Q1: Data Provenance

Q: Imagine you're joining a new team and you find that the dataset you need to work on lacks clear records of how it was collected or processed. How would you ensure the data provenance is well understood before building your model?

问：假设你加入了一个新的团队，而你发现需要处理的这个数据集缺乏关于其收集或处理过程的清晰记录。那么在构建模型之前，你将如何确保数据来源能够被充分理解呢?

A: I would first audit the data for artifacts, biases, and distributional shifts. Then, I'd analyze its statistical properties and metadata to infer the collection and processing steps, documenting all findings to establish a reproducible baseline.

回答: 我会首先审计数据中的伪影、偏差和分布变化。然后，我会分析其统计特性和元数据来推断收集与处理步骤，并记录所有发现，以建立一个可复现的基线。

Interview回答

If I join a new team and the dataset lacks clear documentation, I'd start by doing a quick "data forensics" check — looking for any signs of bias, missing values, or strange patterns that might hint at how the data was collected or cleaned. Then, I'd explore its metadata and statistical distributions to infer things like sampling frequency or preprocessing steps. I'd also talk to the team members or previous data owners to fill in the gaps. Throughout this process, I'd carefully document every assumption and observation so we can rebuild a clear data provenance trail and ensure the analysis is reproducible later.

如果我加入一个新团队，而这个数据集没有清晰的来源或处理记录，我会先做一轮"数据取证"检查——看看有没有明显的偏差、缺失值或异常模式，这些往往能反映出数据是怎么被收集或清洗的。接着，我会分析它的元数据和统计分布，比如采样频率、字段命名方式、数值范围等，以推断可能的处理流程。同时，我也会主动和团队成员或之前的数据负责人沟通，补齐背景信息。整个过程中，我会把所有假设、推断和发现都系统地记录下来，确保后续的分析有可追溯性和可复现性。

Q2: Data Quality (GIGO)

Q: Can you describe a dataset you have worked with - or discussed in class - that has poor-quality data, and explain the steps you took to improve it before running any analysis?

问：能否描述一个您曾处理过（或在课堂上讨论过）的、数据质量不佳的案例，并说明在进行任何分析之前，您采取了哪些步骤来改善数据状况?

A: The Google Flu Trends dataset is a classic "Garbage In, Garbage Out" (GIGO) example. It failed due to data drift. To fix it, I'd use data cleaning to remove noisy features and implement monitoring (like the Population Stability Index) to detect and correct for these distributional shifts.

回答: 谷歌流感趋势 (Google Flu Trends) 是一个典型的"GIGO"案例，它因数据漂移而失败。要修复它，我会用数据清洗移除噪声特征，并实施监控（如PSI）来检测和纠正这些分布变化。

Interview回答

One example I often mention is the Google Flu Trends project — it's a great lesson in what happens when data quality goes unchecked. The model started off doing well, but over time, the relationship between people's search behavior and real flu cases drifted. In other words, the data stopped representing reality.

If I were working on a dataset like that, I'd first audit it for outliers, inconsistent features, or proxy variables that might no longer hold. Then, I'd clean or reweight the data to reduce noise and bias. Most importantly, I'd set up a data monitoring system — for instance, using metrics like the Population Stability Index (PSI) — to track whether the data distribution shifts over time. That way, the model can be retrained or recalibrated before things get out of control.

我经常举的一个例子是"谷歌流感趋势"项目，这是一个典型的数据质量问题。这个模型一开始预测得很好，但后来搜索行为和真实流感病例之间的关系发生了漂移，也就是说，数据不再代表现实了。

如果我遇到这样的数据集，我会先做一轮数据审查，比如检查异常值、不一致的特征，或者那些已经失效的代理变量。然后我会通过数据清洗或重新加权的方式，去减少噪声和偏差。最关键的是，我会建立一个数据监控机制，比如用人口稳定指数（PSI）这样的指标，去跟踪数据分布是否随时间发生变化。这样一来，当数据开始"走样"时，模型就能及时地被重新训练或校正。

Q3: Correlation vs. Causation

Q: Suppose you're analyzing a dataset that shows a strong correlation between two variables. How would you explain to a non-technical colleague why that correlation doesn't necessarily mean one causes the other?

问：假设你正在分析一个数据集，该数据集显示了两个变量之间存在很强的相关性。你会如何向一个不懂技术的同事解释为什么这种相关性并不一定意味着一个变量会导致另一个变量的变化?

