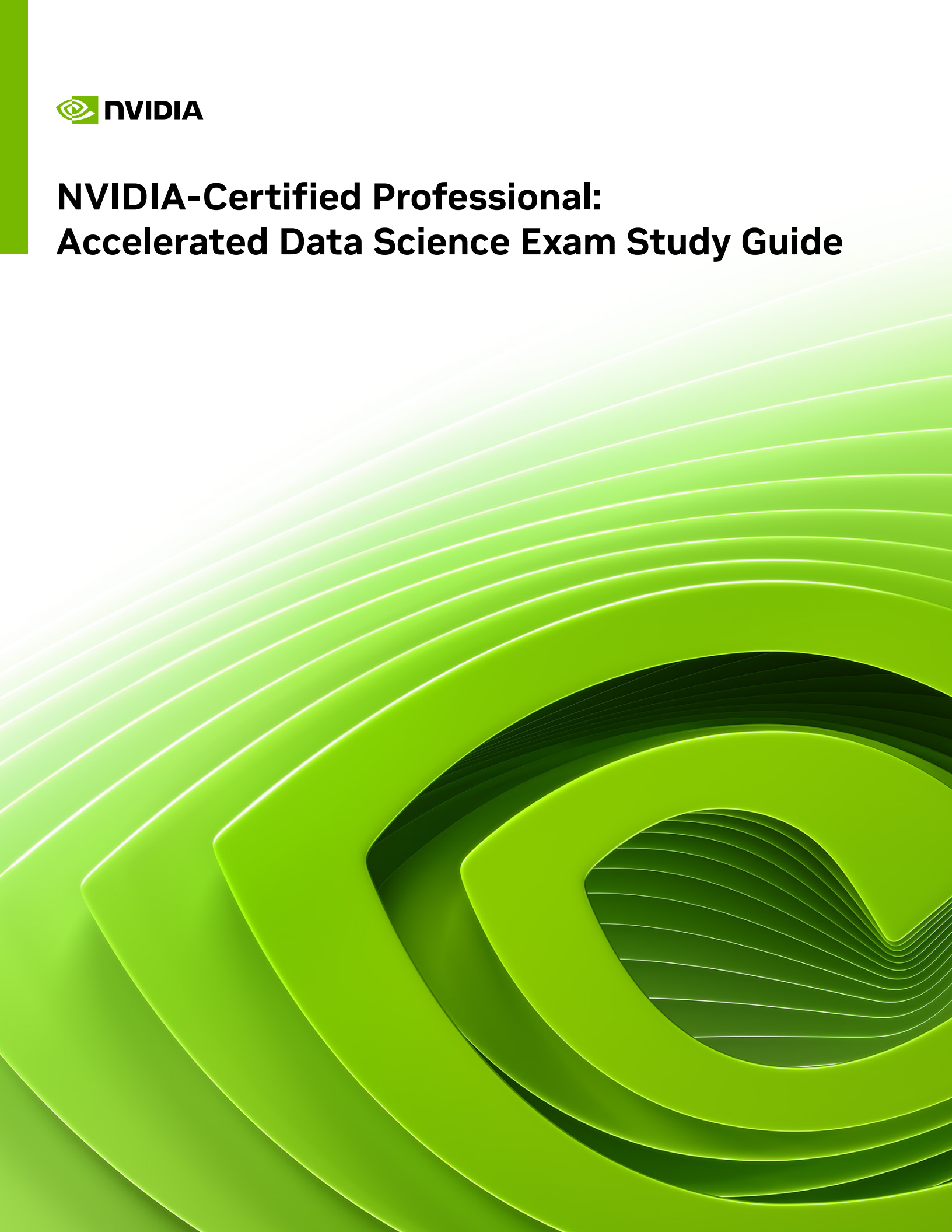
NVIDIA-Certified Professional: Accelerated Data Science Exam Study Guide



NVIDIA-Certified Professional: Accelerated Data Science Exam Study Guide

Contents

This study guide provides an overview of each topic covered on the NVIDIA Accelerated Data Science certification exam, recommended training, and suggested reading to prepare for the exam.

Information about NVIDIA certifications can be found here.

Job Description

An accelerated data science professional leverages cutting-edge GPU-accelerated technologies to optimize data science workflows, from extract, transform, and load (ETL) processes to machine learning model deployment. The ideal candidate will have a strong background in data science, machine learning, and GPU computing, with a passion for pushing the boundaries of what's possible in data analysis and modeling. This candidate is at the forefront of implementing high-performance data solutions, utilizing advanced tools and techniques to significantly enhance the speed and efficiency of data processing, analysis, and model training pipelines.

