

Topic: Python

Emotion: Confident

Q1: Explain the Global Interpreter Lock (GIL) in Python and how it affects concurrency.

Q2: Describe how memory management works in Python.

Q3: What are metaclasses and how are they used?

Q4: How would you optimize performance in large-scale Python applications?

Q5: Explain the differences between shallow and deep copies in Python.

Q6: What are coroutines and how do they differ from generators?

Q7: How does the `@property` decorator work and when would you use it?

Q8: How do you implement operator overloading in Python?

Q9: How does Python's garbage collection mechanism work?

Q10: What are descriptors and how are they used?

Q11: How would you handle circular imports in a Python project?

Q12: What is monkey patching and when is it appropriate to use?

Q13: How can you create an immutable class in Python?

Q14: How would you manage state in a long-running Python process?

Q15: What are the different scopes in Python and how are they resolved?

Q16: How does the `with` statement work and how is it implemented?

Q17: What is the purpose of `__slots__` and how does it affect memory usage?

Q18: How do you implement multithreading and multiprocessing in Python?

Q19: What are some real-world applications of decorators?

Q20: Explain Python's MRO (Method Resolution Order).

Q21: How would you build a plugin architecture in Python?

Q22: Explain how to use the `typing` module effectively.

Q23: What is duck typing? Provide an example.

Q24: How would you implement custom iterators?

Q25: How do you enforce encapsulation in Python?

- Q26: Explain how Python handles pass-by-reference vs. pass-by-value.
- Q27: What are the differences between Python 2 and Python 3?
- Q28: How can you debug memory leaks in Python?
- Q29: What is `__new__` and how does it differ from `__init__`?
- Q30: How would you use contextvars in async code?
- Q31: How do you write thread-safe code in Python?
- Q32: What are some performance profiling tools in Python?
- Q33: What is structural pattern matching introduced in Python 3.10?
- Q34: How can you sandbox Python code securely?
- Q35: How would you implement a plugin discovery mechanism?
- Q36: What's the difference between `yield` and `return`?
- Q37: How can Python integrate with C/C++?
- Q38: What's the role of `__getattr__`, `__getattribute__`, and `__setattr__`?
- Q39: How do you ensure immutability in data classes?
- Q40: Explain `functools.lru_cache` and its use cases.
- Q41: How can you build a command-line tool using `argparse`?
- Q42: What are asynchronous context managers?
- Q43: How would you organize a large Python codebase?
- Q44: How do you implement caching in Python?
- Q45: What are sentinel values and where are they used?
- Q46: How do you use Python's `dataclasses` module?
- Q47: What are weak references and when are they useful?
- Q48: How do you validate user input in CLI applications?
- Q49: What are Python wheels and how do they work?
- Q50: How would you create a DSL (Domain-Specific Language) in Python?

Topic: Python

Emotion: Nervous

- Q1: What is a list in Python and how do you create one?
- Q2: What is the difference between a list and a tuple?
- Q3: How do you write a for loop in Python?
- Q4: What are Python functions and how do you define one?
- Q5: What is the difference between `is` and `==`?
- Q6: How do you handle exceptions in Python?
- Q7: How do you import a module in Python?
- Q8: What is the difference between `None` and `False`?
- Q9: What are dictionaries and how do you use them?
- Q10: What is indentation and why is it important in Python?
- Q11: What are the common built-in data types in Python?
- Q12: How do you take input from a user in Python?
- Q13: What does the `len()` function do?
- Q14: What are conditional statements in Python?
- Q15: How do you check for membership in a list?
- Q16: What does the `break` statement do?
- Q17: How do you define a class in Python?
- Q18: What are instance and class variables?
- Q19: What is the `self` keyword?
- Q20: How do you read and write files in Python?
- Q21: What is a module vs. a package?
- Q22: What does the `range()` function return?
- Q23: What are Boolean values in Python?
- Q24: What is a `while` loop?
- Q25: What is string formatting?
- Q26: How do you comment in Python?

