WEEK-1 QUESTION AND ANSWERS

1.Describe AI and its applications in various fields.

Artificial Intelligence (AI) is a part of computer science.
It creates machines that can think and act like humans.
AI systems can learn from data.
They can make decisions and solve problems.
Here are some simple examples of AI applications:

---

i. Healthcare

- AI can study medical images like X-rays and MRIs.
- It helps doctors detect diseases such as cancer.
- AI can find new drugs and test their effects.

---

ii. Finance

- AI helps in stock trading and market analysis.
- It finds patterns and helps make better financial decisions.

---

iii. Autonomous Vehicles

- AI helps cars drive by themselves.
- It helps them see their surroundings and make quick decisions.

---

iv. Natural Language Processing (NLP)

- AI chatbots talk to people and answer questions.
- AI tools like Google Translate convert one language to another.

---

v. Education

- AI helps students learn at their own pace.
- It checks assignments and gives quick feedback.

---

vi. E-commerce

- AI suggests products based on what users like.
- It manages stock and supply chains efficiently.

---

vii. Manufacturing

- AI robots check products for defects.
- It predicts machine failures to reduce downtime.

---

viii. Robotics

- AI robots do tasks like welding and assembling.
- Some robots talk and help people in hospitals and shops.

---

ix. Agriculture

- AI helps farmers check crop health and predict weather.
- It also helps in smart irrigation and pest control.

2. How AI software development lifecycle is different from traditional software. Explain?

1. Data-Driven

- AI software depends on data.
- It needs large, high-quality data for training and testing.
- Traditional software uses data only as support.

---

2. Experimentation and Iteration

- AI software uses trial and error.

- Developers test many algorithms to find the best one.
- Traditional software follows a fixed plan or design.

---

3. Model Selection

- In AI, choosing the right model is very important.
- It can take a lot of time.
- In traditional software, algorithms are chosen early and rarely change.

---

4. Model Evaluation and Performance

- AI models are tested using accuracy, precision, recall, and F1 score.
- Traditional software is tested using functional requirements.

---

5. Deployment and Maintenance

- AI software needs regular updates and retraining.
- Traditional software needs fewer updates after release.

3. Write steps to create a repository in GitHub and add a file.

1. Sign in to your GitHub account.
   If you don't have one, create it on github.com.
2. Click the "+" icon at the top-right corner.
   Then select **"New repository"**.
3. Enter a **repository name**.
   Avoid spaces and special characters.
4. Add a **description** (optional).
   It helps others understand your project.
5. Choose the **visibility** — public or private.
6. **Initialize** the repository with a **README file**.
   This file explains your project.
7. Add **.gitignore** and a **license** if needed.
   .gitignore decides which files to skip in version control.
8. Click **"Create repository"**.
   Your repository is now ready.

9. To **clone** it to your computer, click the green **"Code"** button.
   Copy the **repository URL** and clone it using Git.

Add a file to the repository

1. Create a new file or copy an existing file into the folder.
   Example: `example.txt`
2. Open your terminal and go to the repository folder.
3. Use the following commands:

```
4. git add example.txt
5. git commit -m "Added example.txt"
```

6. Push your changes to GitHub:

```
7. git push origin master
```

8. Refresh your GitHub repository page.
   You will see the **example.txt** file uploaded.

WEEK-2 QUESTION AND ANSWER

1.Differentiate between supervised machine learning and unsupervised machine learning.

Supervised Learning

- Supervised learning uses labeled data.
- The model learns from labeled examples.
- It takes feedback to check if the prediction is correct.
- The model predicts the output.
- Input data and output data are given to the model.
- The goal is to train the model to predict new data correctly.
- Supervised learning needs supervision.
- Common algorithms: Linear Regression, Logistic Regression, Decision Tree, SVM, KNN.
- It is used for classification and regression problems.

Unsupervised Learning

- Unsupervised learning uses unlabeled data.

