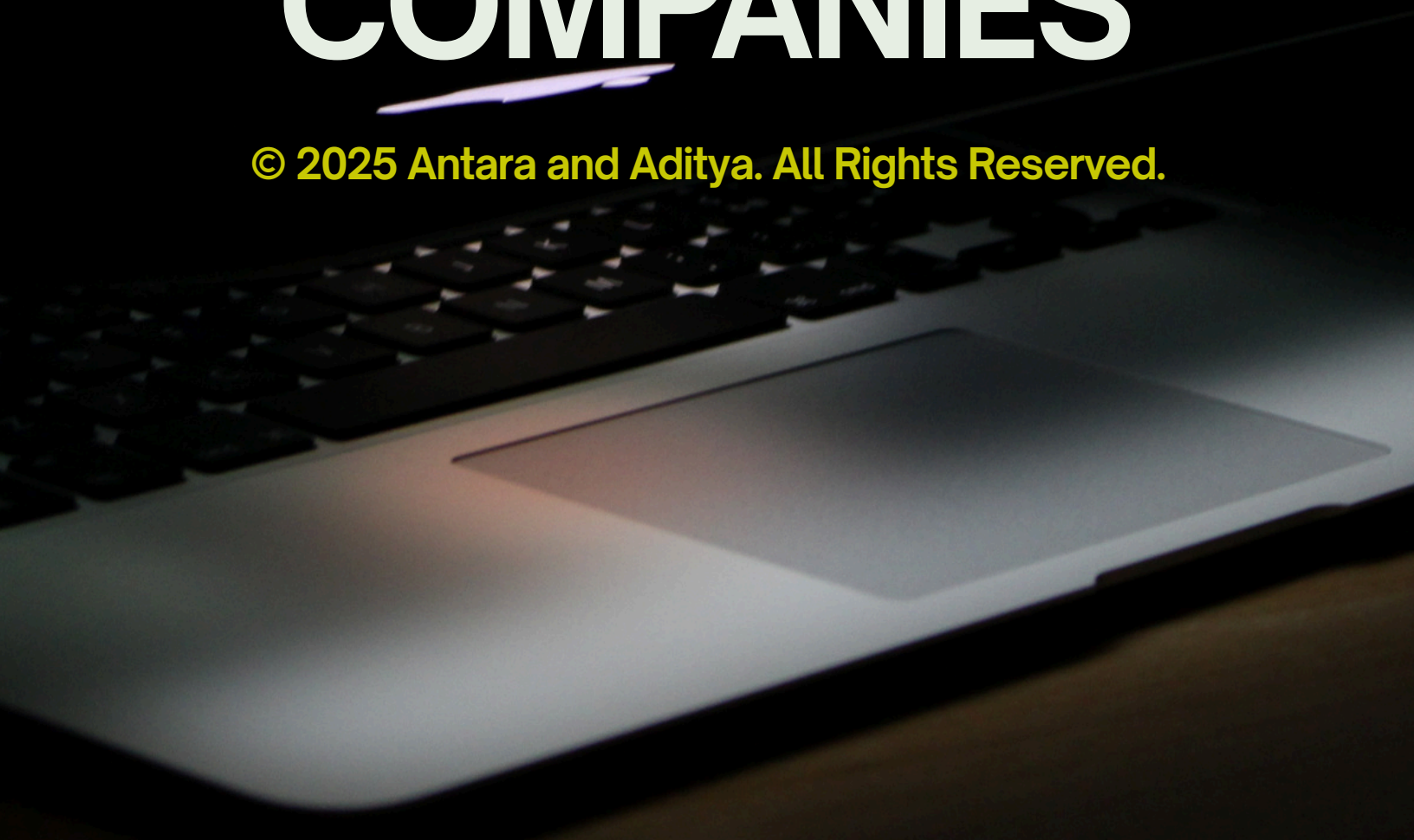
ACTUAL AI/ML INTERVIEW QUESTIONS FROM TOP COMPANIES

© 2025 Antara and Aditya. All Rights Reserved.



I. Data Preprocessing & Feature Engineering

1. How did you select features to use out of the total features? I used a combination of techniques: correlation analysis to remove highly correlated features, feature importance from tree-based models like Random Forest, and domain knowledge. I also applied statistical tests like chi-square for categorical variables and performed recursive feature elimination (RFE) to identify the most impactful features.

2. What is feature engineering? Tell me what you did in your project. Feature engineering is creating new features from existing data to improve model performance. In my project, I created interaction features, binned continuous variables into categories, extracted date components like day/month/year, and applied log transformations to skewed data to make it more normally distributed.

3. How do you handle null/missing values and which algorithm did you use to fill them? For numerical data, I used mean/median imputation or KNN imputer for better accuracy. For categorical data, I used mode imputation or created a separate 'missing' category. I also used algorithms like IterativeImputer (MICE) which predicts missing values based on other features. **4. How do you handle outliers (e.g., in a dataset of family salaries)?** I first identify
outliers using IQR method or Z-score. Then either cap them using winsorization (setting to 95th percentile), transform using log/box-cox, or remove them if they're genuine errors. For domain-specific cases like salaries, I consider if extreme values are valid before removing.