Q27: What is a Python script?

Q28: How do you create a list comprehension?

Q29: What are ``if``, ``elif``, and ``else``?

Q30: How do you convert a string to an integer?

Q31: What is the output of ``print(2 ** 3)``?

Q32: What are the logical operators in Python?

Q33: What does ``int()`` do?

Q34: What does the ``append()`` method do?

Q35: What does ``strip()`` do in a string?

Q36: How do you reverse a list?

Q37: What is slicing in Python?

Q38: How do you define a function with default parameters?

Q39: How do you create a dictionary?

Q40: What does ``pop()`` do on a list?

Q41: What is the difference between ``input()`` and ``raw_input()``?

Q42: What does ``in`` mean in Python?

Q43: What is ``NoneType``?

Q44: How do you remove an element from a list?

Q45: How do you convert a list to a set?

Q46: What is a lambda function?

Q47: How do you return a value from a function?

Q48: What is ``enumerate()`` used for?

Q49: What are comments and how do you write them?

Q50: How do you install a package using pip?

Topic: JavaScript

Emotion: Confident

Q1: Explain the event loop in JavaScript and how it handles asynchronous operations.

Q2: What is the difference between `==` and `===` in JavaScript?

Q3: Describe the concept of closures with examples.

Q4: How does prototypal inheritance work in JavaScript?

Q7: What are promises and how do they work under the hood?

Q8: What is hoisting in JavaScript?

Q9: How does the call stack work in JavaScript?

Q10: How does `async/await` differ from using `.then()`?

Q11: What is the purpose of the `Symbol` type?

Q12: Explain the concept of a JavaScript engine and JIT compilation.

Q13: What are Web Workers and how do they work?

Q14: How do you handle memory leaks in JavaScript?

Q15: What is the difference between shallow copy and deep copy in JS?

Q16: Explain the concept of reflow and repaint in the browser rendering process.

Q17: What are the different types of scopes in JavaScript?

Q18: How can you implement a debounce function in JavaScript?

Q19: Explain what currying is and its use cases.

Q20: How does garbage collection work in JavaScript?

Q21: What are generators in JavaScript?

Q22: How do you polyfill a feature in JavaScript?

Q23: What are WeakMap and WeakSet used for?

Q24: What is the difference between `Object.freeze` and `Object.seal`?

Q25: What is a service worker and how does it relate to PWA?

Q26: How do modules work in modern JavaScript?

Q27: Explain dynamic imports and code splitting.

Q28: What is the difference between synchronous and asynchronous execution?

Q29: What are event bubbling and capturing?

Q30: How do you handle deep cloning of objects?

Q31: What is the difference between `setTimeout` and `requestAnimationFrame`?

Q32: How would you secure client-side JavaScript applications?

Q33: What are the benefits of using functional programming in JS?

Q34: What are the main principles behind reactive programming?

Q35: How would you implement lazy loading for images?

Q36: What's the purpose of the `Reflect` API?

Q37: What is tail call optimization?

Q38: How does lexical scoping work in JavaScript?

Q39: What's the difference between `var`, `let`, and `const`?

Q40: What is the Temporal API and how is it different from Date?

Q41: How do you implement a pub-sub pattern in JavaScript?

Q42: What is the difference between microtasks and macrotasks?

Q43: What are Proxy objects and how can they be used?

Q44: How do you build a virtual DOM from scratch?

Q45: What are the advantages of using ES Modules?

Q46: What is module bundling and why is it important?

Q47: How does the `new` keyword work internally?

Q48: How would you create a custom event?

Q49: What is the impact of strict mode?

Topic : JavaScript

Emotion: Nervous

Q1: What is JavaScript used for?

Q2: How do you declare a variable in JavaScript?

Q3: What are the different data types in JavaScript?

Q4: How do you write a function in JavaScript?

Q5: What does the `console.log()` function do?

Q6: What is the difference between `null` and `undefined`?