- The model does not take feedback.
- It finds hidden patterns in the data.
- Only input data is given to the model.
- The goal is to discover useful insights from unknown data.
- It does not need supervision.
- Common algorithms: Clustering, KNN, Apriori algorithm.
- It is used for clustering and association problems.

---

2. Challenges in Machine Learning

1. Data Quality and Quantity

- Machine learning needs large and clean datasets.
- Collecting quality data is hard.
- Less data can cause poor results.

2. Overfitting and Underfitting

- Overfitting: Model learns too much and performs badly on new data.
- Underfitting: Model learns too little and misses patterns.

3. Interpretability and Explainability

- Complex models are hard to understand.
- It is difficult to explain how they make decisions.

4. **Bias and Fairness**

- Models can learn biases from training data. This may cause unfair or wrong results. Ensuring fairness is a big challenge.

5. Deployment and Maintenance:

- Putting models into real use is difficult.Maintaining them is also complex. Problems like model drift, scaling, and version control must be managed.

3. A dataset is given to you for creating machine learning model. What are the steps followed before using the data for training the model? Eloborate each step.

1. Data Exploration:
First, study the data and understand it.
Check how many rows and columns it has.
Look at data types and distributions using charts like histograms and box plots.

2. Data Cleaning:
Next, clean the data.
Remove wrong or missing values.
Fix errors and make sure data is correct.

3. Data Transformation:
Change the data into a useful format.
You can scale values, normalize data, or create new columns.

4. Feature Selection:
Choose only the important features for the model.
Use methods like correlation or principal component analysis.

5. Data Splitting:
Divide the data into training, validation, and test sets.
Training data is for learning, validation for tuning, and test data for checking performance.

6. Feature Engineering:
Create new useful features.
You can use combinations, polynomial terms, or grouped values.

7. Evaluation Metric:
Pick a method to measure model performance.
Common ones are accuracy, precision, recall, F1 score, and ROC curve.

8. Model Selection:
Finally, choose the best model.
Compare results using the evaluation metrics.

4.Explore different sources of big data in machine learning.

- **Social Media:**
  Sites like Facebook, WhatsApp, Twitter, YouTube, and Instagram produce large amounts of data daily.
- **Sensors:**
  Devices placed in cities or public places collect data on traffic, safety, and environment.
- **IoT Devices:**
  Smart devices like smart TVs, washing machines, and ACs collect and share data.

- **Customer Feedback:**
  Websites like Amazon and Flipkart collect feedback about products and services.
- **E-commerce:**
  Online transactions through credit/debit cards create huge data.
- **GPS:**
  Vehicles with GPS collect data on location, fuel use, and travel time.
- **Transactional Data:**
  Shops and businesses record transactions like sales, date, and location details.

---

5. For the following scenarios you are required to build a predictive model . Which machine learning technique/ algorithm can be applied/ best suited for stated problems. Justify your recommendation.

i. Predicting food delivery time:
Use regression algorithms like Linear Regression or SVR.
Because delivery time is a continuous number.

ii. Predicting fraudulent transactions:
Use classification algorithms like Logistic Regression, Decision Tree, or Random Forest.
Because fraud is a yes/no type result.

iii. Predicting credit card limit:
Use regression algorithms like Linear Regression or SVR.
Because the credit limit is a numerical value.

iv. Predicting natural disasters:
AI can help predict and monitor events like earthquakes, floods, hurricanes, and typhoons.

WEEK-4 QUESTION AN AND ANSWERS

1.Explain univariate & multivariate data types with examples.

Univariate and Multivariate Data Types

Univariate Analysis

- Univariate analysis studies **only one variable** at a time.
- It helps to find **mean, median, mode,** and **standard deviation.**
- Example: A shop wants to study the **sales** of one product.
- They find **average sales, highest sales,** and **lowest sales** using univariate analysis.