5. How do you handle imbalanced data? I use techniques like SMOTE for oversampling minority class, undersampling majority class, or combining both. I also adjust class weights in algorithms, use ensemble methods, or choose appropriate evaluation metrics like F1-score, precision-recall instead of just accuracy.

6. What is SMOTE, and can it be used on categorical data? SMOTE (Synthetic Minority Over-sampling Technique) creates synthetic samples by interpolating between minority class instances. Standard SMOTE works only on numerical data. For categorical data, we use SMOTE-NC (Nominal-Continuous) which handles mixed data types.

7. Why is scaling necessary in algorithms? Scaling ensures all features contribute equally to distance-based algorithms like KNN, SVM, and Neural Networks. Without scaling, features with larger ranges dominate. Algorithms like Decision Trees don't need scaling, but gradient descent converges faster with scaled data.

© 2025 Antara and Aditya. All Rights Reserved.

8. Which feature techniques did you use for Cat vs Cat (categorical vs categorical) variables? I used Chi-square test to measure dependency between categorical variables, Cramér's V for correlation strength, and created contingency tables for frequency analysis. This helps identify which categorical features are associated with each other.

9. What is univariate vs bivariate analysis? Univariate analysis examines one variable at a time using histograms, box plots, or summary statistics. Bivariate analysis studies relationships between two variables using scatter plots, correlation, or cross-tabulation to understand dependencies.

10. What is correlation? Correlation measures linear relationship between two variables, ranging from -1 to +1. Positive correlation means variables move together, negative means inverse relationship, and zero means no linear relationship. Pearson correlation is for numerical data, Spearman for ordinal.

11. If data is not available with the client, what will you do to get data? I would explore public datasets, web scraping if legal, use APIs, synthetic data generation, or data augmentation. I'd also suggest conducting surveys, purchasing data from vendors, or using transfer learning with similar domain datasets.

12. How can you say that an image is "ok" for the model? I check image quality through resolution, brightness/contrast levels, noise detection, and ensure proper labeling. I verify the image isn't corrupted, has sufficient features, matches the distribution of training data, and contains the object of interest clearly.

II. Machine Learning Algorithms & Concepts

13. What are the core OOPs concepts and their benefits? The four core concepts are: Encapsulation (data hiding), Inheritance (code reusability), Polymorphism (flexibility), and Abstraction (hiding complexity). Benefits include code reusability, modularity, easier maintenance, and better organization of complex systems.

14. Write a program using OOPs concepts.

```python
class DataProcessor:
    def __init__(self, data):
        self.__data = data  # Encapsulation

    def process(self):
        return f"Processing {len(self.__data)} records"
```

© 2025 Antara and Aditya. All Rights Reserved.

```python
class MLProcessor(DataProcessor):  # Inheritance
    def train_model(self):
        return "Model trained"

processor = MLProcessor([1, 2, 3])
print(processor.process())
```

15. How to protect/hide data using OOPs (Encapsulation)? Use private variables with double underscore prefix (__variable) in Python. This makes attributes inaccessible directly from outside the class. Provide getter and setter methods to control access and validate data before modification.

16. What is Shallow Copy vs Deep Copy? (With example code) Shallow copy creates a new object but references nested objects. Deep copy creates completely independent copies including nested objects.

```python
import copy
original = [[1, 2], [3, 4]]
shallow = copy.copy(original)
deep = copy.deepcopy(original)
original[0][0] = 99
# shallow[0][0] = 99, deep[0][0] = 1
```

17. Explain Method Overriding and Operator Overloading. Method Overriding: Child class redefines parent's method with same name. Operator Overloading: Defining custom behavior for operators using special methods like `__add__`, `__mul__` to make +, * work with custom classes.

18. What is the difference between a static method and a class method? Static method (@staticmethod) doesn't access class or instance, just a utility function. Class method (@classmethod) receives class as first parameter (cls), can access class variables, and is used for factory methods or alternative constructors.

19. What are the different types of data structures in computer science? Linear: Arrays, Linked Lists, Stacks, Queues. Non-linear: Trees, Graphs, Heaps. Hash-based: Hash Tables, Dictionaries. Others include Sets, Tuples, and specialized structures like Tries and Bloom Filters.

20. What is the difference between Supervised and Unsupervised Learning? Supervised learning uses labeled data to learn input-output mapping (classification,

© 2025 Antara and Aditya. All Rights Reserved.

regression). Unsupervised learning finds patterns in unlabeled data (clustering, dimensionality reduction). Supervised needs target variable, unsupervised doesn't.

21. What algorithms are used in Unsupervised Learning? K-Means Clustering, Hierarchical Clustering, DBSCAN, PCA (Principal Component Analysis), t-SNE, Autoencoders, Gaussian Mixture Models, and Apriori algorithm for association rules.

22. Explain the Decision Tree (DT) algorithm. Decision Tree splits data recursively based on features that best separate classes using metrics like Gini impurity or entropy. Each node represents a decision rule, branches represent outcomes, and leaf nodes represent final predictions. It's interpretable and handles non-linear relationships.

23. What is Random Forest? Random Forest is an ensemble of multiple decision trees trained on random subsets of data and features (bootstrap sampling). It combines predictions through voting (classification) or averaging (regression), reducing overfitting and improving accuracy over single trees.