Q7: How do you write an if-else statement in JavaScript?

Q8: What are arrays and how do you create them?

Q9: How do you loop through an array in JavaScript?

Q10: What is a for loop?

Q11: What is an object in JavaScript?

Q12: How do you access object properties?

Q13: What is a string in JavaScript?

Q14: How do you combine two strings?

Q15: What is the `length` property used for?

Q16: How do you check if two values are equal?

Q17: What does the `typeof` operator do?

Q18: How do you comment in JavaScript?

Q19: What is an event in JavaScript?

Q20: How do you add a click event to a button?

Q21: What is DOM in JavaScript?

Q22: How do you select an element by ID?

Q23: What does `addEventListener()` do?

Q24: What is an array method? Give an example.

Q25: What does `push()` do?

Q26: What is a loop?

Q27: What is a conditional statement?

Q28: How do you write a while loop?

Q29: What is `NaN` in JavaScript?

Q30: What are truthy and falsy values?

Q31: How do you create a constant?

Q32: What is the difference between `let` and `var`?

Q33: What does `parseInt()` do?

- Q34: How do you make an alert pop up?
- Q35: What does `return` do in a function?
- Q36: What is a boolean in JavaScript?
- Q37: What is a callback function?
- Q38: What are some string methods you know?
- Q39: How do you convert a string to uppercase?
- Q40: How do you remove the last item in an array?
- Q41: How do you check the type of a variable?
- Q42: What is a ternary operator?
- Q43: What is a template literal?
- Q44: What is the difference between `forEach` and `map`?
- Q45: How do you round a number?
- Q46: What is a syntax error?
- Q47: What is the use of `break` in loops?
- Q48: How do you make a number into a string?
- Q49: What is the difference between `++i` and `i++`?
- Q50: How do you get the current date in JavaScript?
- Q51: How does JavaScript handle floating-point precision?

Topic: JavaScript

Emotion : Curious

1. Why is JavaScript called a scripting language?
2. What exactly happens when I type a script tag in HTML?
3. What is the difference between undefined and not defined?
4. Why is JavaScript single-threaded?
5. What is the purpose of setTimeout inside a for loop?
6. How does JavaScript handle floating-point arithmetic?
7. What's really happening with closures under the hood?

8. Can JavaScript ever run truly multithreaded?
 9. Why do arrays in JS start at index 0?
 10. What's the difference between map and forEach—and why does it matter?
 11. How do async/await and Promises really differ?
 12. Is JavaScript compiled or interpreted?
 13. What is the purpose of the event loop in plain terms?
 14. How do JavaScript engines like V8 optimize code?
 15. Why are functions “first-class citizens” in JS?
 16. What's the exact difference between deep vs. shallow comparison?
 17. What are the historical reasons for var, let, and const?
 18. How is memory allocated and garbage collected in JS?
 19. What is tail call optimization and does it work in JS?
- What happens when you `JSON.stringify` a function?

Topic : JavaScript

Emotion: Stressed

- ❑ Can you show me a working example of a closure?
- ❑ What's the simplest way to debug a JavaScript error?
- ❑ How do I stop my loop from freezing the browser?
- ❑ I keep getting “undefined”—why?
- ❑ What's a quick way to check if an object is empty?
- ❑ What is the simplest way to deep clone an object?
- ❑ How can I find out what type a variable is?
- ❑ What are the most common mistakes new JS devs make?
- ❑ Is there a quick cheat sheet for array methods?
- ❑ What's the fastest way to filter items in an array?
- ❑ I don't get hoisting—can you explain it visually?
- ❑ How do I cancel a `setTimeout` or `setInterval`?

- 🔗 My event listeners don't work—what should I check first?
- 🔗 What's the difference between == and === again?
- 🔗 When should I use let vs const in practice?
- 🔗 What's the best way to handle errors in async functions?
- 🔗 How do I avoid callback hell?
- 🔗 What's a safe way to access nested object properties?
- 🔗 I'm confused between null and undefined—help?
- 🔗 What's the simplest way to make an HTTP request in JS?