Multivariate Analysis

- Multivariate analysis studies **two or more variables** at the same time.
- It helps to find the **relationship** between variables.
- Example: An e-commerce company studies the **number of reviews, ratings,** and **sales.**
- This helps find which products have **best sales** or **best ratings.**

Univariate Data

- Deals with **a single feature** in a dataset.
- It helps to understand the **distribution** of that variable.
- Example: Studying the **ages** of people in a group to know the **age pattern.**
- You can use **histograms** or **box plots** to show this data.

Multivariate Data

- Deals with **two or more variables** together.
- It helps to study **how variables affect each other.**
- Example: Studying houses using **square footage, number of bedrooms,** and **price.**
- This helps to **predict house price** using other details.
- Multivariate analysis is used in **machine learning** and **statistical modeling.**

WEEK-5 QUESTION AND ANSWERS

1.How to Handle Missing Values in a Dataset

Handling missing data is an important step in data preprocessing because missing values can affect the accuracy of analysis or models.

There are several ways to handle missing values:

---

1. Dropping Rows or Columns

- Remove rows or columns that have missing values.
- Use this only when the missing data is very small.

```
df = train_df.dropna()
```

---

2. Imputing Missing Values

- Replace missing values with some estimated value like **mean**, **median**, or **mode**.
- Helps keep all data but may cause bias if data is not random.

Examples:

```
# Replace with mean
train_df['LoanAmount'].fillna(train_df['LoanAmount'].mean(), inplace=True)

# Replace with median
train_df['LoanAmount'].fillna(train_df['LoanAmount'].median(), inplace=True)

# Replace with mode
train_df['Gender'].fillna(train_df['Gender'].mode()[0], inplace=True)
```

---

3. Using Machine Learning Algorithms

- Algorithms like Decision Trees or Random Forests can predict and fill missing values automatically.

---

4. Advanced Imputation Techniques

- Use methods like **Multiple Imputation**, which creates several imputed datasets and combines the results for better accuracy.

---

5. Replacing with Arbitrary Value

- Replace missing values with a constant (like 0 or "Unknown").

```
train_df['Dependents'].fillna(0, inplace=True)
```

---

6. Keeping Missing Values as Is

- If missing data is very small (e.g., less than 3%), you may ignore it.
- Sometimes keeping missing values doesn't affect results much.

 Summary:

Method	Description	When to Use
Drop rows/columns	Remove missing data	When few values are missing
Mean/Median/Mode	Replace with central values	When data is numeric or categorical
Machine Learning	Predict missing data	For advanced models
Multiple Imputation	Create several imputations	When accuracy is important
Arbitrary Value	Fill with 0 or "Unknown"	For simplicity
Keep as Is	Ignore missing data	If very small amount missing

WEEK-6 QUESTION AND ANSWERS

1. Techniques of Cross Validation

Hold Out Method:

- The dataset is divided into two parts — training set and testing set.
- Common splits are 70-30, 75-25, or 80-20.
- The training set is always larger than the test set.
- The model is trained on the training data and tested on the test data.

Leave-One-Out Cross Validation (LOOCV):

- One observation is used for testing, and the rest for training.
- This process repeats for every observation in the dataset.
- Each time, a different observation becomes the test data.
- It gives an accurate result but takes more time.

K-Fold Cross Validation:

- The dataset is divided into *k* equal parts (folds).
- One fold is used as the test set, and the remaining *k–1* folds are used for training.
- The process repeats *k* times.
- The final result is the average of all *k* test results.

---

2. Data Exploration, Pre-Processing, and Splitting

Data Exploration:

- Helps understand the data and find patterns or problems.
- Involves checking missing values, outliers, and relationships.
- Common steps:
    - **Summary Statistics:** Find mean, median, and standard deviation.
    - **Data Visualization:** Use plots or charts to see data patterns.
    - **Handling Missing Data:** Fill or remove missing values.
    - **Feature Exploration:** Study each feature's type and importance.
    - **Outlier Detection:** Find unusual values that can affect the model.