24. On what basis is the root node split in Random Forest? The root node splits based on the feature that provides maximum information gain (reduction in entropy) or maximum decrease in Gini impurity. Random Forest randomly selects a subset of features at each split for diversity.

25. What are the hyperparameters in Random Forest? n_estimators (number of trees), max_depth (tree depth), min_samples_split, min_samples_leaf, max_features (features per split), bootstrap (sampling method), and criterion (gini/entropy).

26. If there is a column with Null values, how will a Decision Tree split on that? Decision Trees can't handle nulls directly. We must impute them beforehand using mean/median/mode or use algorithms like XGBoost that have built-in null handling by learning the best direction for missing values during splitting.

27. Explain Support Vector Machine (SVM) classifier/algorithm. SVM finds the optimal hyperplane that maximally separates classes by maximizing the margin between support vectors (closest points). It uses kernel trick for non-linear boundaries, transforming data to higher dimensions where it becomes linearly separable.

28. What is XGBoost (e.g., Explain, why prefer, implementation)? XGBoost is an optimized gradient boosting algorithm that builds trees sequentially, each correcting previous errors. Preferred for high accuracy, handles missing values, built-in regularization, parallel processing, and feature importance. It's fast and wins many competitions.

© 2025 Antara and Aditya. All Rights Reserved.

29. Explain Bagging vs Boosting. Bagging trains models independently on random subsets in parallel (Random Forest), reducing variance. Boosting trains models sequentially where each corrects previous model's errors (AdaBoost, XGBoost), reducing bias. Bagging prevents overfitting, boosting improves accuracy but can overfit.

30. What is the difference between K-Means Clustering and K-Nearest Neighbors (KNN)? K-Means is unsupervised clustering that groups similar data points into K clusters. KNN is supervised classification that predicts based on K nearest neighbors' majority vote. K-Means finds patterns, KNN classifies based on proximity.

31. How do you decide the number of clusters in K-Means? Use Elbow method (plot inertia vs K, look for bend), Silhouette score (measures cluster cohesion), Gap statistic, or domain knowledge. The optimal K is where adding more clusters doesn't significantly improve metrics.

32. What is the difference between Lasso and Ridge regression? Ridge (L2) adds squared magnitude of coefficients as penalty, shrinks coefficients but keeps all features. Lasso (L1) adds absolute value, can shrink coefficients to exactly zero, performing feature selection. Ridge for multicollinearity, Lasso for feature selection.

33. What is the difference between Regression and Classification? Regression predicts continuous numerical values (house prices, temperature). Classification predicts discrete categories or labels (spam/not spam, disease type). Regression uses MSE/RMSE metrics, classification uses accuracy/F1-score.

34. If 30k students will buy a course and 70k will not, which algorithm will you use and why? I'd use algorithms that handle imbalanced data well: XGBoost or Random Forest with class_weight='balanced', or apply SMOTE for balancing. These algorithms with proper handling prevent bias toward majority class and improve minority class detection.

35. Which basic algorithms did you try first in your project? I started with Logistic Regression as baseline for interpretability, then tried Decision Tree for non-linear patterns, Random Forest for better accuracy, and finally gradient boosting (XGBoost) for optimal performance. This progression helps understand complexity vs performance tradeoff.

36. What is Reinforcement Learning? Reinforcement Learning is learning through trial and error where an agent interacts with an environment, takes actions, receives rewards/penalties, and learns optimal policy to maximize cumulative reward. Used in game AI, robotics, and recommendation systems.

III. Deep Learning & Neural Networks

© 2025 Antara and Aditya. All Rights Reserved.

37. Explain Convolutional Neural Network (CNN) architecture and mathematical explanation. CNN has Convolutional layers (apply filters to extract features), Pooling layers (downsample), and Fully Connected layers. Convolution operation: Output = (Input * Kernel) + Bias. Each filter slides across input performing element-wise multiplication and summation, detecting patterns like edges, textures.

38. How to calculate parameters in a CNN? For Conv layer: (kernel_height × kernel_width × input_channels + 1 for bias) × number_of_filters. For FC layer: (input_neurons × output_neurons) + output_neurons (bias). Example: 3×3 kernel, 32 filters, 3 channels = (3×3×3+1)×32 = 896 parameters.

39. What is Computer Vision? Explain the process of image extraction. Computer Vision enables machines to interpret visual information. Image extraction involves: loading image, preprocessing (resize, normalize), feature extraction using CNNs (edges, shapes, objects), and using extracted features for tasks like classification, detection, or segmentation.

40. If we use ML instead of CNN for image processing, what difficulties/results will we see? ML algorithms require manual feature engineering and can't capture spatial hierarchies. We'd need to flatten images losing spatial information, struggle with high dimensionality, and get poor accuracy. CNNs automatically learn hierarchical features and spatial patterns, vastly outperforming traditional ML for images.

41. What is Backpropagation and its mathematical explanation? Backpropagation calculates gradients by applying chain rule from output to input layers. It computes loss function derivative with respect to each weight: ∂Loss/∂weight. These gradients flow backward through network, and weights are updated using gradient descent: weight_new = weight_old - learning_rate × gradient.