Topic :React

Emotion : Confident

1. How does React manage reconciliation?
2. What's the difference between useMemo and useCallback?
3. How does React batching work under the hood?
4. Explain how React Fiber architecture works.
5. What are React Concurrent Features?
6. When would you prefer controlled vs uncontrolled components?
7. Explain useEffect vs useLayoutEffect with examples.
8. How does React handle error boundaries internally?
9. How do you create a performant virtualized list?
10. Describe how context API differs from Redux in memory consumption.
11. How do you optimize rendering performance in large apps?
12. What are Suspense and React.lazy doing behind the scenes?
13. How would you write a custom hook for debounced search?
14. How can you implement code splitting in a React app?
15. What are the performance implications of useEffect without dependencies?
16. How does server-side rendering affect hydration?

17. What's the best way to memoize expensive calculations in React?
18. How do React portals work and when would you use them?
19. How does strict mode help in React?
20. How would you build a form library from scratch in React?

Topic : React

Emotion : Nervous

1. What is a React component?
2. How do you render something in React?
3. What's JSX and why do we use it?
4. What's the difference between props and state?
5. How do I pass data between components?
6. Why does my component re-render?
7. How do I use useEffect without crashing my app?
8. What's the easiest way to conditionally render something?
9. How do I update state correctly in React?
10. What's the best way to loop over data in JSX?
11. Why is my prop undefined?
12. How do I show a loading spinner while fetching data?
13. What is the key prop and why does React want it?
14. How do I prevent default form submission?
15. How do I create a button that does something when clicked?
16. What's the difference between class and function components?
17. How do I install React and start a project quickly?
18. How do I debug a blank screen in React?
19. How do I handle forms in React?
20. How do I fetch data using fetch() in React?

Topic : React

Emotion : Curious

21. Why does React use a virtual DOM instead of manipulating the DOM directly?
22. How does React's diffing algorithm work?
23. Why do we need to use keys when rendering a list?
24. What's the actual difference between mounting and updating in React?
25. Can you nest useEffect calls?
26. How do React hooks internally work?
27. Why is React considered declarative?
28. What happens if you update state too often?
29. How does React maintain component identity?
30. Can useState persist between page reloads?
31. Why doesn't useEffect run on initial render?
32. What's the lifecycle equivalent of useEffect?
33. What happens if I mutate state directly?
34. Why doesn't React batch state updates inside timeouts?
35. What's the difference between SSR and SSG in React?
36. How do React apps scale across teams?
37. What's React's stance on global state vs component state?
38. How do React DevTools actually hook into React?
39. How are React internals different from Preact or Vue?
40. What will React 19 introduce?

Topic : React

Emotion : Stressed

1. How do I stop useEffect from running twice?
2. Why does my state not update immediately?
3. Why is my component stuck in an infinite loop?
4. How do I show a modal conditionally?
5. How do I debug "Objects are not valid as a React child"?
6. What's the easiest way to create reusable components?
7. How do I deal with props drilling?
8. What's the simplest way to use context API?
9. How do I keep form data in sync with state?
10. Why is useEffect not firing?
11. How do I avoid multiple re-renders?
12. My component won't render—where do I start?
13. How do I properly set up a controlled input?
14. Why does the component unmount and remount suddenly?
15. What is causing "Too many re-renders"?
16. How do I debounce an input in React?
17. Why is my event handler not firing?
18. How do I debug state that doesn't persist?

