Get into the Right Mindset

**You ...**

* are preparing for interviews for jobs in machine learning, data science, or research
* are interested in tackling interviews less as a "test" and more like a conversation
* want to show how your journey — whether you're entry-level, a career changer, or upskilling — prepares you for the job

In this course, we will cover interviewing techniques and walkthroughs of common questions with all of the above in mind. You will first walk through the following:

* **Go inside the mind of the employer** - why is the company hiring for this role?
* **Identify a business problem** - showcase you're interested in solving the root issues in order to prevent further problems.
* **Explore potential data sources** - see how your solution depends on your unique dataset.
* **Discuss machine learning solutions** - get familiar with answering interview questions for this unique field.
* **Practice answering technical questions** - bolster your confidence for interview day by practicing interview techniques.

Identify a Business Problem

## Go inside the mind of the employer

If a company wants to hire a machine learning engineer (or data analyst or scientist), it’s clear that they are trying to solve a complex problem where traditional algorithmic solutions are hard to apply or simply do not work well enough. They are also extremely motivated to solve that problem (otherwise they wouldn’t be in this business!).

### Reflection Exercise

Before the interview, or even before you apply to a job, investigate possible answers to these questions:

#### Why is the company hiring?

Does it look like they're backfilling (ie: someone left the company or was promoted, so they need to fill in the job)? Or, are they expanding their team (therefore investing more in data science) or even starting their data team for the first time? These answers will help you see how you're needed. If the company is looking for a member of a large team, they likely need you to fill in a specific role and may want you to show expert skill in a specialization. If they're building out their data team, they may be looking for someone who's a fast learner and can help them make decisions - since they may not know what they need from a data team.

#### How will this company's data team grow?

This will give you insight on longterm opportunities for you. If they're growing rapidly, that may signal you have potential to become a team lead in the future.

#### Which business problem are they trying to solve?

##### Identify a Business Problem

Imagine yourself in the role – find out as much as possible about the company and position. Then ask yourself: What is one core problem I need to solve? The answer should excite you, and drive you to find out more about the problem, existing approaches, and recent developments in that domain. Choose a problem that is essential for the company.

For example, as a Machine Learning Engineer at Udacity, your primary responsibility could be to improve student engagement and retention. It is widely known that MOOC completion rates are abysmally low. Why do students drop out of our programs? Do they find our content too hard, or too easy? How can we improve student engagement in the classroom?

Although you may not be able to figure this out, you can think of realistic possibilities. For example, an e-vendor may want to make better recommendations based on search history. A healthcare company may want to better predict health risks. When you are in the interview, connect your past experiences to how you can fix these hypothetical issues.

### What else?

Don't forget to ask your interviewers the questions in this exercise, also! Not only do you want to know the answers, asking will showcase your interest and thoughtfulness.

NEXT

Prepare to Discuss ML Solutions

## Explore Potential Data Sources

Next, think about what data you would need to answer these questions. Some of this may be readily available, while you may have to build in additional hooks to gather certain pieces of information. Dig into the company’s infrastructure and operations - what stack do they operate on, what APIs do they have, what data are they already collecting, etc. Most companies today have a blog where they often discuss their challenges, approaches, successes, and failures. This should give you further insight into how they operate, and what products and services they might have in the pipeline.

## Prepare to Discuss Machine Learning Solutions

Now that you have a business problem to solve, you need to frame it in terms of machine learning. Given what you’re trying to achieve, and the data you think might be available, can you frame it as an ML problem?

Here are a couple questions to ask yourself:

* **What is an appropriate model to use?**
* **How would you go about training and evaluating it?** For example, the primary challenge that a lot of recommendation systems like Netflix and Amazon face is clustering, not prediction - i.e. once you are able to figure out groups of users who seem to have similar preference and behavior, it becomes a whole lot easier to recommend products that they may find useful.

This thought process will help you be prepared to talk about issues that matter to the company the most. Nobody expects you to walk into an interview and lay out a complete solution for something they’ve been working hard on for months or years! But everybody likes a candidate who shows genuine interest, motivation, and curiosity for a problem that is close to their hearts.

Depending on your interviewer and the stage of your interview, you may be asked more technical questions, but you should try to use any opportunity you get to demonstrate that you have thought about the company and role.

**Pro Tip:** When asked more open-ended questions such as “Describe a technical challenge you faced when working on a project and how you solved it,” try to pick something that aligns well with the company’s interests.

NEXT

Focus on Essential Skills

## Prepare to Demonstrate these Technical Skills

Here are the essential skills that a Machine Learning Engineer needs, as mentioned in the first video of this lesson. Within each group are topics that you should be familiar with.

**Study Tip:** Copy and paste this list into a document and save to your computer for easy referral.

#### Computer Science Fundamentals and Programming

**Topics**

* Data structures: Lists, stacks, queues, strings, hash maps, vectors, matrices, classes & objects, trees, graphs, etc.
* Algorithms: Recursion, searching, sorting, optimization, dynamic programming, etc.
* Computability and complexity: P vs. NP, NP-complete problems, big-O notation, approximate algorithms, etc.
* Computer architecture: Memory, cache, bandwidth, threads & processes, deadlocks, etc.

#### Probability and Statistics

**Topics**

* Basic probability: Conditional probability, Bayes rule, likelihood, independence, etc.
* Probabilistic models: Bayes Nets, Markov Decision Processes, Hidden Markov Models, etc.
* Statistical measures: Mean, median, mode, variance, population parameters vs. sample statistics etc.
* Proximity and error metrics: Cosine similarity, mean-squared error, Manhattan and Euclidean distance, log-loss, etc.
* Distributions and random sampling: Uniform, normal, binomial, Poisson, etc.
* Analysis methods: ANOVA, hypothesis testing, factor analysis, etc.

#### Data Modeling and Evaluation

**Topics**

* Data preprocessing: Munging/wrangling, transforming, aggregating, etc.
* Pattern recognition: Correlations, clusters, trends, outliers & anomalies, etc.
* Dimensionality reduction: Eigenvectors, Principal Component Analysis, etc.
* Prediction: Classification, regression, sequence prediction, etc.; suitable error/accuracy metrics.
* Evaluation: Training-testing split, sequential vs. randomized cross-validation, etc.

#### Applying Machine Learning Algorithms and Libraries

**Topics**

* Models: Parametric vs. nonparametric, decision tree, nearest neighbor, neural net, support vector machine, ensemble of multiple models, etc.
* Learning procedure: Linear regression, gradient descent, genetic algorithms, bagging, boosting, and other model-specific methods; regularization, hyperparameter tuning, etc.
* Tradeoffs and gotchas: Relative advantages and disadvantages, bias and variance, overfitting and underfitting, vanishing/exploding gradients, missing data, data leakage, etc.

#### Software Engineering and System Design

**Topics**

* Software interface: Library calls, REST APIs, data collection endpoints, database queries, etc.
* User interface: Capturing user inputs & application events, displaying results & visualization, etc.
* Scalability: Map-reduce, distributed processing, etc.
* Deployment: Cloud hosting, containers & instances, microservices, etc.

**Move on to the final lesson of this course to find lots of sample practice questions for each topic!**

NEXT