A: I'd use an analogy. Ice cream sales correlate with drowning, but ice cream doesn't *cause* drowning. The hot weather (a confounder) causes both. Correlation just shows they move together; it doesn't prove one *causes* the other.

回答: 我会用一个类比。冰淇淋销量与溺水相关，但冰淇淋不*导致*溺水。炎热天气（一个混杂因素）同时导致了两者。相关性只显示它们一起变动，并不证明一个是另一个的*原因*。

Interview回答

If I were explaining this to a non-technical colleague, I'd start with a simple example. For instance, ice cream sales and drowning cases often rise together — but that doesn't mean ice cream *causes* drowning. In reality, both go up when it's hot, so the real driver is the temperature — what we call a *confounding factor*.

So correlation just tells us that two things move together, but not *why* they do. To figure out causation, we'd need more evidence — for example, experiments, time-series analysis, or controlling for other variables to rule out alternative explanations.

如果我要跟一个不懂数据分析的同事解释，我会先举个简单的例子：冰淇淋销量和溺水人数通常会一起上升，但这并不代表吃冰淇淋会导致溺水。真正的原因是天气变热——这是一个我们叫"混杂因素"的东西，它同时影响了两者。

所以，相关性只能告诉我们两个变量"同时变化"，但并不能说明"谁导致了谁"。要验证因果关系，还需要更多证据，比如做实验、分析时间序列，或者控制其他可能的影响因素，才能真正判断出因果。

Q4: Dimensionality Reduction

Q: Describe a project where it is important to reduce the dimensionality of your data. What methods did you use, and how did it help you solve the problem?

问：请描述一个需要降低数据维度的项目。您采用了哪些方法？这些方法是如何帮助您解决该问题的？

A: For hyperspectral images, which can have over 200 bands, I'd use Principal Component Analysis (PCA). This method compresses the highly correlated bands into just a few principal components that capture most of the variance, which simplifies visualization and makes classification models more efficient.

回答: 对于高光谱图像，它可能有200多个波段，我会使用主成分分析 (PCA)。这种方法将高度相关的波段压缩为少数几个主成分，捕获大部分方差，这简化了可视化，也使得分类模型更高效。

Interview回答

In one of my projects, I worked with hyperspectral images — those are images with hundreds of wavelength bands, sometimes more than 200. The problem was that each pixel had such high-dimensional information that it became really hard to visualize or run any classification efficiently.

To deal with that, I used Principal Component Analysis, or PCA. It basically compresses all those highly correlated spectral bands into just a few components that still capture most of the variance — say, the first 5 or 10 can already explain over 95% of the information. That reduction not only made the data much easier to visualize, but also helped my classification models train faster and avoid overfitting. So dimensionality reduction really turned the data from “too complex to handle” into something both interpretable and efficient to use.

在我做的一个项目中，我们处理的是高光谱图像——这类图像在不同波长上可能有上百个波段，有时候超过200个。问题在于，每个像素都带着这么多维度的信息，既难以可视化，也让分类模型运行得非常慢。

为了解决这个问题，我用了主成分分析（PCA）。它能把这些高度相关的波段压缩成少数几个主成分，同时仍然保留绝大部分的方差信息。比如前几个主成分通常就能解释95%以上的变化。这样不仅让数据更容易可视化，也让模型训练更高效，减少了过拟合风险。换句话说，降维让原本“复杂到难以下手”的数据变得既直观又可用。

Q5: The Manifold Hypothesis

Q: Imagine you're working with a complex dataset and suspect the data lies on a lower-dimensional manifold. How would you uncover that structure, and why would that be useful?

问：假设你正在处理一个复杂的数据集，并且怀疑这些数据存在于一个低维的曲面上。那么，你会如何揭示这种结构呢？这又有何实际用途呢？

A: I’d use methods like Diffusion Maps or a VAE to find that low-dimensional structure. This is useful because it allows for denoising by projecting data onto the manifold and using geodesic (on-manifold) distances, which are more meaningful than raw Euclidean distance in the high-D space.