19. Why is my component not updating on prop change?
20. How do I organize my code to avoid confusion?

SQL

Confident

1. How would you optimize a slow-running SQL query with multiple joins?
2. Can you explain the use and performance benefits of indexes in SQL?
3. How do window functions differ from GROUP BY?
4. What's the difference between a clustered and non-clustered index?
5. Explain the use of CTEs vs subqueries with examples.
6. How would you design a normalized database for an e-commerce platform?
7. What's the difference between RANK(), DENSE_RANK(), and ROW_NUMBER()?
8. How do transactions work and how do you manage deadlocks?
9. What are the ACID properties and why do they matter?
10. How do you enforce data integrity in relational databases?
11. What are some strategies for partitioning large datasets?
12. How does query execution plan analysis help performance tuning?
13. How would you implement soft deletes in SQL?
14. When would you denormalize a schema and why?
15. How do you manage concurrent updates in high-load systems?
16. How would you detect anomalies using SQL queries?
17. What's the purpose of the HAVING clause?
18. How do materialized views work and when should they be used?
19. What are common pitfalls with NULL in SQL logic?
20. Can you explain the difference between EXCEPT and NOT IN?

SQL

Nervous

1. What is a primary key in SQL?
2. What does SELECT * FROM table do?
3. How do I filter results using WHERE?
4. What's the difference between WHERE and HAVING?
5. How do I sort query results?
6. How can I limit the number of rows returned?
7. What is a JOIN and why is it useful?
8. How do I combine two tables in SQL?
9. What does GROUP BY do?
10. What is the difference between COUNT(*) and COUNT(column)?
11. How do I update a row in SQL?
12. What's the easiest way to delete data from a table?
13. What are NULL values and how do I handle them?
14. What's the purpose of aliases in SQL?

15. What is a subquery?
16. How do I check if a value exists in another table?
17. What's the simplest way to create a table?
18. How do I insert data into a table?
19. Why is my query not returning any results?
20. What are common SQL syntax mistakes?

Curious

1. Why are there so many types of JOINS?
2. What's the story behind SQL's design?
3. Why do we need normalization at all?
4. How does the database engine actually parse a query?
5. What's the exact difference between an inner join and a natural join?
6. Why is SQL considered a declarative language?
7. How does SQL handle case sensitivity in queries?
8. How does SQL manage large transaction logs?
9. What happens under the hood when I run DELETE?
10. Can you explain the internals of a B-tree index?
11. Why are relational databases still dominant?
12. How does SQL work differently in MySQL vs PostgreSQL vs MSSQL?
13. What's the difference between DDL and DML?
14. How does SQL injection work, and how is it prevented?
15. Why isn't SQL object-oriented?
16. How do foreign keys improve relational structure?
17. Why are window functions more powerful than subqueries in some cases?
18. What happens if I violate a foreign key constraint?
19. How is data stored physically in a SQL database?
20. How do databases handle auto-increment behind the scenes?

Topic : SQL

Emotional : Stressed

- ❓ My JOIN query returns duplicates—why?
- ❓ How do I fix “syntax error near ‘FROM’”?
- ❓ What do I do when a query takes forever?
- ❓ How do I stop a long-running SQL query?
- ❓ What's the fastest way to find the slow part of a query?
- ❓ How do I back up my database safely?

- ❓ Why does ORDER BY suddenly break my query?
- ❓ How do I check for missing values?
- ❓ What's the simplest way to count rows that meet a condition?
- ❓ How do I delete only the last 10 rows?
- ❓ Why isn't my GROUP BY working?
- ❓ How do I rename a column safely?
- ❓ How can I undo a DELETE statement?
- ❓ How do I prevent accidental data loss?
- ❓ How do I avoid NULL-related logic bugs?
- ❓ What's the fix for "cannot insert NULL into column"?
- ❓ My COUNT is off—what should I check?
- ❓ How do I filter by dates correctly?
- ❓ Why is my UPDATE statement affecting zero rows?
- ❓ What's the safest way to write queries during a production deployment?

Topic : Java

Emotion: Confident

1. How does the JVM manage memory and garbage collection?
2. Explain the difference between checked and unchecked exceptions.
3. What are the core principles of OOP in Java?
4. What's the difference between abstract classes and interfaces?
5. How do lambda expressions improve Java code?
6. What's the role of the volatile keyword?
7. How do you implement multithreading in Java?
8. What's the difference between HashMap and ConcurrentHashMap?
9. How does Java handle reflection?
10. What's the structure and purpose of the Java Memory Model?
11. How does Java's Stream API work under the hood?
12. What are the key changes in Java 17?
13. How do generics work with type erasure?

