**How to Introduce Your Self**

**Tip:** *Do not mix your past company names, projects, etc., in the introduction. It is all about how you professionally see yourself. Your experience is visible to the interviewer in your CV.*

**Template 1**

Hello, I'm a seasoned Data Engineer specialized in Data Science and General AI with a passion for leveraging advanced technologies to extract meaningful insights and drive innovation. With over 6 years of experience, I've honed my skills in developing and implementing large language model-based applications, fine-tuning and deploying AI systems, and building robust ETL jobs. My expertise extends to working with cutting-edge frameworks like LangChain and LlamaIndex, ensuring seamless integration and deployment. I thrive in collaborative environments, bringing a strong knowledge of API integration, ML Ops, and data management to the table. My goal is to contribute to the development of innovative products for data analysis and visualization, like those at LexisNexis, and to push the boundaries of what's possible in AI and machine learning.

**Template 2**

I'm an experienced Data Engineer with a deep-rooted passion for Data Science and General AI. Over the past 6 years, I've dedicated myself to mastering the intricacies of data engineering, focusing on the development and deployment of sophisticated AI systems. My expertise lies in the seamless integration of large language models and the implementation of robust ETL pipelines, ensuring data integrity and accessibility. I'm well-versed in cutting-edge technologies like LangChain and LlamaIndex, which have been instrumental in my work on API integration and ML Ops. My approach is collaborative and results-driven, aiming to enhance product capabilities in data analysis and visualization. I'm excited about the opportunity to bring my skills to a forward-thinking company like LexisNexis, where I can contribute to pioneering advancements in AI and machine learning."

Explaining a Project

Use STAR format (Situation Task, Action, and Result).

**DO NOT USE THE TITLE SITUATION TASK ACTION IN YOUR INTERVIEW RESPONSE.**

**Template**

**Situation:** Faced with the need to enhance sales-related inquiries, a project was initiated to develop a responsive and context-aware chatbot.

**Task:** The objective was to implement a chatbot using the GPT-3.5 turbo architecture that could handle dynamic SQL queries and provide accurate responses to both database-related and general questions.

**Action:** Collaborating closely with the business, I tailored prompts for the chatbot and engineered its core functionality in Python. This included dynamic SQL query generation and data retrieval from the Snowflake Database, utilizing the Retrieval Augmented generation optimization technique. I employed the Langchain SQL agent method for database-related queries and leveraged the LLM for general responses. Additionally, I conducted Exploratory Data Analysis on the tables, created limited views to reduce ambiguity, and pre-processed data by performing calculations and updating resultant values.

**Result:** The chatbot's few-shot learning capabilities were enhanced, requiring minimal training data for accurate functioning. Through Prompt Engineering and precision-focused query optimization techniques, such as index optimization, vector DB, and database statistics refinement, the chatbot's response accuracy was significantly improved.

**Use the same for specific questions like the one below.**

**Tip:***It is a good practice to reflect on your understanding of questions in brief or ask clarifying questions.*

Question: How do we handle imbalanced data classification problems in machine learning?

Template Answer:

**Situation:** In a project aimed at predicting customer churn for a telecom company, we encountered an imbalanced dataset where the number of customers who did not churn was significantly higher than those who did.

**Task:** The challenge was to develop a machine learning model that could accurately predict churn despite the data imbalance, ensuring that the minority class (customers who churn) was not overlooked.

**Action:** To tackle this, we employed several strategies:

1. **Resampling the Data:** We balanced the dataset by oversampling the minority class and undersampling the majority class to ensure equal representation.
2. **Using Appropriate Metrics:** Instead of accuracy, we focused on precision, recall, and the F1 score as our evaluation metrics, which are more indicative of true performance in imbalanced datasets.
3. **Algorithmic Adjustments:** We utilized algorithms like Decision Trees and Random Forests that are less sensitive to imbalanced data.
4. **Penalty Weights:** We assigned higher penalty weights to the minority class during model training to make the model pay more attention to those cases.
5. **Anomaly Detection:** We also experimented with anomaly detection techniques, as we recognized that churn could be viewed as an anomaly.

**Result:** These actions led to the development of a robust model that improved the F1 score by 25%, enhancing the company's ability to identify and address potential churn, ultimately leading to a more proactive customer retention strategy.

Coding Round

Tip:

1. Always ask clarifying questions and reflect on the question
2. Start with thinking loud and discuss the approach (Don’t be silent at all and write code)
3. Focus pseudo-code first
4. If possible, articulate time complexity (In terms of BigO Notation)

I am curating some of the worked-out answers from previous questions here - <https://github.com/jaganadhg/backtobasics/tree/main/leetcode>

I will add more.