11. What is overfitting?

When a model performs well on training data but poorly on test data

Q32. How can overfitting be prevented?

- Cross-validation
- Regularization (Lasso/Ridge)

Q33. What is cross-validation?

A method to split data into multiple parts for better and more reliable evaluation.

Q34. What is feature scaling?

Rescaling data so all features contribute equally to the model.

ML -EMAIL

Q5. Why is preprocessing important in text classification?

Emails are raw text; preprocessing converts them into a numerical format understandable by machine learning algorithms.

Q6. What preprocessing steps did you perform?

- Converted text to lowercase

- Removed punctuation and stop words
- Tokenized words
- Used **TF-IDF vectorization** to convert text into numerical features

Q7. What is TF-IDF?

TF-IDF = *Term Frequency – Inverse Document Frequency*

It measures how important a word is in a document relative to all documents.

Q9. What is K-Nearest Neighbors (KNN)?

It's a non-parametric algorithm that classifies a sample based on the majority class of its *k* nearest neighbors.

Q10. What is the role of 'k'?

'k' defines how many nearest data points to consider for decision making.

Q11. What distance metric is used in KNN?

Usually Euclidean distance, though Manhattan or Cosine similarity can also be used for text data.

Q13. What is a Support Vector Machine (SVM)?

SVM is a supervised algorithm that finds the **best hyperplane** that separates data points of different classes with the **maximum margin**.

Q14. What is a hyperplane?

A line or plane that divides data into different classes.

Q18. Why is SVM better than KNN here?

Because SVM works efficiently with high-dimensional sparse data (like text), while KNN becomes slow as feature size increases.

ML- NEURAL NETWORK

Q5. What is a neural network?

A computational model inspired by the human brain that consists of interconnected nodes (neurons) that learn complex relationships in data.Q6. What are the main layers in an ANN?

1. **Input Layer** – takes features as input
2. **Hidden Layers** – perform computations and feature learning
3. **Output Layer** – gives final prediction

Q7. What activation functions did you use?

- **ReLU** for hidden layers
- **Sigmoid** for output layer (since it's binary classification)

Q8. Why use ReLU activation?

It speeds up training and avoids the vanishing gradient problem.

Q9. Why use Sigmoid in output layer?

It converts the output to a probability between 0 and 1.

Q10. What is the optimizer used?

Adam (Adaptive Moment Estimation) – efficient and adaptive learning optimizer.

Q11. What loss function did you use?

Binary Cross-Entropy, since it's a binary classification problem.

Q18. What is a confusion matrix?

A 2×2 matrix showing correct and incorrect classifications.

ML- PCA

What is PCA?

PCA (Principal Component Analysis) is a **statistical technique** used to **reduce the number of features (dimensions)** in a dataset while retaining as much variance (information) as possible.

How PCA Works (Step-by-Step)

1. **Standardize the dataset**
   → Scale features so that each has a mean of 0 and standard deviation of 1.
2. **Compute the covariance matrix**
   → Measures how variables vary together.

3. Calculate eigenvalues and eigenvectors
   → Eigenvectors represent principal components, and eigenvalues represent variance captured by each component.
4. **Select top k components**
   → Choose the components that explain the most variance (based on eigenvalues).
5. **Transform data**
   → Multiply the original standardized data by the selected eigenvectors to get new reduced dimensions.

Q11. What are the limitations of PCA?

- Loses interpretability (new components are linear combinations).
- Assumes linearity.
- Sensitive to scaling.

Q12. Can PCA be used for visualization?
Yes — it's commonly used to visualize high-dimensional data in 2D or 3D.

ML- KMEANS
What is Clustering?
Clustering is an **unsupervised learning** technique used to group similar data points into clusters based on similarity or distance.

Q6. What is WCSS?
Within-Cluster Sum of Squares — the sum of squared distances of samples to their closest cluster center.Q10.

Q12. What distance metric does K-Means use?
Euclidean distance.

Why do we scale data before clustering?
To prevent features with large values from dominating the distance calculation.

BT-SMART CONTRACT

What is a Smart Contract?
A **smart contract** is a self-executing piece of code stored on the **blockchain** that automatically enforces and executes terms of an agreement **without third-party intervention**.

Q1. What is Solidity?
A high-level programming language for writing smart contracts on Ethereum.

Q4. What is `msg.sender` and `msg.value`?

- `msg.sender` → the address calling the function.
- `msg.value` → amount of Ether sent in the transaction.

Q5. What is the use of the `payable` keyword?
It allows the function to receive Ether in the transaction.

Q7. Why do we use `require()` in Solidity?
To check conditions before executing code — it helps prevent invalid transactions.

Q8. What is the difference between `public`, `private`, and `view`?

- `public`: accessible from outside and inside contract.
- `private`: only accessible inside the same contract.
- `view`: reads state but does not modify it (no gas cost).

Q10. What happens if you withdraw more than the balance?
The `require` condition fails → transaction is **reverted**.

BT - STUDENT

Q1. What is a structure in Solidity?
A user-defined data type that groups related variables of different types.
Q2. What is the difference between struct and class?
Solidity doesn’t have classes — structs are lightweight containers without methods.
Q3. Why do we use arrays in Solidity?
To store multiple instances of similar data (like student records).
Q4. What is a fallback function?
An unnamed function that executes when no function matches the call or when Ether is sent without data.

Q5. What is the purpose of the `payable` keyword?
It allows the contract to receive Ether.