42. What is the role/use of an activation function in deep learning? Activation functions introduce non-linearity, enabling networks to learn complex patterns. Without them, multiple layers would collapse to single linear transformation. They determine neuron's output based on weighted input, helping network approximate any function.

43. Name different activation functions and explain their use. ReLU: f(x)=max(0,x), most common, fast computation. Sigmoid: f(x)=1/(1+e^-x), outputs 0-1, used in output for binary. Tanh: outputs -1 to 1, zero-centered. Softmax: multi-class probability distribution. Leaky ReLU: prevents dying ReLU problem.

44. Why use ReLU instead of other activation functions in the hidden layer? ReLU is computationally efficient, reduces vanishing gradient problem, provides sparse activation (some

© 2025 Antara and Aditya. All Rights Reserved.

neurons output zero), and trains faster than sigmoid/tanh. It allows better gradient flow in deep networks and has become the default choice.

45. Explain the difference between Softmax and Sigmoid functions. Sigmoid outputs independent probabilities [0,1] for each class, used in multi-label classification. Softmax outputs probability distribution summing to 1 across all classes, used in multi-class classification where classes are mutually exclusive.

46. Explain Batch Normalization. Batch Normalization normalizes inputs to each layer using batch statistics (mean, variance), reducing internal covariate shift. It allows higher learning rates, reduces dependence on initialization, acts as regularization, and speeds up training by stabilizing gradient flow.

47. Explain the structure of a Neural Network. Neural Network has Input layer (receives data), Hidden layers (extract features through weighted connections and activation functions), and Output layer (produces predictions). Each neuron performs weighted sum of inputs plus bias, then applies activation function.

48. How do we decide the number of hidden layers and neurons? Start simple with 1-2 hidden layers, gradually increase if underfitting. For neurons, often use powers of 2 (32, 64, 128) between input and output sizes. Use validation performance and cross-validation. More layers for complex patterns, but risk overfitting.

49. What is Learning Rate? Learning Rate controls step size in gradient descent for weight updates. Too high causes overshooting/divergence, too low causes slow convergence. I typically start with 0.001-0.01 and use learning rate schedulers or adaptive optimizers like Adam that adjust it automatically.

50. What are the various optimizers in deep learning? SGD (Stochastic Gradient Descent), SGD with Momentum, RMSprop, Adam (combines momentum and RMSprop), AdaGrad, AdaDelta, and Nadam. Adam is most popular for its adaptive learning rates and fast convergence.

51. What is the vanishing gradient descent problem? In deep networks with sigmoid/tanh, gradients become extremely small during backpropagation through many layers, making early layers train very slowly or not at all. Solutions include using ReLU, batch normalization, residual connections, and careful weight initialization.

52. How are weights updated in Stochastic Gradient Descent (SGD)? SGD updates weights after each sample or mini-batch: weight = weight - learning_rate × gradient. It calculates loss, computes gradients through backpropagation, and updates all weights simultaneously. This process repeats for all batches across multiple epochs.

© 2025 Antara and Aditya. All Rights Reserved.

53. How to solve the local minima problem? Use momentum-based optimizers (Adam, RMSprop), add random noise to gradients, use multiple random initializations, employ learning rate scheduling, or use batch normalization. Modern deep networks with ReLU rarely get stuck in bad local minima.

54. What is the difference between ML and DL algorithms? ML requires manual feature engineering, works well with structured data, needs less data, faster training, interpretable. DL automatically learns features, excels with unstructured data (images, text), needs large data, requires more computation, acts as black box.

55. Is image & audio data used for DL and ML also? Yes, but DL (CNNs for images, RNNs for audio) significantly outperforms traditional ML. ML requires manual feature extraction like SIFT, HOG for images or MFCC for audio. DL automatically learns features, making it the preferred choice for such data.

IV. Regularization & Model Optimization

56. How do you stop/reduce overfitting in Deep Learning? Use Dropout layers, L1/L2 regularization, data augmentation, early stopping, batch normalization, reduce model complexity, collect more training data, and use cross-validation. Dropout randomly deactivates neurons, preventing co-adaptation.

57. How to avoid overfitting in Random Forest? Limit max_depth, increase min_samples_split and min_samples_leaf, reduce n_estimators if too many, use max_features to limit features per split, apply cross-validation, and ensure sufficient training data diversity. **58.**

What is the use of dropout layers? Dropout randomly sets a fraction of neurons to zero during training, preventing neurons from co-adapting and over-relying on specific features. This creates an ensemble effect, reduces overfitting, and improves generalization. Typical dropout rate is 0.2-0.5.

59. What is Bias-Variance Tradeoff? Bias is error from wrong assumptions (underfitting), Variance is error from sensitivity to training data (overfitting). Low bias + high variance: overfitting. High bias + low variance: underfitting. Goal is balancing both for optimal generalization.

60. How will you optimize a model? Tune hyperparameters using GridSearchCV or RandomSearchCV, perform feature engineering, try different algorithms, use ensemble methods, apply regularization, increase training data, use cross-validation, and optimize based on evaluation metrics relevant to business problem.

© 2025 Antara and Aditya. All Rights Reserved.