Data Pre-processing:

- Cleans and prepares data for machine learning.
- Common steps:
    - **Data Cleaning:** Remove duplicates and fix errors.
    - **Feature Scaling:** Normalize or standardize numerical data.
    - **Categorical Encoding:** Convert text data to numbers.
    - **Feature Engineering:** Create or modify features to improve accuracy.
    - **Dimensionality Reduction:** Reduce number of features using PCA.
    - **Data Splitting:** Divide data into training, validation, and test sets.

Splitting Data:

- Data is split to measure model performance.
- Common ratio:
  - 70–80% for training
  - 10–15% for validation
  - 10–15% for testing

Training Set: Used to train the model and find patterns.
Validation Set: Used for tuning and preventing overfitting.
Test Set: Used to check final model performance.

---

3. Overfitting vs Underfitting

Aspect	Overfitting	Underfitting
Model Complexity	Model is too complex.	Model is too simple.
Training Performance	Very good (low training error).	Poor (high training error).
Generalization	Poor on new data.	Poor on both training and test data.
Error Trend	Low training error, high test error.	Both training and test errors are high.
Characteristics	Captures noise and fluctuations.	Fails to capture patterns.
Trade-off	High variance.	High bias.
Remedy	Reduce complexity, use regularization.	Increase complexity, collect more data.

WEEK-7 QUESTION AND ANSWERS

Scalars, Vectors, Matrices, Tensors, and Gradients

Scalars:
A scalar is a single number.
It has only magnitude, no direction.
Examples: temperature (25°C), mass (5 kg), speed (60 km/h).

Vectors:
A vector is an ordered set of scalars.
It has both magnitude and direction.
Examples: position vector in 2D → [x, y], displacement, force.

Matrices:
A matrix is a rectangular array of numbers.
It has rows and columns.
It is used for linear transformations and solving equations.
Example:

A=[1234]A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}A=[1324]

Tensors:
Tensors are extensions of scalars, vectors, and matrices.
They are multi-dimensional arrays of numbers.
They represent data with more than two dimensions.
Example: a color image is a 3D tensor (width × height × color channels).

Gradients:
A gradient is a vector that shows the rate of change of a scalar function.
It points in the direction of the steepest increase.
Gradients are used in optimization and machine learning.
In neural networks, the gradient of the loss function updates model parameters (weights and biases).

WEEK-8 QUESTION AND ANSWERS

1. Classification vs Clustering

- Classification is supervised learning.
- It predicts labels for given input data.
- Examples: Decision Tree, Random Forest, Naive Bayes.
- It needs labelled data.
- Clustering is unsupervised learning.
- It groups similar data points.
- Examples: K-means, Hierarchical, DBSCAN.
- It uses unlabelled data to find hidden patterns.

---

2. K-Means and Curse of Dimensionality

- The curse of dimensionality happens when features increase.
- More dimensions need more data.

- It becomes hard to measure distance accurately.
- In K-means, Euclidean distance is used to measure between points.
- When dimensions increase, distance between points increases.
- Points look far even if they are close.
- K-means becomes less accurate in high dimensions.
- Use PCA to reduce dimensions before K-means.
- DBSCAN can also be used as it handles high dimensions better.

---

3. Choosing Number of Clusters in K-Means

Three methods:

1. Elbow Method
2. Silhouette Method
3. Gap Statistic

Elbow Method

- Run K-means for different k values (1 to 10).
- Calculate within-cluster sum of squares (WSS).
- Plot WSS vs number of clusters.
- The “elbow” point in the graph shows the best k.

Silhouette Method

- Run K-means for different k values.
- Compute average silhouette value for each k.
- The highest value shows the best number of clusters.

Gap Statistic Method

- Run clustering for k = 1 to kmax.
- Generate reference datasets.
- Compare Wk (within-cluster variation) with reference data.
- Choose k where the gap statistic is maximum.

---

4. Ensemble Learning Techniques

Boosting

- Boosting trains models one after another.
- Each model focuses on the errors of the previous one.

- It improves accuracy.
- Examples: AdaBoost, Gradient Boosting, XGBoost.

