# Data science interview

# ## Learning objective

By the end of this checkpoint, you should be able to \*\*prepare for a data science interview\*\*.

# ## Overview

All interviews exist to determine if you’re capable of doing the job that you’re applying to. As such, interviews for data science jobs are conducted to determine if you’re comfortable with both the technical skills required, such as writing Python code and using Git version control, as well as the more nebulous skill of solving business problems with data. Depending on the maturity of the data science team at the company that you’re interviewing at, the business problem-solving piece may be a smaller or bigger part of the job requirement; regardless, in interviews, it’s important to show that you have the ability to connect business goals and actionable recommendations to data science projects that you work on.

To assess if you meet these qualifications, most data science interviews include these steps:

1. Phone screen.
2. Take-home data challenge.
3. Technical screen.
4. Coding test.
5. Technical interview.
6. On-site interview.

Each of these steps is described in more detail below.

# ## Phone screen

The first step after getting through the resume screen is often a phone call conducted by a human resources professional or a recruiter. These individuals have varying degrees of technical knowledge and the purpose of this interview is for them to confirm that you are likely to be a good candidate both culturally as well as professionally. Although they won’t be able to fully vet your technical abilities, they will be able to determine if your skills and past experiences make you a good match for what the company is looking for in this particular role. These interviews are generally 30-40 minutes long. The best way to make sure that these go well is to communicate in a confident, professional, and enthusiastic manner.

The interviewer will introduce themselves and then, most often, describe the role and, potentially, a little about the company as well. They will ask if that role sounds of interest to you or ask you to describe the type of position that you’re looking for. It’s best to be direct and deliberate with your response, review the job description beforehand, and be ready to reiterate some of the job description components that resonate with you. Then the interview may ask questions such as “Tell me about your last project?” or “Describe your experience using machine learning models?” In both of these questions you want to provide enough details to communicate fully what you have done in the past; however, you don’t want to overload the interviewer with the technical steps that you took. You can always follow up a short response, with “I can provide more details on that if you like.” It is good to show that you understand the bigger picture of a machine learning model you built, so describe the high-level problem first and then how you solved it. When asked about using a specific tool, such as “Have you used SQL?”, respond with the project context when possible. An example response would be “ Yes, I use SQL at the beginning of every new project to query the database, then I pull that data into Python and proceed from there. I did this for the last project I described, flagging fraudulent credit card transactions.” Having a concrete example helps the interviewer remember and confirm that you have this particular skill.

Towards the end of this call, the interviewer will likely ask if you have any questions. Take advantage of this opportunity to learn more about the role and the company. For example:

* What is the size and structure of the data science team?
* Why has this role come open, is it a new position?
* Who is the hiring manager?

Building rapport with a company begins with the phone screen, make sure to be gracious and grateful for the interviewer’s time and send a follow up thank you email later that day or the next morning.

# ## Technical screen

While all data scientist jobs have technical screens, the format varies. Most conduct a take-home data challenge, some conduct an additional technical screening interview before the take-home challenge, and most complete a follow up technical interview after the take-home data challenge is returned by the candidate. Some roles require a coding challenge in lieu of a data take-home challenge, and occasionally you are asked to complete both a take-home data challenge and a coding interview.

In general, the company provides a take-home data challenge involving an internal data set or a data set from Kaggle.com. Most of the time, this comes to you by email as a zip file with a short description of the problem and a CSV file. You generally have 1 week to complete the challenge and are asked to return your analysis as a Jupyter Notebook or as a project write up in PDF.

The take-home data challenge is used to evaluate technical skills such as Python programming and data science knowledge such as performing the appropriate data wrangling steps and choosing the correct type of machine learning model for a given project. The project should also demonstrate your ability to stay organized and focused when answering 1-3 specific questions as well as 1-2 open-ended questions. Every company has a slightly different approach to conducting these, but often they are a combination of open-ended and specific questions.

Below are two examples of take-home data challenge instructions. The first example is from a well-developed data science team in the healthcare domain. The second example is from an insurance company also with a large data science department. Notice how the first example is very open-ended and requires the candidate to develop a path for themselves in order to stay in line with the time constraint. The second example has more specific steps outlined but also expects model performance to be the driving force in determining to get to the next stage of the interview.