Job Responsibilities

1. Accelerated Data Science: Use advanced tools and techniques to speed up the data science process, focusing on optimizing performance through GPU acceleration.
2. GPU Acceleration: Utilize GPU computing to significantly reduce the time required for data processing, analysis, and model training.
3. Extract, Transform, Load (ETL): Efficiently manage and transform large datasets, preparing them for analysis using accelerated ETL processes.
4. Data Analysis: Conduct comprehensive data analysis to extract meaningful insights, using accelerated workflows to manage large-scale data.
5. Feature Engineering: Create and optimize features to improve model performance, leveraging GPU-accelerated methods to streamline the process.
6. Data Modeling: Develop and fine-tune data models, employing advanced GPU-accelerated algorithms to enhance accuracy and speed.
7. Graph Analytics: Create and analyze graph data using GPU-accelerated tools. Implement and optimize graph-based data structures and algorithms for data relationships and network analysis.
8. Data Preprocessing: Perform data cleaning, transformation, and normalization using accelerated techniques to prepare data for analysis and modeling.
9. MLOps: Integrate machine learning models into production environments, ensuring scalability and efficiency through accelerated pipelines.
10. GPU-Accelerated Machine Learning: Design, train, and evaluate machine learning models using GPU-accelerated libraries to reduce training time and improve performance.
11. Data Visualization: Create high-quality visualizations to communicate data-driven insights effectively, utilizing accelerated tools for large datasets.
12. Time-Series Analysis: Apply advanced time-series forecasting methods, leveraging accelerated techniques for faster and more accurate predictions.
13. GPU-Accelerated Data Manipulation: Efficiently manipulate large datasets using GPU-accelerated libraries, ensuring optimal performance in data processing tasks. Utilize cuDF for efficient data manipulation, feature extraction, and transformation tasks.
14. Data Cleansing: Perform thorough data cleansing to ensure high-quality data input, using accelerated methods to manage large and complex datasets.

Recommended Qualifications and Experience

1. Two to three years of hands-on experience in accelerated data science
2. A strong foundation in machine learning and GPU-accelerated computing
3. Experience in GPU-based optimization strategies and accelerated data manipulation techniques
4. A deep understanding of end-to-end data science workflows, from data preparation and cleansing to model development and deployment, with a focus on leveraging GPU acceleration for enhanced performance and efficiency

Certification Topics and References

Data Analysis: Exam Weight 14%

Conducting time-series analysis to detect anomalies, visualize trends, and perform exploratory data analysis (EDA) for insights. Analyzing graph data using tools like cuGraph and determining when data qualifies as big data to apply the appropriate acceleration methods for efficient processing. These steps help in understanding and optimizing data for better decision-making.

1.1	Detect anomalies in a time-series dataset.
1.2	Conduct time-series analysis.
1.3	Create and analyze graph data using something like cuGraph.
1.4	Identify how much data is big data (or when to use which acceleration method).
1.5	Perform exploratory data analysis (EDA).
1.6	Visualize time-series data.

Recommended Training (Optional)

- Fundamentals of Accelerated Data Science or Accelerating End-to-End Data Science Workflows
- Enhancing Data Science Outcomes With Efficient Workflow

Suggested Readings

- Supercharge Graph Analytics at Scale With GPU-CPU Fusion for 100x Performance
- Exploratory Data Analysis in Python
- Anomaly Detection for Time-Series Data: Anomaly Types
- NYC Taxi Data Using dask_cudf
- Beginner's Guide to GPU-Accelerated Graph Analytics in Python
- NetworkX Tutorial
- Big Data: A Revolution That Will Transform How We Live, Work, and Think by Viktor Mayer-Schönberger and Kenneth Cukier
- Anomaly Detection: A Survey by Varun Chandola, Arindam Banerjee, and Vipin Kumar
- Forecasting: Principles and Practice by Rob J. Hyndman and George Athanasopoulos
- Iglewicz, B., & Hoaglin, D. C. (1993). How to Detect and Handle Outliers. Sage Publications.
- Zimek, A., & Schubert, E. (2017). A Survey on Evaluation Methods for Anomaly Detection. ACM Computing Surveys, 50(3), 1-34.

Data Manipulation and Software Literacy Exam Weight 19%

Designing and implementing efficient ETL workflows using accelerated processes, data caching, and distributed data processing frameworks to manage big data. Data parallelism with Dask for multi-GPU scaling is utilized, alongside profiling deep learning models with tools like DLProf to optimize performance. Additionally, tasks include selecting optimal data processing libraries for different dataset sizes and performing GPU-accelerated data transformation and standardization using cuDF. Effective GPU memory management and scaling inference for large datasets across single-node and multi-node systems are also key components.