Bagging

- Bagging trains models independently on random data samples.
- It reduces variance and avoids overfitting.
- Example: Random Forest.

Stacking

- Stacking combines multiple models.
- Their predictions are given to another model (meta-model).
- The meta-model gives the final output.

Blending

- Blending is like stacking but uses a validation set.
- It combines different models for better performance.

Voting

- Voting combines predictions from multiple models.
- Final output is based on majority or average vote.
- **Hard Voting:** Takes the most frequent class.
- **Soft Voting:** Takes the class with highest probability.

WEEK-8 QUESTION AND ANSWERS

1. Importance of Dimensionality Reduction in Machine Learning

- Features or input variables help predict the target in machine learning.
- Not all features are equally useful.
- Dimensionality reduction helps find only the important features.
- It reduces the number of features and keeps only the relevant ones.

Importance:

- **To reduce model complexity:**
  Too many features make the model complex. Reducing features helps simplify it.
- **To prevent overfitting:**
  High dimensional data can cause overfitting. Reducing features improves performance on new data.
- **To achieve faster computation:**
  Fewer features mean less time for training and testing.
- **To save storage:**
  Lower dimensions need less memory and storage space.
- **To improve model performance:**
  It removes noise and keeps only useful data.

---

2. What are MLOps? Briefly explain different stages in MLOps lifecycle

- **MLOps** means **Machine Learning Operations**.
- It helps manage, deploy, and monitor ML models.
- It combines **Machine Learning** and **DevOps**.

Stages of MLOps lifecycle:

1. **ML Development:**
   Build a pipeline for data processing, training, and testing.
2. **Model Training:**
   Train the model on data. Update it when new data arrives.
3. **Model Evaluation:**
   Test the model's accuracy and performance.
4. **Model Deployment:**
   Deploy the model for real-world use.
5. **Prediction Serving:**
   The model predicts results from new input data.

6. **Model Monitoring:**
   Check if the model still performs well over time.
7. **Data and Model Management:**
   Manage data, model versions, storage, and security.

WEEK-10 QUESTION AND ANSWERS

1. Activation Functions in Neural Networks

- Activation functions add non-linearity to neural networks.
- They help the network learn complex patterns.
- They decide if a neuron should be activated or not.
- The output of an activation function is sent to the next layer.

Types of Activation Functions:

a. Binary Step Function

- It works on a threshold value.
- If the input is greater than the threshold, the neuron activates.
- If not, it deactivates.
- Formula:
  f(x)=1 for x>0, f(x)=0 for x≤0f(x) = 1 \text{ for } x > 0, \; f(x) = 0 \text{ for } x \le 0f(x)=1 for x>0,f(x)=0 for x≤0

b. Linear Activation Function

- It gives output equal to the input.
- It is also called an identity function.
- Formula: f(x)=xf(x) = xf(x)=x
- It is simple but cannot handle complex data.

c. Sigmoid (Logistic) Function

- It outputs values between 0 and 1.
- It is useful for probability prediction.
- Formula: f(x)=11+e−xf(x) = \frac{1}{1 + e^{-x}}f(x)=1+e−x1
- It gives smooth output and helps avoid sudden jumps.

d. Tanh (Hyperbolic Tangent) Function

- It outputs values between -1 and +1.
- It is zero-centered.
- Formula: f(x)=ex−e−xex+e−xf(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}f(x)=ex+e−xex−e−x
- It is used in hidden layers.
- It helps the network learn faster.

e. ReLU (Rectified Linear Unit) Function

- It gives output as f(x)=max⁡(0,x)f(x) = \max(0, x)f(x)=max(0,x).
- If input < 0, output = 0.
- It is simple and fast.
- It helps in faster learning.
- Only some neurons activate at a time, making it efficient.

f. Leaky ReLU Function

- It improves ReLU by fixing the "dead neuron" problem.
- It allows small output for negative inputs.
- Formula: f(x)=xf(x) = xf(x)=x if x>0x > 0x>0; else f(x)=0.01xf(x) = 0.01xf(x)=0.01x.
- It helps learning even when inputs are negative.