61. How to increase parameters without changing the architecture? Increase number of filters in convolutional layers, add more neurons to dense layers, use larger kernel sizes, or increase the number of channels. However, this may lead to overfitting and requires more computational resources.

62. What is Backward Elimination? Backward Elimination is a feature selection method starting with all features, iteratively removing the least significant feature based on p-value or performance metric, and repeating until only significant features remain. It's useful for reducing dimensionality.

V. Natural Language Processing (NLP)

63. What is NLP? Natural Language Processing enables computers to understand, interpret, and generate human language. Applications include sentiment analysis, machine translation, chatbots, text summarization, and named entity recognition. It bridges communication between humans and machines.

64. What is lemmatization? Lemmatization reduces words to their base or dictionary form (lemma) considering context: "running"→"run", "better"→"good". Unlike stemming which just chops endings, lemmatization uses vocabulary and morphological analysis for accurate root words.

65. What are N-grams? N-grams are contiguous sequences of N words from text. Unigram (1 word), Bigram (2 words), Trigram (3 words). They capture word context and relationships. Example: "Machine learning rocks" → Bigrams: "Machine learning", "learning rocks".

66. Explain Word2Vec and Doc2Vec. Word2Vec creates dense vector representations of words capturing semantic meaning using CBOW or Skip-gram. Similar words have similar vectors. Doc2Vec extends this to entire documents, creating fixed-length vectors for variable-length texts, useful for document similarity.

67. What is TF-IDF and its formula? TF-IDF measures word importance in documents. TF (Term Frequency) = (count of term in doc)/(total terms in doc). IDF (Inverse Document Frequency) = log(total docs/docs containing term). TF-IDF = TF × IDF. High score means word is important but not too common.

68. What is Word Embedding and its techniques? Word Embedding represents words as dense vectors in continuous space capturing semantic relationships. Techniques include Word2Vec (CBOW, Skip-gram), GloVe, FastText, and contextual embeddings like BERT and ELMo. These enable machines to understand word meanings and relationships.

© 2025 Antara and Aditya. All Rights Reserved.

69. Have you made any chatbot? Yes, I built a rule-based chatbot using intent classification and entity extraction. I used NLTK for preprocessing, trained a classifier on labeled intents, and implemented response generation based on identified intents. For advanced versions, I'd use transformer models like BERT or GPT.

70. If you are given a PDF, tell the process of extracting data and making a summary. Extract text using PyPDF2 or pdfplumber, preprocess (remove special characters, lowercase), tokenize sentences, apply TF-IDF or TextRank to score sentence importance, select top-scoring sentences, and order them logically. For abstractive summarization, use transformer models like T5 or BART.

71. How is text data of an image converted into text for processing (OCR)? OCR (Optical Character Recognition) involves: image preprocessing (binarization, noise removal), text detection (locate text regions), character segmentation, feature extraction, character recognition using CNN or Tesseract, and post-processing (spell check, formatting). It converts visual text to machine-readable format.

VI. Advanced DL Architectures

72. What is the difference between Transformer and a Neural Network? Transformers are specialized neural networks using self-attention mechanism to process sequences in parallel, capturing long-range dependencies. Traditional NNs process data sequentially (RNNs) or spatially (CNNs). Transformers excel in NLP tasks and power models like BERT and GPT.

73. Do you know about Transformer in Deep Learning? Yes, Transformers use self-attention to weigh importance of different parts of input sequence. They have encoder-decoder architecture with multi-head attention, positional encoding, and feed-forward networks. They're highly parallelizable, handle long sequences well, and revolutionized NLP.

74. What is BERT? BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that understands context bidirectionally. It uses masked language modeling and next sentence prediction during pre-training, then fine-tuned for tasks like classification, QA, and NER.

75. What is LSTM? LSTM (Long Short-Term Memory) is a special RNN with gates (forget, input, output) that control information flow, solving vanishing gradient problem. It maintains cell state to remember long-term dependencies, making it effective for sequential data like time series and text.

© 2025 Antara and Aditya. All Rights Reserved.

76. What is the difference between LLM, RNN, and BERT? RNN processes sequences step-by-step, struggles with long dependencies. BERT is a transformer-based model pre-trained on massive text, understanding bidirectional context. LLM (Large Language Model) like GPT refers to very large transformer models trained on vast data, capable of various language tasks.

77. Explain Graph Neural Network (GNN). GNN processes graph-structured data where nodes represent entities and edges represent relationships. It aggregates information from neighboring nodes through message passing, learning node embeddings that capture both node features and graph structure. Used in social networks, molecules, and recommendation systems.

VII. Evaluation and Metrics

78. Which evaluation metrics did you use? For classification: Accuracy, Precision, Recall, F1-score, ROC-AUC, Confusion Matrix. For regression: RMSE, MAE, R-squared. For imbalanced data, I prioritize F1-score and ROC-AUC over accuracy. Choice depends on business requirements and cost of false positives vs false negatives.

79. What is the difference between Precision and Recall? Precision = TP/(TP+FP), measures accuracy of positive predictions - "of predicted positives, how many are correct?" Recall = TP/(TP+FN), measures completeness - "of actual positives, how many did we catch?" High precision: few false alarms. High recall: catch most positives.