2.1 Design and implement ETL workflows using accelerated ETL processes.
2.2 Implement data caching to reduce shuffle.
2.3 Use distributed data processing frameworks to process big data.
2.4 Implement data parallelism using Dask for multi-GPU scaling.
2.5 Profile deep learning models using tools such as DLProf.
2.6 Determine the optimal data processing libraries to use for varying dataset sizes.
2.7 Transform and standardize data using cuDF.
2.8 Manage and allocate GPU memory effectively to improve performance, utilizing strategies to optimize memory.
2.9 Scale inference for large datasets in both single-node and multi-node systems.

Recommended Training (Optional)

- Fundamentals of Accelerated Data Science or Accelerating End-to-End Data Science Workflows
- Enhancing Data Science Outcomes With Efficient Workflow

Suggested Readings

- Accelerating ETL on KubeFlow With RAPIDS™
- FAQ and Known Issues
- Dask cuDF Best Practices
- NVIDIA Triton™ Inference Server
- Categorical Features in XGBoost Without Manual Encoding
- User Guide
- Unlocking Multi-GPU Model Training with Dask XGBoost
- RAPIDS on Databricks: A Guide to GPU-Accelerated Data Processing
- ETL Principles
- Accelerating Apache Spark 3.0 With GPUs and RAPIDS

Data Preparation: Exam Weight 17%

Cleansing and preprocessing data using cuDF and pandas, followed by transforming and standardizing it to maintain consistency across features. Synthetic data is generated using cuDF/RAPIDS to enhance datasets, while relevant datasets are identified and acquired. Additionally, data processing pipelines are monitored to detect bottlenecks, ensuring efficient organization, processing, and storage of the datasets for further analysis.

3.1	Perform data cleansing and preprocessing using cuDF and pandas.
3.2	Transform and standardize data.
3.3	Standardize data as needed to ensure uniformity across features.
3.4	Identify and acquire datasets.
3.5	Monitor data processing pipelines to recognize bottlenecks.
3.6	Process, organize, and store datasets.

Recommended Training (Optional)

- Fundamentals of Accelerated Data Science or Accelerating End-to-End Data Science Workflows
- Enhancing Data Science Outcomes With Efficient Workflow

Suggested Reading List

- Synthetic Data for Deep Learning
- NVIDIA NeMo™ Curator for Developers
- Optimizing Access to Parquet Data With fsspec
- Curating Non-English Datasets for LLM Training With NVIDIA NeMo Curator
- Working With JSON Data
- Faster Resampling With Imbalanced-Learn and cuML
- Machine Learning Frameworks Interoperability, Part 2: Data Loading and Data Transfer Bottlenecks
- Data Science: Understanding Feature Scaling in Machine Learning
- Encoding and Compression Guide for Parquet String Data Using RAPIDS
- Preprocessing Data
- The Importance of Feature Scaling
- Dealing With Outliers Using Three Robust Linear Regression Models
- Data Loading
- Best Practices for Monitoring Data Pipeline Performance

GPU and Cloud Computing: Exam Weight 16%

Leveraging GPU acceleration to optimize data science workflows, including analyzing graph data using cuGraph and improving performance across processes. They involve following the CRISP-DM methodology, managing software dependencies with frameworks like Docker or Conda, and selecting optimal data types for features. Additionally, tasks include benchmarking, comparing the performance of various frameworks, and understanding when and how to scale data processing pipelines to the cloud using platforms such as AWS, GCP, or Databricks.

4.1 Analyze graph data using GPU-accelerated tools like cuGraph.
4.2 Optimize performance of the data science process through GPU acceleration.
4.3 Describe, follow, and execute the CRISP-DM process.
4.4 Utilize dependency management frameworks, such as Docker and Conda to manage software-versioning conflicts.
4.5 Determine the optimal data type choice for each feature.
4.6 Compare frameworks' performance by designing and implementing a benchmark.

