1. What are Python's key features?

- Simple syntax, easy to learn (Beginner-friendly).
- Interpreted & dynamically typed (no need to define variable types).
- Supports OOP, vast libraries (NumPy, Pandas, etc.).

✅ 2. Difference between list, tuple, and set

- **List**: Mutable, ordered, allows duplicates – `[ ]`
- **Tuple**: Immutable, ordered, allows duplicates – `( )`
- **Set**: Mutable, unordered, no duplicates – `{}`

✅ 3. What is the use of `self` in Python?

- `self` refers to the current object of the class.
- It's used to access instance variables/methods.
- Har method ka pehla parameter `self` hona chahiye.

✅ 4. Explain Python's memory management.

- Python uses **reference counting** and **garbage collection**.
- `gc` module manages unused memory automatically.
- Memory blocks are managed in **private heaps**.

✅ 5. What are Python modules and packages?

- **Module**: Ek single `.py` file with functions/classes.
- **Package**: A folder with `__init__.py`, contains multiple modules.
- Reusability ke liye dono ka use hota hai.

✅ 6. What is the difference between `is` and `==`?

- `==` compares **values** (content).
- `is` compares **object identity** (memory address).
- Example: `a == b` true, but `a is b` false (if not same object).

✅ 7. What is list comprehension?

- A compact way to create lists.
- Syntax: `[expression for item in iterable if condition]`
- Example: `[x*x for x in range(5)]` → `[0,1,4,9,16]`

✅ 8. What is a lambda function?

- Anonymous function defined using `lambda`.
- Used for short, one-line functions.
- Example: `lambda x: x+5`

✅ 9. Explain map(), filter(), and reduce()

- **map()**: Applies function to all elements → returns new iterable.
- **filter()**: Selects elements that match a condition.
- **reduce()**: Combines all items using a function (from `functools`).

✅ **10. What are *args and kwargs?

- `*args`: Accepts **multiple positional arguments**.
- `**kwargs`: Accepts **multiple keyword arguments**.
- Used when number of inputs is not fixed.

✅ 11. What is an iterator and a generator?

- **Iterator**: Object with `__iter__()` and `__next__()` methods.
- **Generator**: Special iterator using `yield`, memory-efficient.
- Generator lazily produces values one by one.

✅ 12. Explain Python’s OOP concepts.

- Python supports **OOP**: Encapsulation, Inheritance, Polymorphism, Abstraction.
- Classes define templates, objects are instances.
- Reusability and modularity are the goals.

✅ 13. Difference between @staticmethod and @classmethod

- `@staticmethod`: No access to class (`cls`) or instance (`self`).
- `@classmethod`: Takes `cls` as the first parameter.
- Staticmethod is utility; classmethod works with class state.

✅ 14. Explain exception handling in Python.

- Use `try`, `except`, `else`, and `finally` blocks.
- Handles runtime errors gracefully.
- Example: `try: 1/0 except ZeroDivisionError: print("Can't divide")`

✅ 15. What is the LEGB rule in Python?

- Python resolves variable names using **LEGB**:
  - **L**ocal → **E**nclosing → **G**lobal → **B**uilt-in.
- Closest scope value is picked first.

✅ 🔷 Pandas Interview Questions

1. What is Pandas in Python?

Answer:
Pandas is a data manipulation and analysis library. It provides two main data structures — `Series` (1D) and `DataFrame` (2D table). It's widely used for data cleaning, filtering, aggregation, and visualization.

---

2. Difference between Series and DataFrame?

Answer:

- **Series**: One-dimensional labeled array.
- **DataFrame**: Two-dimensional table (like Excel sheet) with rows and columns.
- Series = 1 column, DataFrame = multiple columns.

---

3. How do you handle missing data in Pandas?

Answer:

- `df.isnull()`: Check nulls
- `df.dropna()`: Remove missing values
- `df.fillna(value)`: Fill with custom value or mean/median

---

4. How to filter rows in a DataFrame?

Answer:
Use boolean indexing:

python

CopyEdit

```
df[df['age'] > 25]
```

This filters rows where age > 25.

---

5. What does `groupby()` do?

Answer:
`groupby()` groups rows based on column values, then performs operations like `sum()`, `mean()`, `count()`, etc.
E.g., `df.groupby('department').sum()`

✅ 🔶 NumPy Interview Questions

6. What is NumPy and why is it used?

Answer:
NumPy stands for **Numerical Python**. It's used for fast mathematical operations on arrays and matrices. It's the foundation of scientific computing in Python.