回答: 我会使用像扩散图 (Diffusion Maps) 或变分自编码器 (VAE) 这样的方法来找到低维结构。这很有用，因为它允许通过将数据投影到流形上来去噪，并使用（流形上的）测地距离，这比高维空间中的原始欧氏距离更有意义。

Interview回答

When I suspect that a complex dataset actually lives on a lower-dimensional manifold, I’d try to uncover that hidden structure using techniques like Diffusion Maps or a Variational Autoencoder.

In simple terms, these methods help reveal the “true shape” of the data — for example, even if the data lives in a 100-dimensional space, it might really vary along just a few meaningful directions. Once we map it to that low-dimensional space, we can denoise it, visualize it more clearly, and measure similarity using *on-manifold* distances that reflect the data’s real geometry, rather than noisy Euclidean distances.

This helps a lot in tasks like clustering or anomaly detection, because it lets the model focus on the intrinsic structure of the data instead of being distracted by high-dimensional noise.

当我怀疑一个复杂数据集其实“藏在”一个更低维的流形上时，我会用一些方法来揭示它的真实结构，比如扩散图（Diffusion Maps）或变分自编码器（VAE）。

简单来说，这些方法能帮助我们找到数据的“真正形状”——虽然它表面上是高维的，比如有上百个特征，但实际上只沿着少数几个有意义的方向变化。把它映射到那个低维空间后，我们不仅能更清楚地可视化数据，还能进行去噪，并用流形上的测地距离来度量相似性，这种距离比原始欧氏距离更符合数据的真实几何关系。

这样做对聚类或异常检测很有帮助，因为模型能更专注于数据的内在结构，而不是被高维噪声干扰。

Q6: Supervised vs. Unsupervised

Q: Can you share an example where you had to choose between a supervised and an unsupervised learning approach in a scientific problem. What factors guided your decision?

问：您能否分享一个实例，说明在解决某个科学问题时，您需要在有监督学习和无监督学习这两种方法中做出选择。当时您是依据哪些因素来做出这一决定的？

A: The choice depends on having labels. To classify known apple leaf diseases from our dataset, we have labels like 'rust' or 'scab', so we use **supervised** learning. But to discover *new*, previously unknown cell types from raw RNA-seq data where no labels exist, we must use **unsupervised** clustering.

回答: 这个选择取决于是否有标签。要分类我们数据集中已知的苹果叶疾病，我们有像“锈病”或“黑星病”这样的标签，所以我们使用**监督**学习。但要从没有标签的原始RNA序列数据中发现*新的*、未知的细胞类型，我们就必须使用**非监督**聚类。

Interview回答

In one of my projects, I had to decide between using supervised and unsupervised learning. The key factor was whether we had reliable labels and what our goal was — prediction or discovery.

For example, when classifying known apple leaf diseases, we already had labeled images like *rust*, *scab*, and *blight*. So supervised learning made sense — we could train a CNN to recognize those categories and evaluate its accuracy directly.

But in another case, when working with raw single-cell RNA-seq data, there were **no labels** at all. The goal there wasn't to predict, but to *discover* — to find new or previously unknown cell types. In that situation, I used unsupervised clustering methods like PCA and t-SNE followed by DBSCAN to uncover natural groupings.

So my decision always depends on two things: do we have trustworthy labels, and are we trying to predict known outcomes or explore hidden structure in the data.

在我的一个项目中，我需要在监督学习和非监督学习之间做选择。决定性因素其实是两个：**有没有可靠的标签**，以及**目标是预测还是探索**。

比如，在分类苹果叶片疾病时，我们已经有了标注好的样本，如“锈病”“黑星病”“枯斑病”等，这时监督学习最合适——我们可以训练一个CNN模型来识别并评估准确率。

但在另一个案例中，当我处理单细胞RNA测序数据时，完全没有标签，目标不是预测，而是发现新的细胞类型。这种情况下，我用了非监督方法，比如先做PCA和t-SNE降维，再用DBSCAN聚类，从数据本身找出自然分组。

所以我一般的判断逻辑是：**有标签就预测（supervised），没标签就探索（unsupervised）。**

Q7: Vector Quantization

Q: Suppose you're trying to cluster a large set of experimental results to identify common patterns. How might vector quantization help, and what would you look for in the resulting clusters?

问：假设您正试图对大量实验结果进行聚类，以找出其中的共性模式。那么向量量化能起到什么作用呢？在得到的聚类结果中，您又会关注哪些方面呢?