Recommended Training (Optional)

- Fundamentals of Accelerated Data Science or Accelerating End-to-End Data Science Workflows
- Enhancing Data Science Outcomes With Efficient Workflow

Suggested Reading List

- NetworkX Introduces Zero-Code-Change Acceleration Using NVIDIA cuGraph
- Managing Environments
- What Is Docker?
- Installing Conda
- v1.1 Results—Inference: Data Center
- LLM Inference Performance Engineering: Best Practices
- cuGraph Introduction
- Dealing With Small Files Issues on S3: A Guide to Compaction
- Use Cases of Docker: Rollback and Version Control
- RAPIDS 24.04 Release
- Remote Direct-Memory Access (RDMA)
- Data Tiering
- Autoscaling Groups of Instances
- CPU vs. GPU for Machine Learning

Machine Learning: Exam Weight 15%

Feature engineering by efficiently splitting and standardizing data to ensure uniformity across features. In GPU-accelerated machine learning, tasks include determining when to apply acceleration techniques for big data, rapidly experimenting to balance model accuracy and performance, and optimizing hyperparameters. Training machine learning models in both single-GPU and multi-GPU scenarios is essential, alongside using techniques like batching and mixed precision for GPU memory optimization. Additionally, deep learning model training involves profiling models with tools like DLProf to enhance performance and efficiency.

5.1 Split data, choosing the best split type and method.
5.2 Standardize data as needed to ensure uniformity across features.
5.3 Identify how much data is big data (or when to use which acceleration method).
5.4 Perform rapid experimentation to find the balance between model accuracy and inference performance.
5.5 Optimize hyperparameters of machine learning models.
5.6 Train machine learning models for both single-GPU and multi-GPU scenarios.
5.7 Use GPU memory optimization techniques, such as batching and mixed precision, to train machine learning models.
5.8 Profile deep learning models using tools such as DLProf.

Recommended Training (Optional)

- Fundamentals of Accelerated Data Science or Accelerating End-to-End Data Science Workflows
- Enhancing Data Science Outcomes With Efficient Workflow

Suggested Reading List

- What Are the Benefits and Drawbacks of Batch Processing in Machine Learning?
- Performance Optimization
- Model Parallelism
- Parallel Training Methods for AI Models: Unlocking Efficiency and Performance
- Stratified Sampling: You May Have Been Splitting Your Dataset All Wrong
- Considerations for Hyperparameter Tuning
- Feature Scaling in Machine Learning
- Overfitting vs. Underfitting in Machine Learning
- AI Performance Tuning: Answer, Benchmark, Test
- Parallelism
- PyTorch Multi-GPU: 4 Techniques Explained
- Profiling and Optimizing Deep Neural Networks With DLProf and PyProf
- Feature Scaling
- Mixed-Precision Training
- Methods and Tools for Efficient Training on a Single GPU
- Hyperparameter Tuning Guide
- The Importance of Feature Scaling
- GPU Data Analytics: Accelerating Big Data Processing
- AI Inference Performance

MLOps: Exam Weight 19%

Execution, optimization, and deployment of machine learning models within MLOps, including determining optimal data types and assessing memory requirements. They involve selecting the best accelerator for workloads, implementing parallel processing, and optimizing GPU-accelerated workflows through benchmarking. Tasks also cover planning data loading for inference pipelines, deploying models on Triton, and ensuring data security and compliance with encryption. Additionally, continuous integration and deployment (CI/CD) pipelines are implemented to streamline the process, while caching is used to reduce data loading times and optimize machine learning pipelines.

6.1 Determine the optimal data type choice for each feature.
6.2 Assess and verify the size in memory of a dataset.
6.3 Compare the required memory with the available memory on a device.
6.4 Perform benchmarking and optimize different GPU-accelerated workflows.
6.5 Deploy and monitor models in production.

Recommended Training (Optional)

- Fundamentals of Accelerated Data Science or Accelerating End-to-End Data Science Workflows
- Enhancing Data Science Outcomes With Efficient Workflow

Suggested Reading List

- Machine Learning Model Monitoring: Best Practices
- Deep Learning Specialization
- Triton Response Cache
- Protecting Sensitive Data and AI Models With Confidential Computing
- Improving GPU Memory Oversubscription Performance
- NVIDIA Triton Inference Server
- Data Types
- 10 Minutes to cuDF
- Using Multiple GPUs and Multiple Nodes
- Lustre File System
- Parallelism
- Mastering LLM Techniques: Inference Optimization
- NVIDIA TensorRT™
- vLLM
- A Guide to Quantization in LLMs
- Batch Sizes
- GPU Memory Essentials for AI Performance
- Even Faster and More Scalable UMAP on the GPU With RAPIDS cuML
- Estimator Intro
- Boosting Data Ingest Throughput With NVIDIA GPUDirect® Storage and RAPIDS cuDF
- Optimization
- Deploying Machine Learning Models on NVIDIA Triton Inference Server

Questions?

Contact us here.

NVIDIA