---

7. What is the difference between list and NumPy array?

Answer:
NumPy arrays are faster, more memory-efficient, and support element-wise operations. Python lists are slower and don't support vectorized math.

---

8. How to create a NumPy array?

Answer:

python

CopyEdit

```
import numpy as np

a = np.array([1, 2, 3])
```

---

9. How to create arrays filled with zeros, ones, or random numbers?

Answer:

python

CopyEdit

```
np.zeros((2, 3))     # 2x3 zeros

np.ones((3, 1))      # 3x1 ones

np.random.rand(2, 2) # Random 2x2 array
```

---

10. What are broadcasting rules in NumPy?

Answer:
Broadcasting allows arithmetic between arrays of different shapes. It stretches smaller arrays without copying data — fast and memory-efficient.

✅ 🔵 TensorFlow Interview Questions

11. What is TensorFlow?

Answer:
TensorFlow is an open-source deep learning framework by Google. It’s used to build and train neural networks using computational graphs and tensors.

---

12. What is a tensor in TensorFlow?

Answer:
A tensor is a multi-dimensional array (like a matrix or vector). It's the core data structure used in TensorFlow for all computations.

---

13. What is the difference between `tf.Variable` and `tf.constant`?

Answer:

- `tf.Variable`: Changeable during training (e.g., weights)
- `tf.constant`: Fixed, read-only values (e.g., input constants)

---

14. What is a computational graph?

Answer:
It's a graph where nodes are operations and edges are tensors. TensorFlow builds and runs this graph to perform computations efficiently.

---

15. What are some common layers in a neural network using TensorFlow/Keras?

Answer:

- `Dense`: Fully connected layer
- `Conv2D`: For images
- `LSTM`: For sequences
- `Dropout`: Regularization
  All used in `Sequential()` or `Model()` APIs.

---

16. How do you compile and train a model in Keras?

Answer:

python

CopyEdit

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(X_train, y_train, epochs=10)
```

---

17. How do you save and load a TensorFlow model?

Answer:

python

CopyEdit

