Springboard

25 Data Science Interview Questions—With Answers

Springboard

Table of Contents

Introduction 2
Statistics Questions 3
Programming Questions 10
Modeling Questions 18
Problem-Solving Questions 28
Behavior Questions 33
Culture Questions 35
Additional Resources 38

Introduction

No matter the industry or the role, interviewing for a job can be stressful and awkward. Through fairly limited interactions, you're trying to convince a bunch of strangers to hire you—to spend eight hours a day with you—instead of the dozens of other people they've considered.

And when you're on the hunt for a data science role, you have the added pressure of tackling tough technical tests. You may have to solve probability puzzles and write some SQL, then quickly pivot to more casual conversations designed to determine whether you're a cultural fit.

Even if you're fully confident in your skills, it's typically a tremendously taxing experience.

The key to handling pressure and managing stress during the interview process is preparation. And while there's no way to practice every possible question that might come your way, you can increase your confidence by working through sample scenarios and getting guidance from data scientists who have successfully navigated the process.

To help you nail your next interview, we curated a list of 25 data science interview questions that fall into six different categories: statistics, programming, modeling, behavior, culture, and problem-solving. We then asked four data scientists currently working in the field to weigh in with direct answers and/or insights into what would make an answer stand out. For each question, there will be at least two answers, giving you different perspectives on how to construct your response.

Before we get to the questions, **let's introduce the data scientists:**

Michael Beaumier is a data scientist at Google who previously worked in machine learning and data science at Mercedes Benz Research.

Ramkumar Hariharan is a senior machine learning scientist at macro-eyes and a mentor for Springboard's Data Science Career Track.

Mansha Mahtani is a data scientist at Instagram who also was a data scientist at Blue Apron.

Danny Wells is a senior data scientist at the Parker Institute for Cancer Immunotherapy and a mentor for Springboard's Data Science Career Track.

Now, on to the Q&A.

Statistics

1. What is the difference between a type I and type II error?

Michael: Type I and II errors are just a fancy way of talking about hypothesis testing. Hypothesis testing is the practice of asking if an observation is consistent with a statistical model or not. The null hypothesis states that there is no relationship between an observation and a model (or data set A and data set B). Now, just because we can state that an observation is consistent with a model doesn't make us right. This is where type I and type II error comes in.

Type I error occurs when you reject the null hypothesis, but are in actuality wrong. That is to say, you assume that there is a relationship between an observation and model, but in fact there is no relationship. Type II error is the inverse: in this case, you assume that there is no relationship between an observation and a model, but in fact, there is.

Very succinctly, type 1 error = false positives, type II error = false negative.

Ramkumar: Type I and type II errors are typically used in significance testing. Say we have two sets of scores. The question we ask is: do these two sets of scores come from the same underlying normal distribution or do they belong to two distinct normal distributions? Please note that we assume the distributions are normal or bell-shaped (the distribution looks like a bell and has a hump and two tails on either sides).

We start with the null hypothesis, saying that the two sets of scores come from the same distribution. Our goal is to find evidence to reject the null hypothesis. If the two sets of scores come from the two different ends of the distribution, then depending on the significance level (say p = 0.05, which controls how close to the ends of the distribution we are looking at), we may think they come from different distributions. So, we reject the null hypothesis falsely (sometimes called false positive) and get a type I error. It's considered a serious error and can be controlled to some extent by choosing a stringent p-value, say 0.01. This means that unless the two sets of scores come from the last two tiny slivers of this distribution, we will not reject the null hypothesis.

A type II error is the opposite. We say that we don't have enough evidence to reject the null hypothesis, when the scores actually come from different distributions. This may be due to the greater degree of overlap between the two distributions, and we cannot say if the scores belong to one distribution. We get false negatives with the type II error.

Mansha: Although a question like this may not be directly asked, it's often incorporated into a question about A/B testing or significance testing. Simply put, a type I error is known as a false positive (i.e., observing a difference when in reality there is none). A type II error is known as a false negative (i.e., failing to observe a difference when in reality there is one).

In an A/B experiment, type I error rate is also known as the significance level (***α***), while confidence level is 1 - ***α***. By reducing the type I error rate or increasing the confidence level, you can reduce the likelihood of observing a difference between variants A and B if none actually exists.

However, bear in mind that if you are trying to reduce a type I error, it is only natural that you will be less likely to observe a difference that actually exists. Reducing type I error rate tends to increase the prevalence of type II error rate (***β***) and therefore decrease the "power" of the test (1 - ***β***).

Interviewers often ask this followup question once you identify the trade-offs in type I and type II: how do you reduce both type I and type II errors for a given test? Typically, interviewers are looking for a reference to increasing the sample size.

2. What is linear regression? What do the terms p-value, coefficient, and r-squared value mean? What is the significance of each of these components?

