The Deciphering Big Data course provided hands-on experience in handling big data, database design, automation, and data security, and each skill was contextualized with tools like Python, and SQL.

The learning begin with an exploration of big data architectures, focusing on handling vast datasets using both batch and real-time processing methods. Artefact highlights the Lambda and Kappa architectures and reflects on the strengths and complexities of each approach in managing data latency and accuracy. This foundational knowledge is complemented by exercises in data collection through web scraping, for which Python's BeautifulSoup and Requests libraries were utilised to gather and parse data from online sources and structuring output files in JSON format, emphasizing the real-world applicability of these techniques in data analysis.

In data cleaning, we learnt about identifying and rectifying data quality issues, following best practices from resources like Data Wrangling with Python. We learnt the importance of ensuring data reliability and consistency. Through artifacts we normalised data tables, and worked on logical and physical database design that ensured data integrity and supports complex queries.

The course also introduced automation and scaling methods. Firstly, each task must be clarified, including the start time, time limits, required inputs, expected outputs, and criteria for success. In case of failure contingency plan is needed, and outline what should steps should be post-completion. The steps for automation include: Break down the problem into manageable parts. Define inputs, processes, and success criteria for each part. Locate necessary resources and schedule tasks. Develop and test code with sample data. Clean and document the code. Implement logging to track errors and successful completions. Submit, test, and refine code as needed. Replace manual steps with automation. Monitor logs for errors and make adjustments. Establish a regular log-checking schedule. Common issues to watch include database errors, script bugs, timeouts, edge cases, and hardware limitations. Example of automation: Parallel processing, distributed processing, simple automation, large scale automation, automation monitoring. Learning parallel and distributed processing was particularly impactful, and learning the potential for these techniques to boost efficiency in data handling tasks. Security principles, around API and data protection, were another critical area of growth. With rising concerns around data privacy, the practical knowledge of encryption, authentication, and secure coding invaluable for managing sensitive data are necessity for a data scientist.

Skills Matrices are valuable tools for understanding team skills and identifying growth opportunities. Self-assessment is key, but it's important to validate those assessments and allow for alignment to happen over time.

Skill Category	Skill Level (1-5)	Desire to Improve (1-5)	Current Competence (Beginner, Intermediate, Advanced)	Actions/Resources for Development	
Data Engineering					
Programming and Scripting					
SQL					
Python					
Database Design					
Data Integration & APIs					
Data Governance & Security					
Automation					
Transferable Skills					
Time management					
Commercial Awareness					
Critical thinking and analysis					
Decision-making					
Problem-solving					
Initiative					
Entrepreneurial					
Communication and Literacy skills					

Numeracy					
IT and Digital					
Interpersonal					
Critical Reflection					
Research					