14. What is the role of annotations in Java frameworks?
15. Explain the lifecycle of a servlet in a Java web app.
16. What are design patterns you've used in Java?
17. How do you manage large codebases in enterprise Java?
18. What's the difference between static and dynamic binding?
19. What are best practices for writing secure Java applications?
20. How does Spring Boot simplify Java backend development?

Topic : Java

Emotion : Nervous

1. What is a class in Java?
2. How do you declare a variable?
3. What is the difference between int and Integer?
4. How do I run a Java program?
5. What is a constructor?
6. How do I create a method in Java?
7. What is an array?
8. What does public static void main mean?
9. How do I print something in Java?
10. What is the if-else statement syntax?
11. How do I use a loop in Java?
12. What does null mean?
13. What is the difference between = and ==?
14. How do I take input from the user?
15. What is a package in Java?
16. What's the easiest way to debug Java code?
17. What does try-catch do?
18. How do I create an object?
19. What's the purpose of "this" keyword?
20. How do I check if two strings are equal?

Topic : Java

Emotion : Curious

1. How is Java platform-independent?
2. What actually happens during Java compilation?
3. Why is the String class immutable?
4. How does the garbage collector decide what to remove?
5. What's the deal with method overloading vs overriding?
6. Why are objects passed by reference?
7. How does the Java Virtual Machine execute bytecode?
8. What are the different types of class loaders?
9. Why does Java have both heap and stack memory?
10. What's the story behind Java's generics?

11. What are the differences between JVM, JDK, and JRE?
12. What happens if I overload the main method?
13. What's the internal structure of a .class file?
14. Why does Java not support multiple inheritance?
15. How is Java different from C++ in memory management?
16. Why is Java slower than C but still widely used?
17. How does autoboxing work behind the scenes?
18. What's the role of the Java compiler in error detection?
19. Why is synchronization important in Java?
20. What does the final keyword do exactly?

Topic : Java

Emotion : Stressed

1. My Java program won't compile—why?
2. I'm getting a NullPointerException—what does that mean?
3. How do I fix "cannot find symbol" errors?
4. How do I properly close a Scanner?
5. What's the fastest way to debug a crash?
6. How do I fix "incompatible types" in assignments?
7. Why is my loop not terminating?
8. Why is my variable value not updating?
9. How do I safely access array elements?
10. What's the simplest way to read a file?
11. How do I catch a runtime error?
12. How do I avoid stack overflow in recursion?
13. What's the best way to avoid duplicate code?
14. How can I safely convert types?
15. How do I organize my Java files?
16. Why is Eclipse/IntelliJ giving me red lines?
17. What's the difference between == and .equals()?
18. How do I stop a program that's stuck in an infinite loop?
19. How do I import a library correctly?
20. Why does my code run but produce no output?

Topic : Machine Learning

Emotion: Confident

1. How do you tune hyperparameters in a neural network?
2. What's the difference between L1 and L2 regularization?
3. How do you avoid data leakage in model training?
4. Explain the vanishing gradient problem.
5. What's the difference between gradient descent and stochastic gradient descent?
6. How would you detect and handle multicollinearity?

7. What's the role of activation functions in deep learning?
8. How does XGBoost achieve regularization?
9. What's the difference between bagging and boosting?
10. Explain the math behind backpropagation.
11. How do you validate a time-series model?
12. What is the bias-variance tradeoff?
13. How does dropout prevent overfitting?
14. What's the intuition behind PCA?
15. How do you evaluate an imbalanced classification problem?
16. What's the difference between recall and precision?
17. How does a random forest avoid overfitting?
18. Explain ROC vs PR curves.
19. How does early stopping help model performance?
20. How do GANs work internally?