Mansha: A linear regression is a method to observe the relationship between a dependent variable, denoted as y, and one or more independent variables, denoted as x. Below are a few key points to keep in mind when talking about linear regression:

- The dependent variable must be able to take a range of values (i.e., continuous or close to continuous)
- This method assumes the independent variables are linearly related to the dependent variable
- Linear regressions establish correlation, not causality; A/B tests are a way to establish causality

The linear regression will help you independently control for the independent variables in order to quickly identify the direction of the relationship of these variables (positive or negative) with the dependent variable, how strongly they are correlated (r-squared), and finally what the relative importance of them is (coefficients).

Simply put, the r-squared value provides an estimate of the strength of the relationship between the independent variables in the model and the dependent variable.

Interviewers may want you to explain the practical significance of r-squared and a common mistake is to assume that a low r-squared value is bad. However, keep in mind that this is dependent on the type of behavior you're looking to explain or predict. As a data scientist, you are limited by what you can actually measure. Often, human behavior is harder to predict than physical processes. For example, if you want to predict if Annie is likely to purchase a red sweater on September 1, 2021, and the biggest predictor of this is how impulsive Annie is feeling on September 1, 2021, your model will be restricted by the difficulty of capturing Annie's impulsiveness on September 1, 2021.

You could have a low r-squared value but the factors in your model are statistically significant (i.e., have low p-values). If the temperature on September 1, 2021 is 10% of the reason why Annie purchased the sweater and we can say that with high confidence (p-value of temperature is lower than our significance level), this could still be valuable information.

Coefficients establish the relative importance of variables. If the price influences 20% of the decision for Annie to purchase a sweater, then the coefficient will help you understand that price is more important than temperature.

Ramkumar: Linear regression has two parts to its definition: It's a linear model, meaning that you are starting out with a straight line to model your data. And regression means we want our model to predict a continuous variable or number.

If you have three different dimensions, two Xs and a Y, then you will get a 3D plot of data points. We start with a straight line and the goal is to make this straight line go as near as possible to these data points in 3D. In effect, we are trying to learn the slope of this line and where it touches the y-axis (the intercept). Once we learn its slope and intercept, we have the linear regression model. If we get new Xs, we can use this line to predict the Ys. We can have many, many dimensions in our model if we have data for those features.

Thus, simply put, our goal is to find a number (or “slope”) for each dimension which we will multiply with that feature. We then add all these products or terms and finally also add a constant (intercept) to get a number that is as close as possible to the Y value. The slope for each feature is the parameter, or coefficient, and the p-value will tell us if each feature makes a useful contribution to predicting Y or not. The r-squared tells you how much of the variance in the Y variable is explained by the model. A terrible model can have negative r-squared values, while the best model will have an r-squared of 1.

Michael: The simplest form of linear regression is fitting a line to a set of data. As you get into higher dimensions, the best way to think about it is that you can parameterize the relationship between input features, and an outcome with a set of coefficients that never change.

Coefficients refer to the weights (numbers) that are multiplied against each input feature to produce a prediction of an output feature (after summing the results of multiplying every feature with its associated coefficient).

R-squared, also known as the “coefficient of determination,” defines how much of the variance of the data is captured by the model. R-squared of 1 means 100 percent of the variance of the data is explained by the model, while r-squared of 0 means 0 percent of variance is explained.

P-values measure how statistically significant each feature in your regression is to predicting the outcome. The p-value tests for the probability that each feature is not useful as a predictor, so the higher the p-value, the more likely it is that the feature is not useful, and the lower, the more useful.

3. What are the assumptions required for linear regression?

Ramkumar: Some of the key assumptions are (1) low or no correlation between any two variables, (2) there is a linear relationship between the independent variables and the dependent variable, and (3) the residual errors (the difference between model-predicted Ys and actual Ys) are normally distributed.

Michael: Linear regression assumes that the relationship between the input feature space and an outcome is parameterized with a set of weights that never change. This is another way of stating that the outcome variable is simply a linear combination of the input features. “Never change” means that the same linear combination always predicts equally well—i.e., that the data is not heteroskedastic. Finally, linear regression assumes that features themselves are not correlated to each other.

Mansha: When articulating assumptions in a linear regression, it is often helpful to include examples.

Assumption: Linear regressions assume a linear relationship between the independent variable and the dependent variable.

Age and height could be strongly related—not in a linear fashion, but rather a logistic one. There is a point at which height tends to plateau once someone hits a certain age. An interviewer may ask you how you would account for this in your model and a common answer is to alter the feature to account for the relationship it has. For example, instead of including age in your model, you could change the feature to log(age).

Programming

4. Describe a data science project in which you worked with a substantial programming component. What did you learn from that experience?

Michael: I built AleTrail.com, a web app to dynamically generate pub crawls based on intuitively extracting flavor profiles (generated with machine learning and natural language processing) from beer reviews and funneling the front-end user through different flavor profiles to help guide them to a delicious beer walk. I used Python and Python libraries to build a web server, parse user input, and look up machine learning predictions in a precalculated SQL database.