---

2. Neural Network Architecture

- Neural network architecture means its structure.
- It shows how layers, neurons, and connections are arranged.
- Each neuron gets input, does some math, and gives output.
- Layers are:
  - Input layer – takes raw data.
  - Hidden layers – process data.
  - Output layer – gives the result.
- Weights and biases adjust during training to reduce error.
- Common types:
  - CNN (Convolutional Neural Network)
  - RNN (Recurrent Neural Network)

---

3. Strategies of Production Deployment

a. Canary Deployment

- New version is released to a few users first.
- Feedback is checked before full release.

b. Shadow Deployment

- New version runs beside the old one.
- It gets real data but doesn't affect users.
- Used to test performance safely.

c. Blue/Green Deployment

- Two setups: Blue (testing) and Green (production).
- After testing in Blue, traffic moves to Green.

d. Rolling Deployment

- New version releases in parts to few servers.
- Gradually, it replaces the old version on all servers.

e. A/B Testing

- Two versions (A and B) are tested on different user groups.
- The better one is chosen based on performance.

WEEK-12 QUESTION AND ANSWERS

Components of Docker

1. **Docker Client**

- The Docker client sends commands to the Docker daemon.
- It uses commands and REST APIs.
- When a user runs a Docker command, it goes to the Docker daemon.
- The daemon receives and processes the command.
- Common commands are:
    - `docker build`
    - `docker pull`
    - `docker run`

---

2. **Docker Registry**

- The Docker Registry stores Docker images.
- There are two types of registries:
    - **Public Registry** (Docker Hub)
    - **Private Registry** (used within an organization)

3. **Docker Daemon**

- It runs in the background.
- It manages containers on the host machine.
- It handles creating, starting, stopping, and removing containers.
- It also manages network and storage resources.

---

4. **Docker Images**

- Docker images are templates used to create containers.
- They are read-only.
- Images can be shared using public or private registries.
- Each image contains everything needed to run an application.

---

5. **Docker Containers**

- Containers are units that run the application.
- They hold all files and dependencies needed.
- Containers use very few resources.
- Each container is a copy of an image.

Cloud Deployment Models

1. Public Cloud

- Public cloud is open to everyone.
- It provides access to systems and services over the internet.
- Example: Google App Engine.

Advantages:

- Less investment.
- No need for infrastructure management.
- No maintenance for users.
- Easy scalability.

Disadvantages:

- Less secure.
- Low customization.

---

2. Private Cloud

- Private cloud is for a single user or organization.
- It is more secure.
- It is also called an internal cloud.

Advantages:

- Better control.
- High data security and privacy.
- Works with legacy systems.
- Allows customization.

Disadvantages:

- Less scalable.
- More expensive.

Ethical Challenges in AI

1. **Bias and Discrimination**

- AI can show bias if trained on biased data.
- It can cause unfair decisions in hiring or lending.

2. **Privacy and Security**

- AI collects and uses personal data.
- It can cause privacy issues.

3. **Lack of Transparency**

- AI systems are complex.
- It is hard to understand how they make decisions.

4. **Job Loss**

- AI can replace human jobs.
- This can cause unemployment.

5. **Autonomy and Control**

- AI can make its own decisions.
- It raises questions about control and responsibility.

6. **Ethical Dilemmas**

- AI may face tough choices, like in self-driving cars.
- It is hard to decide the right action.

7. **Societal Impact**

- AI affects society, economy, and ethics.
- It must be used responsibly.

Ethics in AI

- Ethics in AI means following moral rules while developing AI.
- It ensures AI is fair, transparent, and safe.
- It respects human rights and values.

Reasons for Ethical Practices:

- To prevent harm to people.
- To avoid bias and discrimination.
- To make AI transparent and explainable.
- To make AI reliable and safe.
- To respect human dignity and well-being.