# BarRaiser Interview Experience

## Coding (ML)

1. Given an array of integers, you need to return the product of given array elements except including the current element itself.

2. Design an algorithm to schedule meetings for a company, maximizing the number of non-overlapping meetings. Given a list of meetings with start and end times, output the maximum number of meetings that can be scheduled without any conflicts.
   Sample Input:
   Meetings: [(1, 3), (2, 4), (4, 6), (5, 7)]

   Sample Output:
   Maximum number of meetings: 2
   Explanation: The optimal schedule would be to select meetings (1, 3) and (5, 7) without any conflicts.

3. Design an algorithm to find the optimal itinerary for a trip, visiting multiple cities exactly once and returning to the starting city. Given a list of flights between different cities with departure and arrival times, output the itinerary that minimizes the total travel time.
   Sample Input:
   Flights: [(A, B, 8 AM, 9 AM), (B, C, 10 AM, 12 PM), (C, A, 1 PM, 3 PM)]

   Sample Output:
   Optimal itinerary: A -> B -> C -> A
   Explanation: The optimal itinerary starts at city A, then moves to city B, followed by city C, and finally returns to city A.

4. Implement a data structure and algorithms to efficiently handle product listings and user transactions in an online marketplace. Given a list of products with their prices, implement a function to return the highest-priced product and another function to add a new product to the marketplace.
   Sample Input:
   Products: [(Product A, $10), (Product B, $20), (Product C, $15)]

   Sample Output:
   Highest-priced product: Product B
   Explanation: Product B has the highest price of $20.

5. Optimize the bus routes for a city's public transportation system. Given a set of bus stops and their connections, design an algorithm to find the shortest path between any two bus stops.
   Sample Input:

Bus Stops: A, B, C, D
Connections: [(A, B), (B, C), (C, D), (A, D)]

Sample Output:
Shortest path from A to D: A -> D
Explanation: The shortest path from bus stop A to D is directly from A to D.

6. Design an algorithm to suggest the top restaurants in an area based on a user's location and preferences. Given a list of restaurants with their cuisines and ratings, output the top-rated restaurants that match the user's preferences.
   Sample Input:
   User's Location: City X
   User's Cuisine Preference: Italian
   Restaurants: [(Restaurant A, Italian, 4.5), (Restaurant B, Mexican, 3.8), (Restaurant C, Italian, 4.2)]

   Sample Output:
   Top-rated Italian restaurants in City X: Restaurant A, Restaurant C
   Explanation: The algorithm suggests Restaurant A and Restaurant C as the top-rated Italian restaurants in City X.

7. You are given an m x n grid where each cell can have one of three values:
   a. representing an empty cell,
   b. representing a non virus infected system, or 2 representing a virus infected system.

   Every minute, any non virus system that is connected to any 4-directionally virus infected system becomes virus infected system.
   Return the minimum number of minutes that must elapse so that all system become virus infected.
   Input: grid =
   [
   [2,1,1],
   [1,1,0],
   [0,1,1]
   ]
   Output: 4
   Discuss your approach and write the code.

8. Given n non-negative integers a1, a2, ..., an, where each represents a point at coordinate (i, ai). n vertical lines are drawn such that the two endpoints of line i are (i, ai) and (i, 0). Find two lines, which, together with the x-axis form a container, such that the container contains the most water.
   Examples:
   Input: height = [1,8,6,2,5,4,8,3,7] Output: 49
   Input: height = [1,1] Output: 1

9. You are given an integer array nums. You are initially positioned at the array's first index, and each element in the array represents your maximum jump length at that position.
   Return true if you can reach the last index, or false otherwise.
   Example 1:
   Input: nums = [2,3,1,1,4]
   Output: true
   Explanation: Jump 1 step from index 0 to 1, then 3 steps to the last index.
   Example 2:
   Input: nums = [3,2,1,0,4]
   Output: false
   Explanation: You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.

10. **Autoregressive Token Generation**

Problem:
https://colab.research.google.com/drive/1DEZaH6gFw9b70pweLr_yCENbc9dFZHC?usp=sharing" rel="noopener noreferrer" target="_blank">Collab Notebook

We have a model that predicts the next token based on the previous N. Implement a generate method that takes a sequence of tokens and predicts the next N tokens. It should do this auto regressively using the top-k =1.