From this experience, I learned the importance of abstracting my codebase into functional parts and how important it was to define interfaces between code for modularity and interoperability.

Ramkumar: I once led a data science project where I had to get multiple kinds of data from different web resources. To this end, I built different kinds of web scrapers and a Twitterbot. The web scraper got me images from a set of websites, and the Twitterbot brought in tweets with certain keywords. I quickly realized how powerful the Python programming language was since I did everything, including data gathering and all downstream data pre-preprocessing, machine learning, and data visualization steps, in the same language!

Mansha: In questions like this, it is imperative to focus on the importance of the various steps in the project and the practicality of performing those steps.

Keep in mind that interviewers are not just trying to gauge your technical abilities, but also the trade-offs you made in the project. If there were technical limitations to obtaining a certain feature, did you creatively come up with a proxy feature?

Often, important features for your model may not be available in your current data sets and certain features require you to set up a data extract from an external source. Going this extra mile for good quality variables may result in additional effort, but can be well worth it if it helps boost the performance of a model significantly.

5. Explain how MapReduce works as simply as possible.

Mansha: Sometimes the best way to explain a complex concept is to use a simple example. MapReduce can be thought of in two parts. “Map” is akin to delegating a task to a group of people and “reduce” is combining the result of each person’s effort to produce the final output. From a technical perspective, the group of people are “worker nodes” while the person coordinating the efforts is the “master node,” In a question like this, it is important to mention the purpose of MapReduce—that is, the processing of large amounts of data in a parallel manner.

Michael: MapReduce abstracts a calculation job and a summarization job, allowing one longer serial process to be parallelized over an arbitrary number of computers (mapping) to enable faster computation time. These parallelized computations are “reduced” by summarizing all the parallel computation and remerging.

Danny: MapReduce is a framework to enable massively parallel computing over very large data sets. I think it is best explained with an example:

Suppose you ran a zoo and wanted to calculate the total mass of all N animals in your zoo. Well, one thing you could do is weigh all animals in the zoo one at a time, record their masses, and add them up. If each animal takes one hour to weigh, then the total amount of time to calculate the sum mass of all animals will take N hours. This is analogous to a traditional linear approach.

Alternatively, to calculate the sum mass, you could find M unpaid interns (everyone wants to work at the zoo!) and assign each of them to weigh a subset of the animals. To make it easy, you assign each intern to a single species, but you first have to train them about which species are which, which takes time T. But once trained, the interns can work in parallel and find the sub-total mass of their particular mass—that is, each intern can find the total mass of all their species—and then you could just add up those sub-sums to find the total mass. Mathematically, the total time to calculate the total mass will then be (roughly):

$$T + N/M + M$$

Thus, by mapping each species to an intern, having them calculate the sub-mass in parallel, then reducing those sub-sums to a final sum, you have accelerated your calculation. This is exactly how MapReduce works, where “interns” are now independent computing cores.

6. How would you sort a large list of numbers?

Mansha: Although this question is more typical in software engineering interviews, understanding this can be helpful when evaluating which functions to use in your analysis. A common sorting algorithm is mergesort. In simple terms, mergesort is the process of sorting through dividing the list and sorting the list independently, and eventually combining the independent lists to perform the same iterative process.

Sorting algorithms which are comparing a single number against every other number are less efficient but still accomplish the same goal. The interviewer is interested in knowing whether you are able to appreciate how different approaches of solving a problem could result in different computational effort.

Ramkumar: Either mergesort or quicksort can be used. While quicksort is faster, mergesort may be more stable for very large arrays of numbers.

Michael: It depends on how large the list of numbers is and how much memory the computer I was using to sort the numbers had. For most cases, I would just use a pre-built sorting algorithm, such as Python's "sort" function. If the list of numbers is very large, I might need to use a method that can do out-of-core operations (i.e., sorting a subset of the list, serializing the middle step, sorting another part of the list) and then merge back together.

7. What is the difference between a tuple and a list in Python?

Mansha: This question may be masked in a larger question on Python. Although both tuples and lists store a collection of items, the key difference is mutability. Items in a list can be updated but items in a tuple cannot be updated (i.e., they are immutable). Often, tuples are used over lists to improve performance and speed up processes.

Michael: A list is mutable and a tuple is immutable. Mutability means that after instantiation, an object can change (mutate), such as with lists, which support re-assigning values in the list, popping or pushing values. Once you instantiate a tuple, you can never change it.

Ramkumar: Both tuples and lists are built-in data structures in Python, to hold an array of objects. The key difference between a tuple and list is that lists are mutable while tuples are not. What this means is that there are many "things" or operations you can do to a list and this will result in the original list being changed forever!