```
model.save('my_model.h5')         # Save

model = tf.keras.models.load_model('my_model.h5')  # Load
```

AIML INTERVIEW QUESTIONS

✅ 1. What is the difference between AI, ML, and DL?

- **AI (Artificial Intelligence)**: Machines that mimic human intelligence.
- **ML (Machine Learning)**: Subset of AI that learns from data.
- **DL (Deep Learning)**: Subset of ML using neural networks.
- Example: AI is the umbrella, ML is inside AI, DL is inside ML.

✅ 2. What are supervised, unsupervised, and reinforcement learning?

- **Supervised**: Labeled data (e.g., spam detection).
- **Unsupervised**: No labels (e.g., customer clustering).
- **Reinforcement**: Learn from reward/punishment (e.g., game AI).
- Teenon ka goal learning hai, but process alag hota hai.

✅ 3. What is overfitting and underfitting?

- **Overfitting**: Model learns noise, high accuracy on training but poor on test data.
- **Underfitting**: Model too simple, misses patterns in data.
- Overfit = too much learning, Underfit = too little learning.
- Aim is to find a balance using validation techniques.

✅ 4. Explain bias-variance trade-off.

- **Bias**: Error from overly simplistic model (underfitting).
- **Variance**: Error from too complex model (overfitting).
- Bias ↑ → simple model, Variance ↑ → complex model.
- Good ML model maintains a balance (low bias + low variance).

✅ 5. What is a confusion matrix?

- It's a 2x2 table showing predictions vs actual values.
- Helps evaluate classification model performance.
- Shows TP, TN, FP, FN values.
- Used to calculate accuracy, precision, recall.

✅ 6. What is accuracy, precision, recall, and F1 score?

- **Accuracy** = (TP+TN)/Total
- **Precision** = TP / (TP + FP) → exactness
- **Recall** = TP / (TP + FN) → completeness
- **F1 Score** = Harmonic mean of precision & recall

✅ 7. What is the difference between classification and regression?

- **Classification**: Predict categories (e.g., pass/fail, spam/ham).
- **Regression**: Predict continuous values (e.g., house price).
- Classification = discrete output, Regression = numeric output.

- Algorithms are different for both types.

✅ 8. What is the role of a cost function?

- Measures how wrong the model's predictions are.
- Model tries to minimize this error.
- Common cost functions: MSE (for regression), log loss (for classification).
- Lower cost = better model.

✅ 9. What is gradient descent?

- Optimization algorithm to minimize the cost function.
- It updates weights based on slope (gradient).
- Takes small steps in the direction of minimum error.
- Used in most ML and deep learning models.

✅ 10. What are hyperparameters? Give examples.

- Settings you define **before training**, not learned from data.
- Examples: learning rate, number of trees, number of layers.
- Affect training process and performance.
- Tuned using techniques like GridSearch or RandomSearch.

✅ 11. What is feature scaling? Why is it important?

- Standardizes data to the same range or scale.
- Helps models like KNN, SVM, Gradient Descent work properly.
- Without scaling, large features dominate learning.
- Common methods: MinMaxScaler, StandardScaler.

✅ 12. Explain k-NN algorithm in simple words.

- Stands for **K-Nearest Neighbors**.
- Finds the ‘k’ closest points to a new data point.
- Majority vote of neighbors decides the class.
- Simple and effective for small datasets.

✅ 13. What is cross-validation?

- Technique to test model performance more reliably.
- Splits data into k-parts → train on k-1, test on 1 → repeat.
- Common method: k-fold cross-validation.
- Reduces overfitting and ensures generalization.

✅ 14. What are the differences between bagging and boosting?

Bagging	**Boosting**

Builds models in parallel	Builds models sequentially
Reduces variance	Reduces bias
E.g., Random Forest	E.g., AdaBoost, XGBoost
Stable and simple	More complex but powerful

✅ 15. Name a few Python libraries used in ML

- **Scikit-learn**: For ML algorithms
- **Pandas/Numpy**: For data handling
- **Matplotlib/Seaborn**: For visualization
- **TensorFlow / PyTorch**: For deep learning

✅ Project 1: Digit Recognition System

Key Technologies: Python, MNIST dataset, CNN (Convolutional Neural Network), TensorFlow/Keras, OpenCV

---

1. What is the MNIST dataset?

Answer:
 MNIST is a collection of 70,000 handwritten digit images (0–9), used for training image classification models. Each image is 28x28 pixels and grayscale.

---

2. Why do we use CNN for digit recognition?

Answer:
 CNNs are best for image data because they capture spatial features using convolutional layers. They extract edges, curves, and patterns — essential for recognizing digits.

---

3. What are the main layers in a CNN?

Answer:

- **Convolutional Layer**: Feature extraction
- **ReLU**: Adds non-linearity
- **Pooling Layer**: Downsamples feature maps
- **Fully Connected Layer**: Final classification

---

4. What is the purpose of flattening in CNNs?

Answer:
 Flattening converts the 2D feature maps into a 1D vector before passing it to the fully connected layers.

---

5. How does OpenCV help in digit recognition?

Answer:
OpenCV helps capture or preprocess images (e.g., resizing, grayscale conversion, thresholding) before passing them to the trained model.

---

6. What activation function is commonly used in digit classification output?

Answer:
Softmax is used in the output layer to predict the probability distribution over 10 digit classes (0 to 9).

✅ Project 2: AI-Powered Image Caption Generator

Key Technologies: CNN + LSTM, Feature Extraction (VGG16/ResNet), Natural Language Processing, TensorFlow/Keras, Tokenizer, Beam Search

---

7. How does an image caption generator work?

Answer:
It extracts visual features from an image using a CNN and generates a sentence using an LSTM. It combines computer vision and NLP.

---

8. Why do we use CNN and LSTM together?

Answer:
CNN handles **image feature extraction**, while LSTM generates **text sequences** (captions) by understanding temporal dependencies in words.

---

9. What is transfer learning and how is it used in this project?

Answer:
Transfer learning uses pre-trained models like **VGG16** or **ResNet** to extract features from images instead of training CNN from scratch. It saves time and improves accuracy.

10. What is a tokenizer in caption generation?

Answer:
Tokenizer converts words into integers (token IDs) and helps in preparing input sequences for the LSTM model.

11. What is Beam Search and why is it used?

Answer:
Beam Search generates the best possible sentence by exploring multiple paths and keeping top k predictions at each time step. It gives better captions than greedy search.

12. Why is padding required in caption generation?

Answer:
Sentences vary in length. Padding ensures all sequences are of the same length before training the LSTM.

13. How do we evaluate a captioning model?

Answer:
Using metrics like **BLEU Score**, which compares generated captions to reference (human-written) captions.

14. What is the role of embedding in NLP?

Answer:
Embeddings convert tokenized words into dense vector representations that capture semantic meaning, which LSTM can process effectively.