Solution:
https://colab.research.google.com/drive/1EGM1W4pJ4JCV2Sbs05Ov0DBO6GzvFLfm?usp=sharing"
rel="noopener noreferrer" target="_blank">Collab NoteBook

11. **Custom Loss Implementation**

Problem: https://colab.research.google.com/drive/1CzUJ4KVmbSkRHCs-9lWOW1_ieSeivQpj?usp=sharing" rel="noopener noreferrer" target="_blank">Collab NoteBook

You are working on a computer vision task where you need to implement a custom loss function that combines the Mean Squared Error (MSE) loss with a structural similarity (SSIM) index. The MSE should have a weight of 0.8, and the SSIM should have a weight of 0.2. The SSIM index is calculated using the torchvision library. Implement the custom loss function and demonstrate how it would be used in a training loop with a simple convolutional neural network.

Solution Link:
https://colab.research.google.com/drive/1JbEpU2QV7GzXchcNVjwI3DZ0dlqmnry?usp=sharing

12. **Image transformation using OpenCV**

Problem:
https://colab.research.google.com/drive/19Fb3SYxsJl6IhlFwtsHO59t2cFckG0t?usp=sharing" rel="noopener noreferrer" target="_blank">Collab Notebook:

Given a kernel, develop a function that applies the kernel to an image. It should be used without padding and with a stride of 1. If input image size is N * M, the output image size should be (N - 2) * (M - 2) with a kernel size 3 * 3

Solution Link:
https://colab.research.google.com/drive/1FTSoCDvAxWsO7xREotZyQjzGlB9E0c5Y?usp=sharin g

## DS Python Programming

Dataset: Airbnb dataset.ipynb - Colab (google.com)

Solution link: Solutions - Google Docs

Q1: What is the neighborhood in which superhosts have the biggest median price difference with respect to non superhosts? Use the following three columns in the 'listings' dataset to answer this question: 'host_is_superhost', 'neighbourhood_cleansed', and 'price'.

Q2: Which of the review scores has the highest correlation to price? Use the following review score columns in the 'listings' dataset: 'review_scores_rating', 'review_scores_accuracy', 'review_scores_cleanliness', 'review_scores_checkin', 'review_scores_communication', 'review_scores_location', 'review_scores_value'.

Q3: What is the average price difference between a professional host and a nonprofessional one? Consider a host as professional if they have listings in more than 5 different locations (location is defined by the 'neighbourhood_cleansed' column).

Q4: What is the median price premium given to entire homes / entire apartments with respect to other listings of the same neighborhood? Report the average across all neighborhoods. Use the 'room_type' column in the 'listings' dataset to distinguish between entire homes / entire apartments and other types of listings

Q5: What is the listing with the best expected revenue based on the last 12 months, considering 60% of guests leave reviews and every guest will stay only the minimum number of nights? Use both the 'listings' and 'reviews' datasets for this question and only use listings with minimum nights of stay <= 7. The 'minimum_nights' column indicates the required minimum number of nights of stay for any listing.

Q6: What is the average difference between review scores of superhosts vs normal hosts? Use the 'review_scores_rating' column for determining the average review scores

Q7: Which host attribute has the second-highest correlation with the number of reviews of the listing? Use the following columns as the host attributes: 'host_since', 'host_listings_count', 'host_identity_verified', 'calculated_host_listings_count', 'host_is_superhost', Use 'number_of_reviews' as the column to find correlation with

## ML Questions

**EASY QUESTION (At least3)**

Why is data cleaning necessary before analysis? ( DS/DA - Easy) What does the term "epoch" mean in machine learning?
Explain the difference between precision and recall. In which scenarios would you prioritize one over the other?
How would you split a dataset into training and testing sets using Scikit-learn? What is the purpose of regularization in machine learning?
What is the difference between supervised and unsupervised learning? Can you provide an example of each?
How do you handle missing values in a dataset?
When performing k means clustering how do you choose K
What is overfitting in machine learning, and how can it be prevented?