Topic : Machine Learning

Emotion: Nervous

1. What is machine learning?
2. How is ML different from traditional programming?
3. What is supervised learning?
4. What is a label in machine learning?
5. What is training data?
6. How do I evaluate if a model is good?
7. What is a model in ML?
8. What is overfitting?
9. What does accuracy mean?
10. What is a feature?
11. What is the simplest ML algorithm to start with?
12. What is the difference between classification and regression?
13. What is a test set?
14. What is a neural network?
15. What is a confusion matrix?
16. What is a decision tree?
17. What are epochs?
18. How does the model learn?
19. What is a loss function?
20. What is model training?

Topic : Machine Learning

Emotion : Confident

1. How do you tune hyperparameters in a neural network?
2. What's the difference between L1 and L2 regularization?

3. How do you avoid data leakage in model training?
4. Explain the vanishing gradient problem.
5. What's the difference between gradient descent and stochastic gradient descent?
6. How would you detect and handle multicollinearity?
7. What's the role of activation functions in deep learning?
8. How does XGBoost achieve regularization?
9. What's the difference between bagging and boosting?
10. Explain the math behind backpropagation.
11. How do you validate a time-series model?
12. What is the bias-variance tradeoff?
13. How does dropout prevent overfitting?
14. What's the intuition behind PCA?
15. How do you evaluate an imbalanced classification problem?
16. What's the difference between recall and precision?
17. How does a random forest avoid overfitting?
18. Explain ROC vs PR curves.
19. How does early stopping help model performance?
20. How do GANs work internally?

Topic : Machine Learning

Emotion : Nervous

1. What is machine learning?
2. How is ML different from traditional programming?
3. What is supervised learning?
4. What is a label in machine learning?
5. What is training data?
6. How do I evaluate if a model is good?
7. What is a model in ML?
8. What is overfitting?
9. What does accuracy mean?
10. What is a feature?
11. What is the simplest ML algorithm to start with?
12. What is the difference between classification and regression?
13. What is a test set?
14. What is a neural network?
15. What is a confusion matrix?
16. What is a decision tree?
17. What are epochs?
18. How does the model learn?
19. What is a loss function?
20. What is model training?

Topic : Learning Machine

Emotion: Curious

1. How does a machine learning model generalize?
2. Why does gradient descent work so well?
3. What happens when your data has missing values?
4. How do algorithms deal with high-dimensional data?
5. What's the intuition behind learning rate?
6. How does unsupervised learning find structure?
7. Why do models degrade over time?
8. What's the deal with bias in AI models?
9. How do word embeddings like Word2Vec work?
10. What is transfer learning and why is it powerful?
11. How does a transformer learn attention?
12. What are the philosophical limits of ML?
13. Why is deep learning better at image tasks?
14. How does reinforcement learning differ fundamentally?
15. What are the hidden layers really "learning"?
16. What's the difference between training loss and validation loss?
17. How do ensemble models work together?
18. Why does dimensionality reduction help?
19. How do we explain predictions from black-box models?
20. Can machine learning ever be 100% accurate?

Topic : Machine Learning

Emotion : Stressed

1. Why is my model not learning?
2. My accuracy is stuck—what should I check?
3. Why is my loss not decreasing?
4. How do I fix "input shape mismatch"?
5. How do I prevent overfitting in my model?
6. What's the easiest way to clean messy data?
7. My model works on train but fails on test—why?
8. What are best practices for splitting data?
9. Why is my training taking forever?
10. How do I balance an imbalanced dataset?
11. What metric should I use for classification?
12. Why is sklearn giving me shape errors?
13. What's the fastest way to preprocess numeric data?
14. How do I install TensorFlow properly?
15. Why does my GPU keep crashing?
16. What is NaN loss and how do I fix it?
17. How do I visualize my training process?
18. Why are my predictions all the same?
19. What does "data leakage" mean in ML?
20. How do I save and load a trained model?