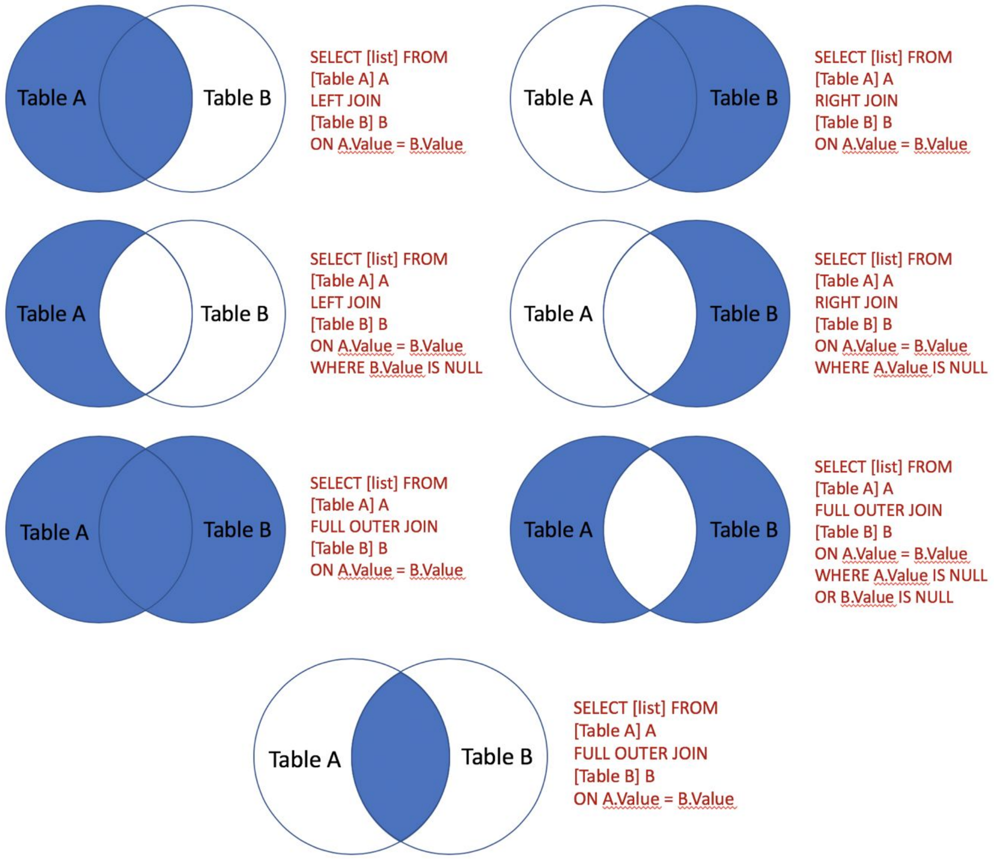
Python tuples do not allow any operation on it that will change what it contains. It's a good idea to store a sliced copy of your original or starting list before operating upon it—just in case you may want to retrieve the original list at some point. Also, it's a good idea to store things that should not change inside a tuple.

8. Tell me the difference between an inner join, left join/right join, and union.

Mansha: Although these questions may not be directly asked, they may come in the form of a SQL interview question. It is helpful to understand the purpose of these functions when answering SQL questions.

- An inner join helps you find the commonalities between two data sets and removes the rows on both data sets that aren't in common
- Left join helps you keep all the rows on the first data set and removes all the rows on the second data set that have no commonalities with the first data set
- Right join helps you keep all the rows on the second data set and removes all the rows on the first data set that have no commonalities with the second data set
- Union adds two tables with the exact same columns on top of each other to create a larger data set

Danny: This is best described by a picture:



```
SELECT [list] FROM
[Table A] A
LEFT JOIN
[Table B] B
ON A.Value = B.Value
```

```
SELECT [list] FROM
[Table A] A
RIGHT JOIN
[Table B] B
ON A.Value = B.Value
```

```
SELECT [list] FROM
[Table A] A
LEFT JOIN
[Table B] B
ON A.Value = B.Value
WHERE B.Value IS NULL
```

```
SELECT [list] FROM
[Table A] A
RIGHT JOIN
[Table B] B
ON A.Value = B.Value
WHERE A.Value IS NULL
```

```
SELECT [list] FROM
[Table A] A
FULL OUTER JOIN
[Table B] B
ON A.Value = B.Value
```

```
SELECT [list] FROM
[Table A] A
FULL OUTER JOIN
[Table B] B
ON A.Value = B.Value
WHERE A.Value IS NULL
OR B.Value IS NULL
```

```
SELECT [list] FROM
[Table A] A
FULL OUTER JOIN
[Table B] B
ON A.Value = B.Value
```

Michael: Unions just concatenate the rows of table A onto the rows of table B, adding null columns in cases where the tables don't have the same columns.

Modeling

9. What are your favorite data visualization techniques?

Ramkumar: My favorite data visualization technique depends on the problem we are intending to solve! It also depends, obviously, on the kind of data we are trying to visualize (e.g., continuous vs. categorical).