80. How did you calculate the accuracy? Accuracy = (TP+TN)/(TP+TN+FP+FN), the ratio of correct predictions to total predictions. In code: `from sklearn.metrics import accuracy_score; accuracy = accuracy_score(y_true, y_pred)`. For multi-class, it's correctly classified samples divided by total samples.

81. What is the best model accuracy according to you? There's no universal "best" accuracy. It depends on problem complexity, data quality, and business context. For medical diagnosis, even 95% might be insufficient. For some applications, 80% is excellent. Focus on the right metric for your use case rather than chasing perfect accuracy.

82. What is a Loss Function? Loss function measures how wrong model predictions are. During training, we minimize loss to improve model. Common losses: MSE for regression, Cross-Entropy for classification, Hinge Loss for SVM. Log Loss (binary cross-entropy) penalizes confident wrong predictions heavily.

83. What are y_preds and why are they needed? y_preds are predicted values from the model. They're compared with actual values (y_true) to calculate evaluation metrics, understand model performance, identify errors, tune hyperparameters, and decide if model is ready for deployment. Essential for validation and testing.

© 2025 Antara and Aditya. All Rights Reserved.

84. What is Adjusted R-squared? Adjusted R^2 modifies R^2 by penalizing addition of irrelevant features. R^2 always increases with more features, but Adjusted R^2 only increases if new feature improves model significantly. Formula: $1 - [(1-R^2)(n-1)/(n-k-1)]$ where n=samples, k=predictors.

85. What is the ROC curve? ROC (Receiver Operating Characteristic) plots True Positive Rate (Recall) vs False Positive Rate at various threshold settings. AUC (Area Under Curve) measures overall model performance; 1.0 is perfect, 0.5 is random. Higher AUC means better discrimination between classes.

86. What is the formula for Variance? Variance = $\Sigma(xi - mean)^2 / n$ for population, or /(n-1) for sample. It measures spread of data points around mean. Standard deviation is square root of variance. In Python: np.var() for population variance, np.var(ddof=1) for sample variance.

VIII. Deployment & Cloud

87. Do you know about AWS & Azure? Yes, both are cloud platforms. AWS (Amazon Web Services) offers EC2, S3, SageMaker, Lambda. Azure (Microsoft) provides similar services. They provide scalable infrastructure for deploying ML models, storage, computing resources, and managed services for end-to-end ML pipelines.

88. What is Amazon S3? S3 (Simple Storage Service) is AWS's object storage service for storing and retrieving any amount of data. It's highly scalable, durable, and secure. Used for data lakes, backups, hosting static websites, and storing datasets for ML training.

89. What is the model deployment process/steps?

1. Train and validate model 2) Save model (pickle/joblib) 3) Create API using Flask/FastAPI 4) Containerize with Docker 5) Deploy to cloud (AWS/Azure/GCP) 6) Set up monitoring and logging 7) Implement CI/CD pipeline 8) Test endpoints 9) Monitor performance and retrain as needed.

90. Where is the model deployed? Models can be deployed on cloud platforms (AWS SageMaker, Azure ML, GCP AI Platform), on-premise servers, edge devices, or as REST APIs using Flask/FastAPI. Choice depends on latency requirements, data privacy, scale, and infrastructure availability.

91. What is AWS Sagemaker? SageMaker is AWS's fully managed service for building, training, and deploying ML models. It provides Jupyter notebooks, built-in algorithms, distributed training, hyperparameter tuning, model hosting with auto-scaling, and monitoring - covering entire ML lifecycle.

© 2025 Antara and Aditya. All Rights Reserved.

92. What is the use of Docker? Docker creates containers that package application with all dependencies, ensuring consistency across environments. It enables easy deployment, scalability, isolation, and version control. Critical for ML deployment as it handles library dependencies and ensures model runs identically everywhere.

93. Do you have experience with Docker and Kubernetes? Yes, I use Docker to containerize ML applications ensuring reproducibility. Kubernetes orchestrates these containers, handling scaling, load balancing, and automatic failover. Together they enable robust, scalable ML deployments in production with minimal downtime.

94. What are different IDEs for Deep Learning? Jupyter Notebook/Lab (interactive development), PyCharm (Python IDE), VS Code (lightweight, extensions), Google Colab (free GPU), Kaggle Notebooks, Spyder (scientific computing). For deep learning specifically, platforms with GPU support like Colab and Kaggle are popular.

95. What do you know about CUDA? CUDA is NVIDIA's parallel computing platform enabling GPUs to accelerate deep learning computations. It allows writing programs that execute on GPU cores in parallel, dramatically speeding up matrix operations crucial for neural networks. PyTorch and TensorFlow leverage CUDA automatically.

IX Programming (Python & SQL)

96. Write a function for Shallow copy and Deep copy.

```python
import copy
original = [[1, 2], [3, 4]]
shallow = copy.copy(original)
deep = copy.deepcopy(original)
original[0][0] = 99
print(f"Original: {original}, Shallow: {shallow}, Deep: {deep}")
# Original: [[99, 2], [3, 4]], Shallow: [[99, 2], [3, 4]], Deep: [[1, 2], [3, 4]]
```

97. Sort a list without using built-in functions.

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
```

© 2025 Antara and Aditya. All Rights Reserved.