**Moderate Level (At least 2):**

How is K-nearest neighbors (KNN) different from K-means clustering? ( DS/DA - Medium)
What are the advantages of using ensemble techniques? Provide examples. ( DS Medium)
What do you understand by Confusion Matrix? ( DS/DA - Medium)
What is an F1 score and when would you use it? ( DS/DA - Medium)
How might you avoid overfitting in a model? ( DS/DA - Medium)
What is the difference between bagging and boosting? ( DS/DA - Medium)
What is Maximum Likelihood Estimation (MLE)? ( DS/DA - Medium ) What is PCA?
What is backpropagation? Describe a backpropagation workflow in either Pytorch or Tensorflow?

**Hard Level:**

What is the primary difference between L1 and L2 regularization?

## Machine Learning:

How do you choose the appropriate machine learning algorithm for a given problem? What factors influence your decision?

How do you handle imbalanced datasets? What techniques do you employ to ensure fair model performance?

How do you evaluate model performance for different types of machine learning tasks (e.g., classification vs. regression)?

How do you approach feature selection and engineering in your machine learning projects? Can you describe a specific technique you have used?

How do you ensure that your model generalizes well to unseen data? What methods do you implement to avoid overfitting?

How do you use cross-validation in your machine learning workflow? Can you explain its benefits?

How do you handle categorical variables in your datasets? What encoding techniques do you find most effective?

How can you handle imbalanced datasets in anomaly detection?

How can you handle multicollinearity or heteroscedasticity in linear regression?

How can you handle missing or categorical data when applying K-means clustering?

How can you handle imbalanced class distributions in Naive Bayes?

How can you handle outliers or skewed variables before applying PCA?

What checks can you perform to assess the goodness-of-fit and predictive performance of a logistic regression model?

What real-world applications can benefit from anomaly detection techniques?

## Deep Learning:

How do you decide between using a CNN and an RNN for specific tasks? What are the key factors in your decision-making process?

How do you optimize hyperparameters in your deep learning models? What methods or tools do you prefer for this task?

How do you address overfitting in your deep learning models? What regularization techniques do you commonly apply?

How do you manage and preprocess large datasets for deep learning projects? What tools do you find effective?

How do you evaluate the performance of your deep learning models? What metrics do you use, and why?

How do you implement transfer learning in your deep learning models? Can you share a specific example?

How do you handle different types of input data (e.g., images, text) in a multi-modal deep learning model?

## Statistics

How would you segment customers based on their purchasing behavior using demographic and purchase history data? What techniques and evaluation methods would you employ?

How would you design and analyze an A/B test to compare user engagement between different UI layouts? Which statistical methods would you use, and how would you interpret the results?

How would you build a predictive model for estimating trip duration in a ride-sharing company? Which statistical algorithms would you consider, and how would you evaluate model performance?

How would you design an anomaly detection system for identifying fraudulent transactions in credit card data? What statistical features or patterns would you look for, and how would you set the classification threshold?

How would you analyze a dataset of daily stock prices to identify trends, seasonality, and predict future prices? Which statistical techniques or models would you use, and how would you interpret the results?

## ○ AI Model Development:

- How do you approach the process of selecting an appropriate model for a given dataset? What factors do you consider when making your choice?
- How do you handle missing data when preparing your dataset for model training? Can you describe the techniques you use to address this issue?
- How do you ensure that your model generalizes well to unseen data? What strategies do you implement to prevent overfitting during training?
- How do you evaluate the performance of your AI model? What metrics do you typically use for classification and regression tasks?
- How do you implement feature engineering in your projects? Can you provide an example of a feature transformation that improved your model's performance?
- How do you manage the model training process, particularly when it comes to tuning hyperparameters? What tools or techniques do you find most effective?
- How would you deploy a trained AI/ML model into a production environment? What steps would you take to ensure the deployment is smooth and successful?

  **Follow-up**: What are some challenges you might face when deploying a model, and how would you mitigate them?

- How would you serve an AI model as an API endpoint to allow external applications to access and use it in real-time?

  **Follow-up:** What tools or frameworks would you use to make the model available through RESTful APIs (e.g., Flask, FastAPI, TensorFlow Serving)?