That said, I love using clustermaps in some of my analysis. Clustermaps can be very useful for visualizing multiple dimensions. For one, you can see a color-coded variation across three different features or dimensions on a 2D plot. And when you apply clustering on either dimension, you get to see correlation based structures in the data.

I also love simple bar plots that can show fundamental trends in the data. And you can see the mean and standard deviation very clearly in a well-constructed bar plot.

Michael: I like using matplotlib with seaborn to visualize data. Generally, I find statistical-based summarizations such as box plots or violin plots to communicate relationships most clearly.

Mansha: As a data scientist, a large part of your role will be to communicate insights in an understandable way. The visualization technique you choose will be highly dependent on the context of the problem, the message you are trying to land,

and your audience. In general, there is no compulsion to choose one tool over the other as long as the visual is simple to digest by the expected audience.

10. How would you create a logistic regression model?

Michael: I would load the appropriate scikit-learn logistic regression library and fit my data with LR, along with the typical things, like cross-validation, class-confusion, precision, and recall metrics for model evaluation.

Ramkumar: A logistic regression model is simply a linear regression model "squished" by a logistic or sigmoid function. A logistic function non-linearly maps negative numbers to values less than 0.5, and positive numbers to values greater than 0.5. However, the output range for the logistic function is between 0 and 1. This fits in very nicely with the probabilistic view of classifying an object as either one class (0) or the other (1).

A linear regression simply tries to learn the slope and intercept of a straight line model to the data.

Mansha: Right off the bat, the interviewer will be expecting a reference to the binary nature of the response variable in a logistic regression. Although you may not get asked this question directly, an interviewer may ask what model you would use to predict whether an event will occur or not. A common mistake is to not understand the nature of the response variable and choose linear regression instead of logistic regression. You may also be expected to describe how you would transform a data set with string values into categorical variables (0,1) for the purpose of the model.

11. Explain what precision and recall are. How do they relate to the ROC curve?

Michael: Precision describes the fraction of relevant samples your model gets correct. Recall tells you how many relevant samples are chosen by your model. Precision is also defined as "true positives" / "true positives + false positives." Recall is alternatively defined as "true positives" / "all true data." The receiver operating characteristic curve is the plot of true positive rate vs. false positive rate.

Ramkumar: Precision of a classification model is defined as true positives divided by the sum of true positives and false positives. Recall or sensitivity is defined as true positives divided by the sum of true positives and false negatives. So, what are true positives, false positives, and false negatives? When your model says items actually belonging to a positive class are positive, we call the predictions true positives. And when your model says items belonging to the negative class are negative, the predictions are called true negatives. Our model can predict actual positive items as negatives—these are called false negatives. Our model also can predict actual negative items as positive—these are called false positives.

To build the ROC, you plot the sensitivity or recall on the y-axis against fallout or 1-specificity on the x-axis. Sensitivity values range from 0, which is no sensitivity, to 1, which is best. Also, fallout goes from 0 to 1. To make the ROC, we keep changing the threshold a little bit at a time. We then plot the sensitivity and fallout at each new value of the threshold. Then we draw the curve connecting all these points. The shape tells us how good our feature or model is. ROCs with Area Under the Curve (AUC) close to 1 are the best.

Mansha: Precision is the percentage of times your model was actually right when it identified something as true. Recall is the percentage of actual true positives your model identified correctly.

Here it is important to understand that there is a difference between a precision-recall curve and ROC curve. ROC curve represents a relation between sensitivity (i.e., recall) and false positive rate (not precision). In a typical ROC curve, the x-axis has the false positive rate (1 - specificity) while the y axis has "recall" or the true positive rate.

It is important to be comfortable reading an ROC curve. In general, the closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test. The area under the curve is a measure of accuracy of the model.

12. Explain the difference between L1 and L2 regularization methods.

Michael: Both add penalties to the cost function of a model—L1 penalizes residual error absolutely, whereas L2 penalizes the square or residuals.

Ramkumar: L1 is lasso regression while L2 is ridge regression. L1 regression sets many parameters to zero, while L2 shrinks them. L1 is better for feature selection; however, the number of parameters or features returned cannot be greater than the number of samples. A combination of the two, elastic-net, is sometimes preferable with a mixing parameter controlling the relative contribution of L1 and L2.

The penalty term is different. In L2, squared coefficient values are added to the penalty, while in L1, absolute value of coefficients is used.

Mansha: In simple terms, regularization is the process of determining the right level of complexity in your model so it is better at predicting on more data sets (i.e., generalized). Without this step, your models may be too complex (overfit) or too simple (underfit), ultimately resulting in poor predictions.

Regularization adds a penalty for additional complexity in the model. L1 (Lasso regression) and L2 (Ridge regression) regularization have different penalty terms in order to reduce complexity. The main difference is that L1 methods tend to shrink less important features to zero and can be helpful in removing features in a complex model.