```
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
print(bubble_sort([64, 34, 25, 12]))  # [12, 25, 34, 64]
```

98. Calculate product of all elements except current index.

python

```
def product_except_self(nums):
    n    =    len(nums)
    result  =  [1]  *  n
    left_product   =   1
    for i in range(n):
        result[i] = left_product
        left_product *= nums[i]
    right_product = 1
    for i in range(n-1, -1, -1):
        result[i] *= right_product
        right_product *= nums[i]
    return result
print(product_except_self([1,2,3,4]))  #[24, 12, 8, 6]
```

99. Determine maximum profit from stock trading.

python

```
def max_profit(prices):
    min_price = float('inf')
    max_profit = 0
    for price in prices:
        min_price = min(min_price, price)
        max_profit = max(max_profit, price - min_price)
    return max_profit
print(max_profit([7,1,5,3,6,4]))  # 5
```

100. Rotate an array by two positions to the right.

python

```
def rotate_right(arr, k=2):
    k = k % len(arr)
    return arr[-k:] + arr[:-k]
print(rotate_right([6, 86, 4, 86, 24, 75]))  #[24, 75, 6, 86, 4, 86]
```

© 2025 Antara and Aditya. All Rights Reserved.

101. Count occurrences of each digit in a 2 GB text file.

```python
from collections import Counter
def count_digits(filename):
digit_count = Counter()
with open(filename, 'r') as f:
        for line in f:
            digit_count.update(c for c in line if c.isdigit())
    return dict(digit_count)
```

102. Get most frequent character and its count from string.

```python
from collections import Counter
def most_frequent_char(s):
    count = Counter(s)
    char, freq = count.most_common(1)[0]
    return f"{char}{freq}"
print(most_frequent_char('aaaaabbbccdddeeeee'))  #'a5' or 'e5'
```

103. Write code for hybrid inheritance and create objects.

```python
class A:
    def method_a(self):
        print("Method from A")

class B(A):
    def method_b(self):
        print("Method from B")

class C(A):
    def method_c(self):
        print
```

Continue

```python
```

© 2025 Antara and Aditya. All Rights Reserved.

```
class C(A):
    def method_c(self):
        print("Method from C")

class D(B, C):  # Hybrid inheritance
    def method_d(self):
        print("Method from D")

obj    =    D()
obj.method_a()  #From  A
obj.method_b()  #From  B
obj.method_c()  #From  C
obj.method_d()  #From D
```

104. Get count of substring from file using regular expression.

python

```
import re
def count_substring(filename, pattern):
    with open(filename, 'r') as f:
        text = f.read()
        matches = re.findall(pattern, text)
        return len(matches)
# Usage: count_substring('file.txt', r'error')
```

105. Code for GCD of two numbers.

python

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
print(gcd(48, 18))  # 6

# Or using recursion:
def gcd_recursive(a, b):
    return a if b == 0 else gcd_recursive(b, a % b)
```

106. Program for list comprehension.

python

© 2025 Antara and Aditya. All Rights Reserved.

```python
# Square of even numbers
squares = [x**2 for x in range(10) if x % 2 == 0]
# Nested list comprehension
matrix = [[i*j for j in range(3)] for i in range(3)]
# With condition
result = [x if x > 5 else 0 for x in range(10)]
print(squares)  # [0, 4, 16, 36, 64]
```

107. What is the difference between list and tuple? List is mutable (can modify), uses square brackets [], slower, more memory. Tuple is immutable (cannot modify), uses parentheses (), faster, less memory, hashable so can be dictionary keys. Use list for changing data, tuple for fixed data.

108. What is the difference between tuple and dictionary? Tuple is ordered, immutable sequence accessed by index. Dictionary is unordered (Python 3.7+ maintains insertion order), mutable key-value pairs accessed by keys. Tuple: (1,2,3), Dictionary: {'a':1, 'b':2}. Dictionary for lookups, tuple for fixed records.

109. How to sort a DataFrame?

```python
df.sort_values(by='column_name', ascending=True)  # Sort by one column
df.sort_values(by=['col1', 'col2'], ascending=[True, False])  # Multiple columns
df.sort_index()  # Sort by index
```

110. How to fetch a record of age > 40 from a DataFrame?

```python
df[df['age'] > 40]
# Or using query method
df.query('age > 40')
# Multiple conditions
df[(df['age'] > 40) & (df['city'] == 'Mumbai')]
```

111. How to fetch frequency of records from a particular column?

```python
df['column_name'].value_counts()  # Returns frequency count
df['column_name'].value_counts(normalize=True)  # Returns proportions
```

© 2025 Antara and Aditya. All Rights Reserved.

```
df.groupby('column_name').size()  #Alternative method
```

112. What is lambda function? Lambda is an anonymous function defined in one line using lambda keyword. Syntax: `lambda arguments: expression`. Used for short functions passed as arguments. Example: `square = lambda x: x**2` or `sorted(list, key=lambda x: x[1])`.

113. What is Meta Class? Metaclass is a class of a class that defines how a class behaves. Classes are instances of metaclasses. The default metaclass is `type`. Used for class creation customization, enforcing coding standards, or implementing singleton patterns. Advanced Python concept rarely needed.

