# Assessment Rubrics

Bootcamp Final Assignments · Amsterdam Data Academy · NLQF Level 6

This document describes the assessment criteria for all ADA bootcamp final assignments. Each criterion is scored 0–3 and weighted to produce a raw score (max 300). Grading is conducted independently by two qualified assessors; scores are averaged to determine the final result. This document is published in connection with the ADA Examination Regulations.

## Grading scale (max 300 points — scale: 0 = Insufficient, 1 = Needs Improvement, 2 = Sufficient, 3 = Good)

| Score | Result | Meaning |
|---|---|---|
| **255 – 300** | **Excellent** | Outstanding work demonstrating mastery. |
| **210 – 254** | **Good** | Exceeds requirements with clear quality. |
| 165 – 209 | Sufficient | Meets requirements at the expected level. |
| 0 – 164 | Insufficient | Does not meet minimum requirements. |

### A51 · Final Assignment – Data Analytics Bootcamp

Block: Bootcamp · Examiner: Lucia / Fatemeh

Assessment: Slurpini case — Vivino dataset (Excel or Power BI)

| Criterion | Wt. | Good (3) | Sufficient (2) | Needs Improvement (1) | Insufficient (0) |
|---|---|---|---|---|---|
| **Data Cleaning & Preparation** | **15%** | **Documented cleaning pipeline; enrichment present; fully reproducible.** | Dataset correctly cleaned: duplicates removed, types corrected, missing values handled. | Basic cleaning performed but undocumented; some issues remain. | Dataset barely cleaned; NaN values, duplicates and type errors present. |
| **Descriptive Metrics & Insight Quality** | **20%** | **Deep multi-layered analysis with statistically grounded insights, prioritised by business impact.** | Multiple well-structured metrics with actionable insights and cross-dimensional comparison. | Clear descriptive metrics with logical segmentation (by region, grape variety). | Metrics incorrect or only superficially descriptive without segmentation. |
| **Dashboard Design & Visualisation** | **25%** | **Executive-level dashboard: storytelling structure, KPI indicators, colour-coded exceptions.** | Professional dashboard with strong visual clarity, hierarchy and consistency. | Structured dashboard with readable charts and appropriate labels. | Charts unclear, poorly formatted or no logical dashboard structure. |
| **Use of Analytical Tools (Pivot / DAX / Formulas)** | **20%** | **Power Query documented; DAX measures for complex KPIs; reusable templates.** | Advanced use of pivots, filters and calculated fields; slicers correctly linked. | Correct use of pivot tables and calculated measures; basic filters present. | Incorrect or minimal tool usage; manual calculations where formulas required. |
| **Business Interpretation & Recommendations** | **20%** | **Strategic recommendations with ROI indication and risk considerations for Slurpini.** | Prioritised recommendations aligned with Slurpini wine producer selection strategy. | Recommendation linked to data-driven insights. | Descriptive only; no recommendations provided. |

## A52 · Final Assignment – Data Science & Python Bootcamp

Block: Bootcamp · Examiner: Akos / Ali

Assessment: Slurpini case — Vivino dataset · Python (pandas / seaborn / matplotlib)

| Criterion | Wt. | Good (3) | Sufficient (2) | Needs Improvement (1) | Insufficient (0) |
|---|---|---|---|---|---|
| **Data Cleaning & Preparation** | **20%** | **Fully reproducible cleaning pipeline; enrichment; data quality control log present.** | Systematic cleaning with documented assumptions; grape variety and region correctly derived. | Data correctly cleaned and structured; choices briefly justified. | Dataset poorly cleaned; fundamental errors in analysis due to dirty data. |
| **Custom Value-for-Money Metric** | **25%** | **Mathematically rigorous metric; multiple variants compared; sensitivity analysis; reusable function.** | Well-justified composite metric with mathematical reasoning and normalisation. | Working custom metric combining rating and price. | No original metric; only standard averages used. |
| **Analytical Depth** | **20%** | **Statistical foundation: correlation analysis, clustering, distribution analysis; surprising insights named.** | In-depth analysis: correlations, outliers, time trends and cross-dimensional comparison. | Segmented multi-variable analysis by region and grape variety. | Only averages shown; no segmentation or pattern analysis. |
| **Code Quality & Reproducibility** | **20%** | **Production-quality code: classes, type hints, requirements.txt, reproducible with one run command.** | Modular, reproducible workflow; functions reusable; docstrings present. | Readable and logical code; notebook reproducible with random_state. | Unstructured script; notebook does not run top-to-bottom. |
| **Business Recommendation** | **15%** | **Strategic prioritisation matrix: top 5 producers with score, region, grape variety and justification.** | Clear prioritisation aligned with Slurpini efficiency goals. | Recommendation supported by analysis. | No clear recommendation; conclusions vague. |

## A53 · Final Assignment – Data Engineering Bootcamp

Block: Bootcamp · Examiner: Marcel / Ali

Assessment: Slurpini case — Python crawler + ETL pipeline + analysis

| Criterion | Wt. | Good (3) | Sufficient (2) | Needs Improvement (1) | Insufficient (0) |
|---|---|---|---|---|---|
| **Crawler Extension** | **25%** | **Production-grade crawler: async requests, retry logic, incremental updates, validation, logging.** | Robust crawler with error handling, pagination and rate limiting; 5+ new attributes. | Functional extension of crawler with at least 3 new relevant attributes. | Crawler incomplete; additional data points not or incorrectly collected. |
| **Pipeline Automation** | **20%** | **Fully automated pipeline: scheduler/orchestrator, idempotent execution, .env config, health monitoring.** | Automated modular pipeline with clear phase separation. | Structured pipeline script that chains steps together. | Manual steps required; no structured pipeline. |
| **Data Structuring & Transformation** | **20%** | **Data quality framework: schema validation, data lineage documented, versioned dataset.** | Normalised dataset with transformations (grape variety derived, region normalised). | Cleaned and structured dataset with correct types. | Raw data used directly without structuring. |
| **Reproducibility & Documentation** | **20%** | **One-click reproducible: Docker, requirements.txt, seed data, end-to-end test; fully documented.** | Fully reproducible and documented; README with setup and run instructions. | Clear documentation; pipeline reproducible with instructions. | Pipeline not traceable; workflow undocumented. |
| **Technical Stability** | **15%** | **Production-grade stability: rate limit handling, HTTP retry, empty response handling, logging.** | Robust under varying scenarios; basic errors handled. | Stable execution under normal conditions. | Script unstable; crashes on small variations in data. |