- How would you monitor the performance of a deployed AI model over time? What metrics would you track, and why are they important?
  **Follow-up**: How would you detect and handle model drift or degradation in model accuracy once deployed?

- How do you ensure that your deployed AI model can scale effectively when there's an increase in traffic or requests?

  **Follow-up**: Can you explain how you would use technologies like Kubernetes or AWS Sagemaker to scale AI model deployments?

- How would you implement a CI/CD pipeline for AI/ML models to ensure the continuous delivery of updates and improvements?

**Follow-up**: How would you automate the retraining of models in a CI/CD pipeline when new data becomes available?

- How do you optimize an AI model for deployment to ensure that it has minimal latency and performs well in real-time applications?

  **Follow-up**: What techniques would you use for model compression, quantization, or hardware optimization?

- How would you manage and version multiple iterations of a deployed AI model in production?

  **Follow-up**: How would you decide when to switch from one version of the model to another in a production environment?

- How do you ensure that an AI model deployment is secure? What are the security concerns that need to be addressed?

  **Follow-up**: How would you protect sensitive data or personally identifiable information (PII) that might be part of the model's input/output?

## ○ AWS:

How do you start a machine learning project using Amazon SageMaker? Can you outline the basic steps involved in creating a model?

How do you use built-in algorithms in Amazon SageMaker? Can you give an example of an algorithm you've used and the type of problem it solved?

How do you create a simple AWS Lambda function? What are the main components you need to set up?

How do you trigger an AWS Lambda function in response to an event? Can you provide an example of a common event source?

How do you interpret the results from AWS Comprehend after analyzing text? What insights can you derive from the output?

How do you handle different languages when using AWS Comprehend? What options does the service provide for multilingual text analysis?

## ML Frameworks:

How do you choose between TensorFlow and PyTorch for a specific machine learning project? What factors do you consider when making this decision?

How do you implement data preprocessing using a framework like TensorFlow or PyTorch? What tools or libraries do you typically use for this purpose?

How do you optimize model training in TensorFlow or PyTorch to reduce training time while maintaining accuracy? What techniques do you apply?

How do you handle overfitting in your machine learning models when using TensorFlow or PyTorch? What strategies do you implement to mitigate this issue?

How do you utilize GPU acceleration when training models in TensorFlow or PyTorch? What steps do you take to ensure efficient utilization of available resources?

How do you implement model evaluation and validation using machine learning frameworks? What metrics do you consider important for assessing model performance?

## ○ API Understanding:

- How would you retrieve data from a REST API endpoint and integrate it into your data analysis or machine learning pipeline?

  **Follow-up**: What libraries or tools in Python (or other languages) would you use to make API requests and handle responses?

- How do you handle API authentication when making requests to a service that requires secure access (e.g., OAuth, API keys)?

  **Follow-up:** How would you securely manage and store API credentials to ensure they are not exposed?

- How would you handle a scenario where an API returns a large dataset? What strategies would you use to efficiently manage the response (e.g., pagination, rate-limiting)?

  **Follow-up**: What challenges have you encountered with rate limits when working with APIs, and how did you overcome them?

- How would you integrate an external API (e.g., a weather API or a financial data API) into your machine learning model to use as a feature?

  **Follow-up**: How would you handle updating the model as new data becomes available through the API at regular intervals?

- How do you handle errors when an API request fails, such as when the service is down, or the API returns an error code (e.g., 404 or 500)?

  **Follow-up**: How would you implement retry logic or fallback mechanisms to make the integration more robust?

- How would you expose a machine learning model you built through an API, allowing other services to interact with it and get predictions?
  **Follow-up**: What tools (e.g., Flask, FastAPI, Django) would you use to deploy your model as an API, and how would you handle scalability?

- How do you handle rate-limiting restrictions when working with APIs, especially in cases where you need to make frequent requests?

  **Follow-up**: How would you manage multiple API calls in a batch to avoid hitting rate limits, while ensuring efficiency?

- How would you parse and handle JSON responses from an API for data processing in Python?

  **Follow-up:** What techniques do you use to validate and clean API data before using it in your analysis or modeling workflow?

## **Math Questions:**

Question Link: https://t.ly/BE4qI