A: Vector Quantization (VQ) compresses complex, continuous results into a small "codebook" of discrete, representative patterns. This reduces noise. I'd look for clusters that represent stable, recurring experimental outcomes, like the distinct "air-quality regimes" we identified in the India dataset.

回答: 矢量量化 (VQ) 将复杂的连续结果压缩成一个小的、离散的、有代表性的"码本"。这能减少噪声。我会寻找那些代表稳定的、可复现的实验结果的簇，就像我们在印度数据集中识别出的不同"空气质量状态"。

Interview回答

If I have a large set of experimental results and I want to cluster them to find recurring patterns, vector quantization can be really helpful.

In simple terms, it compresses high-dimensional, continuous results into a small "codebook" of representative prototypes. So instead of dealing with thousands of slightly different outcomes, we can represent them with just a few typical patterns — that not only reduces noise but also makes the structure of the data much clearer.

After applying VQ, I'd look at the clusters to see if they correspond to **stable and physically meaningful regimes** — for example, in one project with air-quality data from India, the clusters revealed distinct "air-quality states" that repeated across time and regions. That kind of pattern tells us the quantization captured real, consistent behavior rather than random fluctuations.

如果我面对大量实验结果，想要从中聚类出一些共性的模式，向量量化（VQ）会非常有用。

简单来说，它可以把高维、连续的结果压缩成一个较小的"码本"，每个码代表一种具有代表性的典型模式。这样我们不再需要面对上千个略有差异的实验输出，而是能用几个"代表性原型"去概括它们。这不仅能减少噪声，也让数据结构变得更清晰。

在得到聚类结果后，我会去看这些簇是否反映了**稳定且具有物理意义的状态**。比如在我分析印度空气质量数据的项目中，VQ 聚类结果揭示了不同的"空气质量状态"，这些状态会在不同时间和地区重复出现——这说明模型捕捉到的是真实的规律，而不是随机波动。

Q8: Interpolation vs. Extrapolation

Q: Tell me about a time when you had to explain to a team member why your model performs well when interpolating within the data range but might not be reliable when extrapolating beyond it?

问：请讲述一下这样的一次经历：当时你必须向团队成员解释，为何你的模型在数据范围内进行插值时表现良好，但当超出数据范围进行外推时却可能不再可靠?

A: I'd explain that our model learns the "map" of the training data. **Interpolation** is like finding a point *inside* that map, which is reliable. **Extrapolation** is asking for a point *outside* the map, where the model has no data and its learned rules may fail, making it unreliable.

回答: 我会解释说，我们的模型学习了训练数据的“地图”。**插值**就像在地图*内部*找一个点，这很可靠。**外推**是要求一个地图*外部*的点，模型在那里没有数据，它学到的规则可能会失效，因此不可靠。

Interview回答

This actually came up in one of my projects when a teammate asked why our regression model made great predictions within the training range but completely broke down when we tried to predict beyond it.

I explained that the model basically learns the *map* of the data it has seen — interpolation means we're finding points *inside* that map, where the model has real experience and can make reliable predictions. But extrapolation is like asking the model to guess what's happening *outside* the map — in regions where it has no data. The relationships it learned inside may no longer hold, so it can easily give unrealistic results.

I also added that if we really need to extrapolate, we should either collect more data covering that range, or use models based on physical or theoretical principles rather than purely data-driven ones. That usually helps make the predictions more trustworthy.

我确实遇到过这种情况：有一次团队成员问我，为什么我们的回归模型在训练数据范围内预测得很好，但一旦超出这个范围就完全不准了。

我解释说，模型其实是学习了它“见过的数据地图”。**插值**相当于在这张地图的“内部”找位置——模型在那片区域有经验，预测会比较可靠。**外推**则像是在让模型去“地图之外”猜测，那片区域它从没见过，所以学到的规律很可能不再适用，预测也容易失真。

我还补充说，如果确实要做外推，可以考虑两种办法：要么收集更多覆盖那部分的训练数据，要么采用有理论或物理基础的模型，而不是纯粹依赖数据驱动。这样外推的结果才会更稳健。

Q9: Generative Models for Hypotheses

Q: Imagine you've developed a generative model that can suggest multiple hypotheses/possibilities about possible mechanism. How would you use those generated insights to guide real-world experiments?

问：假设你已经开发出一个生成模型，该模型能够就可能的机制提出多种假设/可能性。那么，你会如何利用这些生成的见解来指导实际的实验呢?