114. What is Multi-Threading? Multi-threading runs multiple threads concurrently within a process, sharing memory space. In Python, due to GIL (Global Interpreter Lock), threads don't provide true parallelism for CPU-bound tasks but are useful for I/O-bound operations like file reading, network requests. Use `threading` module.

115. What is the difference between AWS and SQL database? AWS is a cloud platform providing various services including databases. SQL database is a relational database system using SQL language. AWS offers SQL databases like RDS (MySQL, PostgreSQL) as services. AWS is infrastructure, SQL is database technology. They're not directly comparable.

116. Explain JOINs in SQL.

- **INNER JOIN**: Returns matching rows from both tables
- **LEFT JOIN**: All rows from left table + matching from right
- **RIGHT JOIN**: All rows from right table + matching from left
- **FULL JOIN**: All rows from both tables

```sql
SELECT * FROM table1 INNER JOIN table2 ON table1.id = table2.id;
```

117. What is the difference between UNIQUE and PRIMARY KEY? PRIMARY KEY uniquely identifies each row, doesn't allow NULL, only one per table. UNIQUE constraint ensures column values are unique, allows one NULL, can have multiple UNIQUE constraints. Primary key is automatically indexed and used for relationships.

118. What is a subquery in SQL? Subquery is a query nested inside another query, used in SELECT, WHERE, FROM clauses. Example:

© 2025 Antara and Aditya. All Rights Reserved.

```sql
SELECT name FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);
```

Can return single value, list, or table.

119. If table 1 has 50 records and table 2 has 40 records, final count after FULL JOIN? FULL JOIN returns all records from both tables. If there are matching records, they're combined. If no matches exist, maximum would be 50+40=90 records. Actual count depends on matching conditions - could be between max(50,40) and 90.

120. Find actor name ending with 'khan'.

```sql
SELECT actor_name FROM actors WHERE actor_name LIKE '%khan';
-- Case insensitive:
SELECT actor_name FROM actors WHERE LOWER(actor_name) LIKE '%khan';
```

121. What is the difference between WHERE and HAVING? WHERE filters rows before grouping, cannot use aggregate functions. HAVING filters groups after GROUP BY, can use aggregate functions. Example:

```sql
SELECT dept, AVG(salary) FROM emp WHERE age > 30 GROUP BY dept HAVING AVG(salary) > 50000;
```

122. Find the second highest salary.

```sql
SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees);
-- Or using LIMIT:
SELECT DISTINCT salary FROM employees ORDER BY salary DESC LIMIT 1 OFFSET 1;
-- Using DENSE_RANK:
SELECT salary FROM (SELECT salary, DENSE_RANK() OVER (ORDER BY salary DESC) as rank FROM emp
```

X. Statistics & Hypothesis Testing

123. What is hypothesis testing, and which one did you use? Hypothesis testing is a statistical method to make decisions about population parameters using sample data. I used t-

© 2025 Antara and Aditya. All Rights Reserved.

test for comparing means between two groups, chi-square test for categorical variables, and ANOVA for multiple groups. We test null hypothesis (no effect) vs alternative hypothesis.

124. Explain one-way and two-way ANOVA. One-way ANOVA tests if means of three or more groups differ based on one independent variable (e.g., effect of diet type on weight loss). Two-way ANOVA tests effects of two independent variables and their interaction (e.g., diet type AND exercise on weight loss).

125. What is the difference between population and sample? Population is entire group we want to study (all customers). Sample is subset of population used for analysis (1000 customers surveyed). Sample should be representative of population. We use sample statistics to infer population parameters due to cost and time constraints.

126. How can we say that samples are appropriate representative of the population? Use random sampling methods (simple random, stratified, systematic), ensure adequate sample size using power analysis, check sample demographics match population, verify no sampling bias, and validate that sample distribution resembles population distribution. Statistical tests like Kolmogorov-Smirnov can help.

127. What is probability of getting maximum two heads when three coins are tossed? Sample space: {HHH, HHT, HTH, THH, HTT, THT, TTH, TTT} = 8 outcomes. Maximum 2 heads means 0, 1, or 2 heads (exclude HHH). Favorable outcomes: 7. Probability = 7/8 = 0.875 or 87.5%.

XI. Miscellaneous Concepts

128. What is windows shifting? Window shifting (sliding window) is a technique in time series and sequence data where a fixed-size window moves across data to create samples. Used in forecasting (past N values predict next value), CNNs (convolution operation), and streaming data analysis for real-time processing.

129. What is annotation? How to choose annotation tools? Annotation is labeling data for supervised learning (bounding boxes for object detection, labels for classification). Choose tools based on: data type (LabelImg for images, Doccano for text), budget (LabelStudio is free), team size, integration capabilities, and annotation complexity requirements.

130. How to secure your scripts? Use environment variables for credentials, never hardcode passwords, implement input validation, use try-except for error handling, apply encryption for sensitive data, use version control (.gitignore for secrets), implement access controls, use virtual environments, keep dependencies updated, and use secrets management tools like AWS Secrets Manager.

© 2025 Antara and Aditya. All Rights Reserved.