13. What is one way that you would handle an imbalanced data set that's being used for prediction (i.e., vastly more negative classes than positive classes)?

Ramkumar: Imbalanced data sets are pretty common in the real world. One way of taking care of this is by undersampling the dominant class, oversampling the smaller class, or using a combination of both approaches. The Python package imbalanced-learn has several algorithms that can be invoked using just a few lines of code. This package is also compatible with the scikit-learn machine learning library.

Michael: I would try different approaches depending on how much data I had. You could try regularization, resampling the underrepresented class, undersampling the overrepresented class, or training a generative model to generate equal numbers of both classes to create a training set.

Mansha: In general, the common answer here is to either upsample the minority class or downsample the majority class. However, as a bonus, it can be helpful to reference the pros and cons of these methods. Downsampling causes information loss while upsampling the minority class can result in overfitting.

14. I have two models of comparable accuracy and computational performance. Which one should I choose for production and why?

Ramkumar: The model that can explain feature importance better is the better choice.

Michael: It depends on the constraints of the production environment. For example: how long does it take the models to make a prediction? How often do the models need to be retrained? Does the business require interpretability of the models, or are they OK with a black box?

If I got this question, I would start by listing the various ways external factors might rule out one model over the other, then I would tell a story about how to place the remaining model into production.

Mansha: This question is evaluating your ability to break accuracy down into false positive and false negative rates as well as your ability to think practically. For the specific problem, if the cost of a false positive is higher than the cost of a false negative, it is preferable to go for a model that reduces false positive rates.

15. Is it better to spend five days developing a 90-percent accurate solution or 10 days for 100-percent accuracy?

Michael: It depends on what the penalty of error is and how quickly a solution needs to be in place. Generally, it is unrealistic to achieve 100-percent accuracy—you could double your impact with two 90-percent projects vs. one 100-percent project in the same period of time.

Ramkumar: It really will depend on the situation. Usually, data scientists come up with a quick, less-than-optimal model. And this can be improved later. If the application demands 100-percent accuracy, then there is no question since settling for 90-percent will make the endeavor meaningless.

Mansha: This question is evaluating your ability to understand the impact of your recommendations and ability to prioritize.

Depending on the impact of the decision, this could vary. Of course, 100-percent accuracy is rare, but if the wrong decision leads to incredible costs (e.g., decisions in healthcare), reaching higher levels of accuracy is important.

However, if the decision you are driving just requires a directional answer and spending more time will not add any incremental value, the faster you are able to answer the question the better. It is important to ask yourself: if my answer changes by X percent, how will the business or product decision change?

16. When modifying an algorithm, how do you know that your changes are an improvement over not doing anything?

Mansha: Like with any problem, the first step is to identify the goal and success metric. Once that is established, any updates to the algorithm can be evaluated against the success metric.

Michael: You should always have unit tests for your algorithms and clear metrics that define whether your work is doing something good or bad. If you don't have this infrastructure, you can't reliably produce work that people can trust.

Danny: I think it depends on the case. In some situations, like if you're trying to improve the speed, memory usage, etc., of an algorithm, then it's as easy as applying your new approach to an established set of standards and seeing if you have improvement.

If you're trying to improve, say, the prediction value on a machine learning algorithm, it gets harder, because you have to think about overfitting. Say you left out a test data set in your initial training, and after training your model you use this test set to evaluate your model. If you continue to use this exact test data set over and over as you tweak your model, you will (almost assuredly) be able to improve your model. However, since you're using the same test data set over and over, it could be that you're just fitting your model more and more to the test data, not increasing its overall predictive power.

To avoid this, ideally you're testing your model against fresh data (never before used for any training for this model). In practice, this might be impossible to get. So, at the very least, it's important to take random train-test splits each time you try out a new model. This will help (but not ensure) that when you see an improvement in your model, it is real.

17. Is it better to have too many false positives or too many false negatives?

Mansha: It depends on the problem and what is at stake. If the cost of a false positive is higher than the cost of a false negative, it is preferable to go for a model that reduces false positive rates.

Michael: It depends on the needs of the model. If the cost of a false positive is huge (an autonomous car kills someone, for example) then you should minimize false positives to zero, even at the expense of more false negatives.

Danny: This is very application-dependent and really comes down to the comparative cost of false negatives and false positives. In cancer diagnostics, you may be OK with having two false positives for every true positive, since a false negative potentially means cancer going undiagnosed (very very bad), while a false positive might lead to an unnecessary biopsy (bad, but not as bad as missing the cancer).

Alternatively, say you're building a movie recommendation engine. In this case, an excess of false positives (movies you recommend that a user hates) may lead to users losing trust in your tool (bad) while false negatives (missing a movie a user would like) are less bad since there are only so many movies a person will watch.