## ### Example take-home challenge 1

*```*

*Expectations of the exercise:*

*The goal of this exercise is to demonstrate your capability with data science, machine learning tools, and Python programming. We expect this will take somewhere between 10 and 20 hours to complete, so try to time-box it to that level of effort over the next week or so.*

*We would like you to process the data set below and tell us what important findings you get from this data. What you find and report (classifications, correlations, features, causality, etc) is up to you. We are looking for you to demonstrate your level of knowledge in data science, your effectiveness with ML tools, and your skill with the Python programming language. It is expected you will use Python, but beyond that, you are free to use any ML toolkits or libraries you prefer.*

*Here is the link to the data:* [*https://www.kaggle.com/new-york-state/nys-oasas-medicaid-trend-recipient-summary-profile*](https://www.kaggle.com/new-york-state/nys-oasas-medicaid-trend-recipient-summary-profile)

*We expect your analysis about the data will be meaningful, but we do not expect it to be exhaustive. We recognize that you could spend limitless time on this, but please try to stay in the time box. We want to see what you can quickly accomplish, with the tools available to you, and we want to see that your output is illustrative of your skills in data science, and that it will show us you can meaningfully contribute to solving the kinds of problems that we are working on. We are not prescribing any particular format for your output or analysis, just that it focus on the substance of what you learned by analyzing the data.*

*Deliverables:*

*- a text file containing your Python code*

*- a writeup summarizing your results (~2 pages) in any standard format (Word, PDF, etc).*

*The summary should include the tools and methods you used, why you used them, and what insights you have gained. Graphs or pictures are welcome. Note that, due to the time constraint, it is not expected that your code be polished, just a few comments illustrating what you are doing will be sufficient.*

*```*

## ### Example take-home challenge 2

```

*Some notes before starting:*

*\* Read all the way through the instructions.*

*\* Models must be built using Python.*

*\* No additional data may be added or used.*

*\* While simple techniques may develop adequate models, success in this exercise typically involves feature engineering and model tuning.*

*\* Throughout your code, please use comments to document your thought process as you move through exploratory data analysis, feature engineering, model tuning, etc.*

*\* Please review your submission against the submission expectations.*

*Step 1 - Clean and prepare your data:*

*There are several entries where values have been deleted to simulate dirty data. Please clean the data with whatever method(s) you believe is best/most suitable. Note that some of the missing values are truly blank (unknown answers). Success in this exercise typically involves feature engineering and avoiding data leakage.*

*Step 2 - Build your models:*

*Please train two different machine learning/statistical models to predict the value for y. Please include comments that document choices you make (such as those for feature engineering and for model tuning).*

*Step 3 - Generate predictions:*

*Create predictions on the data in test.csv using each of your trained models. The predictions should be the class probabilities for belonging to the positive class (labeled '1').*

*Be sure to output a prediction for each of the rows in the test dataset (10K rows). Save the results of each of your models in a separate CSV file. Title the two files 'results1.csv' and 'results2.csv'. A result file should each have a single column representing the output from one model (no header label or index column is needed).*

*Step 4 - Compare your modeling approaches:*

*Please prepare a relatively short write-up comparing the pros and cons of the two modeling techniques you used (PDF preferred). Are there choices you made in the context of the exercise that might be different in a business context?*

*Step 5 - Submit your work:*

*Your submission should consist of all the code used for EDA, cleaning, prepping, and modeling (text, html, or pdf preferred), the two result files (.csv format - each containing 10,000 decimal probabilities), and your write-up comparing the pros and cons of the two modeling techniques used (text, html, or pdf preferred).*

*Your work will be scored on techniques used (appropriateness and complexity), evaluation of the two techniques compared in the write-up, model performance on the data hold out - measured by AUC, and your overall code/comments. The threshold for passing model performance is set high, expecting that model tuning and feature engineering will be used. The best score of the two models submitted will be used.*

*Please do not submit the original data back to us.*

*If you have trouble opening the attached file or accessing the data please let us know immediately. In order to reduce the size of the results you return, please only send back your classifications (do not include the original data). While you are welcome to use any resources you have available to perform this work, we expect your work to be original and your own.*

*Please send your results to this email address by the end of day next Monday.*

```

After this technical screen, you may also be asked to complete a coding test. In most cases, you will be asked to either perform a coding test or a take-home data challenge—not both.

# ## Coding test

Coding tests are most often conducted using third-party software, like [Codebunk](https://codebunk.com/) or [Coderpad](https://coderpad.io/), which allows the interviewer to join an IDE or text editor at the same time as you, and allows for video conferencing as well. You are most likely going to be asked to write scripts to solve SQL queries or write programs in Python to answer specific questions. These interviews typically last 1 hour and start with simple questions, building up to more difficult ones. You are not permitted to use Google, Stackoverflow.com, or other reference materials during this interview.