## A54 · Final Assignment – Applied AI Bootcamp

Block: Bootcamp · Examiner: Sam / Joost

Assessment: AI Sommelier Challenge — n8n or Make.com workflow + demo + documentation

| Criterion | Wt. | Good (3) | Sufficient (2) | Needs Improvement (1) | Insufficient (0) |
|---|---|---|---|---|---|
| **Sommelier Prompt Engineering** | **20%** | **Production-grade prompt: modular template, system/user split, tested on 5+ dishes; complete iteration log.** | Iteratively refined prompt with edge case handling and documented improvements per version. | Structured prompt with clear role, wine portfolio context and output format. | Prompt vague, generic or not grounded in Slurpini wine data; inconsistent output. |
| **n8n / Make.com Workflow Implementation** | **25%** | **Production-ready workflow: error handling, logging, rate limit handling, clean node layout.** | Modular workflow with clear decision logic and minimum 5 connected nodes/modules. | Functional workflow accepting dish input, calling LLM API and returning recommendation. | Workflow incomplete or non-functional. |
| **Wine Pairing Quality** | **20%** | **Sommelier-level pairings: terroir knowledge, occasion context, serving tips, justified from data.** | Nuanced recommendations with flavour reasoning (acidity, tannins, weight) from tasting notes. | Correct recommendations with basic pairing explanation per dish. | Recommendations generic or incorrect for the dish type. |
| **Edge Case Handling** | **15%** | **All 3 edge cases robustly handled; fallback for unknown dishes; exception logic in prompt.** | All 3 required edge cases (vegetarian, spicy, dessert) handled with explicit reasoning. | At least 1 edge case correctly handled and documented. | No edge cases tested or handled. |
| **Responsible AI Reflection** | **10%** | **Structured governance analysis: bias sources, auditability, accountability, production recommendations.** | Critical analysis of governance, bias risks and production considerations. | AI risks (hallucination, dataset bias) identified with basic mitigation. | Minimal risk awareness. |
| **Documentation & Demo** | **10%** | **Board-ready: architecture diagram, prompt versions, test results, next steps; demo convinces jury.** | Professional documentation with 5+ test cases and results; demo runs live. | Clear workflow explanation with screenshot walkthrough; demo runs live. | Poor documentation; demo does not work. |

## A55 · Final Assignment – AI Engineering Bootcamp

Block: Bootcamp · Examiner: Ali / Rohit

Assessment: AI Sommelier Challenge — Python + vector DB + LLM + Streamlit/FastAPI

| Criterion | Wt. | Good (3) | Sufficient (2) | Needs Improvement (1) | Insufficient (0) |
|---|---|---|---|---|---|
| **RAG Pipeline Architecture** | **20%** | **Production-grade: layered design, configurable parameters, clear component interfaces; architecture diagram.** | Well-designed modular pipeline with justified choices for chunking, embedding and retrieval. | Functional RAG pipeline: ingestion, embedding, vector store and retrieval correctly implemented. | No structured RAG pipeline; components missing or not working together. |
| **Embedding & Vector Database** | **20%** | **Production config: hybrid search, HNSW index, metadata schema documented; embedding benchmarked.** | Optimised indexing with metadata filtering (region, grape, price); top-k tuned for pairing quality. | Dataset correctly embedded and stored; basic similarity search functional. | Vector store not functioning or insufficiently indexed. |
| **Grounded LLM Generation** | **20%** | **Production-grade grounding: source citation with chunk IDs; anti-hallucination instructions; re-ranking.** | Grounded recommendations with pairing reasoning from tasting notes; source attribution present. | Retrieved wine chunks correctly injected into prompt; wine name and region cited in output. | LLM hallucinates; output not based on Slurpini data. |
| **Wine Pairing Quality** | **15%** | **Sommelier-level; all 3 edge cases robustly handled; qualitative evaluation shows improvement over baseline.** | Nuanced pairings including all 3 edge cases; reasoning traceable to retrieved data. | Correct recommendations with pairing explanation for standard test dishes. | Recommendations incorrect or not grounded in Slurpini data. |
| **Evaluation & Technical Report** | **15%** | **RAGAS framework or equivalent; latency benchmark; scalability analysis; cost-per-query; production roadmap.** | Systematic evaluation: retrieval precision, faithfulness; failure modes identified. | Basic evaluation of retrieval quality with 5 test queries; report present. | No evaluation; technical report absent. |
| **Architecture Documentation & Demo** | **10%** | **Production-quality: C4 diagram, swagger spec, Docker, README with deployment instructions; demo convinces.** | Professional documentation: README, setup instructions, 5 example queries with output. | Architecture diagram and working Streamlit/FastAPI demo present. | No architecture diagram or working interface. |

Questions about your assessment? Contact support@amsterdamdataacademy.com · Full examination regulations: amsterdamdataacademy.com/examination-regulations