Problem-Solving

18. How would you come up with a solution to identify plagiarism?

Mansha: Interviewers are not expecting you to have studied a complex algorithm to detect plagiarism. Instead, they are interested in understanding your thought process behind solving a nuanced and complex problem.

It is often helpful to first rephrase the problem to capture what it is evaluating. Detecting plagiarism is another way of determining whether content in a specific document is found elsewhere. There are key components of this question candidates can address.

1. What is the repository you will use to check document X against?
    a. All Google search results?
    b. Repository of academic papers?
2. What are the key parts of your algorithm? Example considerations below:
    a. Even if you tokenize and lemmatize the document, you will need to ensure the sequence of the words (i.e., tokens) stays the same.
    b. It may not be necessary to perform checks against generic stopwords like “is,” “the,” or “and.”
3. What is the threshold you will use to determine “plagiarism”?
4. Will you require manual review after your solution detects high probability of plagiarism?

Michael: A critical part of plagiarism is that large portions of work are copied exactly. The simplest solution would be to identify the longest common sequences of characters between two works, and use heuristics to raise a flag if too much overlap was determined.

Plagiarism can be tricky, though, because you can rephrase and use different words but still be copying. In this case, I might do a literature review to see if I could use sequence embedding to find similarity metrics between all character sequences, and count how similar how many sequences between two works were.

Danny: In general, this is a very hard problem. However, I think the key thing to realize is, due to the huge size and complexity of the English language, plagiarism can be detected only by comparing small snippets (5-10 word n-grams) to an existing database of text. Using this approach, you could generate a "fingerprint" of a document by looking at the frequency of n-grams in that document, and compare this fingerprint to other documents. Documents in your database that match the fingerprint too closely could be prioritized for manual review.

19. How would you detect bogus reviews, or bogus Facebook accounts used for bad purposes?

Danny: This is another hard problem. I have a few suggestions:

Bogus reviews:

1. Identity: The key here is to have confidence that a user is real, based on their previous posts, reviews, etc. For example, a profile that has been around for many years and only posted the occasional review is probably more likely to be real than a new account suddenly publishing lots of reviews.

2. Confirmed purchases/attendance/etc.: Amazon is moving this way. If you have an idea that a user is real (lots of purchases made over time to the same address), then if that user buys a product, a review on that product is more likely to be real than a review from someone you can't confirm bought the product.

3. Linguistic complexity/details: A bot (or a troll) is not incentivized (or able) to write a two-page review full of precise details. Long and/or complex reviews are more likely to be real than short "good product 5 stars" reviews.

Bogus users:

1. History and identity: This probably matters the most. Real users likely do more than just post reviews and/or comments on a specific set of products/topics all the time. So when such users are identified, they have a higher chance of being fraudulent.

2. Unexpected linguistic mistakes: A user supposedly born and raised in the U.S. is presumably less likely to make basic English grammar mistakes compared to someone pretending to be born and raised in the U.S. (of course, exceptions apply).

Most trolls and review farms have adapted to detection techniques like the above, so it very much is a case of requiring continual adaptation.

Michael: I would start with a simple model like Naive Bayes text classification with labeled data and then measure its performance, and try to identify the factors in the text which might lead to easily detecting a bogus review/bad actor. I would merge the textual information with other contextual features, such as origin/destination IP addresses for posters to accounts, time between posts, etc., to try to allow for certain behavioral elements to be modeled/discovered

/measured.

Mansha: In this question, an interviewer is interested in understanding two aspects: 1) your ability to use technical concepts in a real-world problem and 2) your creativity in brainstorming features to solve the problem.

Consider training a model on a sample of reviews that already have fake reviews identified. A common approach is to use a manual review to tag a sample of reviews that forms the basis of your training data.

When developing features, you can rely on your intuition or interview experts. If you were browsing for fake reviews, what are the characteristics of the review that would seem fishy to you?

The below are example features:

- Is the review overwhelmingly positive? (Note: sentiment analysis will be useful here.)
- Are there similar reviews on other products?
- Are there multiple reviews from the same reviewer around the same time?
- Is the reviewer located in a place where he or she could access this product or service?

Followup questions may cover the evaluation of false positives and false negatives using precision and recall measurements. There will always be limitations to the features you can measure and it's often good practice to raise those to the interviewer.

Past Behavior

20. Tell me about a time when you resolved a conflict.

Michael: In middle school, I was a peer mediator. As part of this experience I learned it was important to first listen to the grievances of the parties involved individually. Then, I would invite each party to repeat the concerns of the other party in their own words. I would find common ground, and point out areas for compromise.

Ramkumar: We once had a situation when our team was waiting on another team's data. After repeated requests, the team did not respond. So, I decided to have a 1:1 conversation with that team lead. I understood that they were short-staffed and had reservations about making any comments public. I resolved the conflict by offering to extend my time and effort to help the other team gather the data. This led to a happy situation and reinforced inter-team support and respect at my organization.