Some interviewers may help guide you to the correct solution if you’re struggling. The best approach to tough coding problems is to verbally walk through how you’re going to solve the problem and start by writing logic code. If you’re on the right track but have a bug or two, the interviewer may gently point them out to allow you to move on. However, not all interviewers are as supportive; don’t be surprised if you get no feedback from the interviewer while you solve the coding problems. It is common for them to simply move on to the next question if you can’t solve it on your own. It is also common for the interviewer to give you a time limit for each question and then move on or end the interview if you can’t solve it. These are definitely tough interviews, but being prepared by practicing questions on [Leetcode](https://leetcode.com/) or [Hacker Rank](https://www.hackerrank.com/) is the best way to prepare for these.

If you’re feeling extra confident in this department, there are hiring agencies that will push you through to onsite interviews if you can pass their coding assessments. [Triple Byte](https://triplebyte.com/a/Y8ubfM2/breakthroughds) is one example; it’s a company started by software engineers looking to improve the tech hiring process and get qualified applicants directly to top companies. Another company currently only offering software engineering roles but with the same premise of passing coding tests and skipping ahead in the hiring process is [Turing](https://developers.turing.com/signup).

After submitting the take-home challenge or coding test, you typically get a follow up technical interview to go over the project with either the hiring manager or a senior data scientist.

# ## Technical interview

The technical interview is usually 1-hour in length and is conducted via Zoom or some other web-conferencing tool that allows both video calls and screen sharing. If this is a follow up to a take-home challenge submission, be sure to review the project before the call so that it’s fresh in your mind. The interviewer will likely ask why you chose specific techniques in your analysis and expects you to explain why that method is a good choice and how it works. Most of the interview focuses on reviewing the project submission, but you should also be prepared to answer questions about data science methods that you have used in your past projects. You may be asked to walk the interviewer through your Jupyter Notebook or the document writeup that you submitted, so be prepared to share your screen and narrate the steps you took in the analysis.

Occasionally, a technical interview is conducted before a data challenge; in this case, be prepared with one or two data science project examples that you can share on GitHub or some other medium. Similar to the take-home challenge review, be prepared to share your screen and walk through a notebook or project write-up describing the process and methods that you took.

# ## On-site interview

The on-site interview is typically the final step in the interview process before an offer is given. In general, this is three to six separate interviews ranging in length from 30 minutes to 1 hour, all conducted in one day. This interview is a combination of behavioral and technical questions. You can expect interviews with several data scientists, data engineers, a project manager, and the hiring manager.

Data scientists and data engineers will likely ask technical questions, such as, “When should you apply L1 regularization?” The project manager often asks more behavioral questions and is interested in how you communicate, handle project timelines and work on a team. The hiring manager is likely to ask both technical and behavioral questions; they want to gauge the depth of your understanding of data science topics and understand how you solve problems. On-site interviews often include paired, side-by-side coding or whiteboarding questions conducted by other data scientists and engineers. Some interviews involve you and one other person, and some are group interviews with several people asking questions.

These are typically conducted in the office that you would be working in; however, they can also be conducted through video conferencing.

In your answers, make sure to provide technical details that match the audience that you’re speaking with. When describing a failed data science project to other data scientists, be sure to include the type of data wrangling, EDA, and modeling method that you applied.

Example questions you may be asked during on-site interviews:

* What is meant by the data classes being imbalanced and how do you handle this in your modeling process?
* What is the definition of variance?
* How do you deal with outliers in your data?
* Describe how Random Forest works, how is it different from decision trees?
* What are bias and variance, and how do they relate to modeling data?
* What does model overfitting mean and how can we resolve it?

The best way to get the most out of these onsite interviews is to be enthusiastic, confident, and ask questions. Below are some examples of questions to ask the interviewers, again make sure you’re asking an appropriate question given someone’s role. It can also be helpful to ask interviewers what they like about their current role or what their biggest challenge is. You can learn a lot about the working expectations of a data science team by asking them what a typical week looks like or what their biggest challenge is.

Example questions for interviewers:

* What would be my first project here? Has someone already been working on this or is it a new project?
* What is the current state of the data infrastructure? How much work needs to be done on getting the infrastructure and pipeline into shape before we start analyzing that data?
* What is the common technology stack; such as tools for version control or Mac versus PC or VMs?
* Considering that data science comprises everything from using traditional statistics in large data sets to deep learning, wherein that range is your data science team?
* What share of this role’s time is spent on data science such as model development and EDA versus meetings and operations needs?
* What is a typical timeline for a project from model development to production? 1 month, 3 months, or longer, what is the primary source of lag?
* What is the relationship between the data science team and the rest of the company?
* How will I know if I am accomplishing what is expected of me?

Finally, no matter the format or experience of an interview, always send a thank you email, later the same day or by the next morning. Thank you emails do not need to be long and drawn out, a simple “I enjoyed our conversation and thanks for your time” will do.