A: The generative model learns a "latent space" of possibilities. We can sample from this space, or interpolate between known points, to generate novel hypotheses. We would then take these specific generated designs—like new molecules or material structures—and attempt to synthesize and validate them in a real-world lab experiment.

回答: 生成模型学习了一个可能性的“潜在空间”。我们可以从中采样，或在已知点之间插值，以生成新颖的假设。然后我们会将这些生成的特定设计——比如新分子或材料结构——带到现实世界的实验室中去合成和验证。

Interview回答

If my generative model can propose multiple hypotheses about possible mechanisms, I'd use it as a *guide* for experiment design rather than a replacement for it.

The model essentially learns a *latent space* — you can think of it as an “idea landscape” where each point represents a plausible mechanism or structure. By sampling different regions of that space or interpolating between known examples, we can generate new, testable hypotheses.

Next, I'd rank those generated candidates based on physical plausibility or predicted performance — for example, which molecule seems most stable, or which material has the most promising properties — and then bring the top few into the lab for synthesis and experimental validation.

This approach is powerful because it lets experiments move from being purely trial-and-error to being *data-guided* — the model helps us explore smarter and faster, but real-world validation still closes the loop.

如果我的生成模型能提出多种关于潜在机制的假设，我会把它当作**实验设计的向导**，而不是取代实验本身的工具。

这个模型其实学习的是一个“潜在空间”，你可以把它理解成一张“创意地图”，其中的每个点都代表一种可能的机制或结构。我可以从这个空间中采样，或者在已知点之间插值，去生成新的、可以验证的假设。

接着，我会根据物理可行性或模型预测表现来对这些生成的候选方案进行排序，比如哪些分子更稳定、哪些材料性质更优，然后挑出最有希望的几个在实验室中合成并验证。

这种方法的优势在于，它让实验从“盲目试错”变成“数据引导”——模型帮助我们更聪明、更高效地探索，而现实实验则是验证和完善整个循环的关键一步。

Q10: Choice of Loss Function

Q: Can you describe a situation where your choice of loss function strongly influenced the outcome of your scientific model, and how you explained that choice to others?

问： 能否描述一个情境，在其中你所选择的损失函数对你的科学模型结果产生了显著影响？你又是如何向他人解释你做出这种选择的理由的？

A: In the Plant Pathology project, the "multiple_diseases" class was very rare. Using standard Cross-Entropy, the model just ignored it. By switching to **Focal Loss**, which down-weights easy examples, we forced the model to focus on this hard, rare class, which significantly improved its detection rate.

A (中文): 在植物病理学项目中，“多种疾病”类别非常罕见。使用标准交叉熵，模型会直接忽略它。通过切换到**Focal Loss**，它会降低简单样本的权重，我们迫使模型专注于这个困难的稀有类别，从而显著提高了它的检测率。

Interview回答

In one of my projects on plant pathology, we had a dataset where one class — called *multiple diseases* — was extremely rare compared to healthy or single-disease samples.

When we first trained the model using standard Cross-Entropy loss, it basically ignored that minority class — the accuracy looked great overall, but the recall for *multiple diseases* was almost zero. So I switched to **Focal Loss**, which down-weights easy, overrepresented samples and puts more emphasis on the hard, rare ones.

After that change, the model started paying attention to those few critical samples, and the detection rate for the rare class improved significantly. When explaining this to the team, I put it simply: *"We changed the loss so the model stops chasing easy wins and starts focusing on the cases that matter most."* That really helped everyone understand the trade-off and the reasoning behind the choice.

在我做植物病理学项目时，我们的数据集中有一个类别叫“多种疾病”，样本非常少，远远低于健康或单一病害的样本数量。

最开始我们用标准的交叉熵损失函数训练模型，结果模型几乎完全忽略了这个少数类——整体准确率看起来很高，但“多种疾病”的召回率几乎为零。于是我改用了 **Focal Loss**。这种损失函数会降低那些“容易分类”的样本权重，让模型把注意力放在困难、稀有的样本上。

切换之后，模型终于开始“看到”这些关键样本，稀有类别的检测率显著提高。后来我向团队解释时，用了一个通俗的说法：“我们换损失函数，是让模型别再只追求简单的正确答案，而去关注那些真正重要、容易被忽略的少数情况。”这样一讲，大家都立刻明白了背后的逻辑。