Mansha: In similar behavioral questions, you are not only expected to provide a structured answer, but also expected to articulate what you learned from the experience. The STAR framework can be handy to help structure your answer: situation, task, action, result.

21. Tell me about a time you failed and what you have learned from it.

Danny: As a scientist, I fail all the time, and (hopefully) learn from each failure. I think the failures I have become most interested in identifying are failures of leadership and teaching. For example, when I was a graduate student I wrote a (very!) large code base to simulate a growing tumor and how it interacts with the immune system (research you can find here if you're interested). The model took years to make. In the end, it worked well and formed the foundation of something that could have been a larger research program. However, I failed to adequately transfer learning around this model and teach other members of my lab how to run it, use it, change it, etc. Because of this, once I finished my degree and transitioned jobs, the model hasn't really been used again and the research around it has not emerged to the extent that we had originally hoped. This emphasized to me the importance of knowledge transfer, as well as the importance of teaching and mentoring others around areas I'm passionate about.

Mansha: When describing a "failure," it's important to take ownership over what went wrong. Interviewers want to make sure you are self-aware and are able to take responsibility for your actions. Most importantly, they are looking for candidates who quickly learn from their past mistakes and are looking to constantly improve.

22. What have you done in the past to make a client satisfied/happy?

Michael: I presented to the board of directors the latest results of my analysis/modeling. By boiling down the results of the study into key metrics, as well as presenting a simplified version of the model I worked on that traded accuracy of representation for layman understandability, I was able to help involve the board in my thought process and feel like they could understand the results on a deeper level.

Mansha: Oftentimes a stakeholder is pleased when you are able to go above and beyond. In a question like this, ensure that you describe a challenging project with clear articulation of the positive impact on the client.

Culture Fit

23. What do you think makes a good data scientist?

Michael: Curiosity. You have to want to solve problems that seem hard without getting discouraged because they are hard. You have to be willing to forge ahead and try things without necessarily "learning everything first." You have to communicate when you're stuck, and develop strategies to get yourself unstuck.

Ramkumar: I think the key is to have motivation, and then temperament to keep learning as new advances flood the field.

Mansha: A good data scientist has empathy for his/her end users and cross-functional partners. Empathy for the end user helps understand the most important problems to be solved while empathy for partners ensures you have enough context when solving a problem.

Good data scientists are also wary of confirmation bias. It is important to be comfortable with being wrong when confronted with data that may be counter to your intuition. Strive to make the right decision, not your decision.

24. How did you become interested in data science?

Michael: I like solving hard problems, and I like the idea that data can be used to enrich our understanding of the world. The idea of predicting the future using data that might not be obviously related to that future is exciting and cool.

Ramkumar: My interest in data science comes from my interest in healthcare. During my Ph.D. days in a cancer center in India, I got interested by a simple problem: different patients with the same stage and grade of cancer, receiving the same treatment, had vastly different survival outcomes. I wanted to see if (a) we can first predict which patients were most and least likely to survive, and (b) if we can find out which factors were responsible for this difference in survival. I built a simple, shallow neural network in matlab for this. It did not work well at all because back then, neural networks had not yet become deep neural networks, and moreover I had much less data than needed. But that got me hooked to machine learning!

Mansha: Being honest and true to yourself is key here. Interviewers don't expect data science candidates to come from specific backgrounds.

25. What is the latest data science book / article you read? What is the latest data mining conference / webinar / class / workshop / training you attended?

Ramkumar: The latest data science book I read was "Visual Explanations" by Edward Tufte, a data visualization expert. The latest data mining conference I attended was an advanced deep learning workshop at the University of Washington.

Michael: The last book I read was "The Book of Why" by Judea Pearl. In it, he lambasts data scientists as glorified curve fitters (they are) and explains why his theory of causal inference is a great way to ask the contrafactual: "What would have happened if I had done something different?"

The last seminar I attended was the seminar that I organize and run at Google for sales and marketing data scientists. The last conference I attended was an internal Google conference on research, statistics, and machine learning. The last external conference I attended was a cool causal inference conference: “Uncertainty in Artificial Intelligence.”

Additional Resources

The 25 questions featured in this guide were culled from a longer Springboard blog post. Check out the full list of more than 100 questions and answers here.

Other relevant posts from the Springboard blog:

20 Python Interview Questions and Answers

27 Essential R Interview Questions (With Answers)

Interview Prep: 40 Artificial Intelligence Questions

How to Ace the Phone Screen: Tips From a Recruiter

More interview questions:

From Towards Data Science: Top 30 data science interview questions

From The Must: 20 Interview Questions Every Data Scientist Should Be Ready to Answer

From Pathrise: 113 data science interview questions to nail your onsite

Finally, for more career-focused training, consider **Springboard's Data Science Career Track.**

Get a complete education in data science along with personalized career coaching—plus a job guarantee! Apply now.