

University of Tehran – ECE Department

# Data Science Comprehensive Final Assessment

## (Extended Edition)

Course Staff (Spring 2025) – Professional Assignment Pack

Version 1.0

**Course Context:** This extended final integrates the full UT-ECE Data Science track seen in Spring 2024/2025 repositories, from Python and scientific studies to SQL engineering, ML, deep learning, NLP, and LLM agents.

## 1. Assessment Overview

**Primary dataset:** GlobalTechTalent\_50k.csv (50,000 rows).

**Primary target:** Migration\_Status (binary).

**Additional allowed datasets:**

- Course assignment datasets from the UT-ECE repos (where relevant).
- Public benchmark datasets with proper citation.

**Required submission artifacts:**

1. One reproducible notebook with clear sectioning (Q1–Q20 + Capstone).
2. One PDF report (max 20 pages excluding appendix).
3. One code package with scripts/modules and dependency file.
4. One presentation deck (10–15 slides).
5. One ethics/fairness memo (1–2 pages).

**Reproducibility requirements:**

- Fix random seeds where applicable.
- State train/validation/test splitting strategy.
- Log software environment and package versions.
- No leakage from post-outcome variables.

## 2. Grading Distribution (260 points)

Block	Focus	Points
A	Foundations: lifecycle, Python, EDA, scientific studies	20
B	Inference + visualization design and storytelling	20
C	SQL engineering + big-data systems thinking	25
D	Supervised ML + optimization + model selection	45
E	Unsupervised learning + dimensionality reduction	20
F	Deep learning + NLP + LMs/LLM agents	30
G	Ethics, fairness, robustness, governance	15
H	Integrated capstone implementation + communication	25
I	Production reliability and advanced diagnostics (Q15–Q20)	60
J (Bonus)	Advanced research/production extensions (optional)	+20
Total (A–I)		260
Optional bonus		+20

### Block A – Foundations (20 points)

#### Q1. Data Science Lifecycle and Problem Framing (10 pts)

Define an end-to-end lifecycle for this migration prediction problem:

- business objective and measurable success criteria,
- data assumptions and potential failure modes,
- deployment setting and monitoring plan.

**Deliverable:** 1-page structured problem statement with a lifecycle diagram.

#### Q2. Python Data Operations and EDA (10 pts)

Using Pandas/NumPy/Matplotlib/Seaborn:

1. perform robust schema checks (types, nulls, outliers),
2. produce at least six EDA plots with interpretation,
3. implement one reusable preprocessing function with tests.

**Deliverable:** EDA section in notebook + tested utility function.

### Block B – Inference and Visualization (20 points)

#### Q3. Scientific Studies and Inference (10 pts)

Address the following:

- observational vs experimental framing for migration analysis,
- sampling bias risks,
- one confidence interval and one hypothesis test with assumptions.

#### **Q4. Visualization Design + Storytelling (10 pts)**

Create a narrative dashboard (or notebook dashboard section) for non-technical stakeholders.  
Must include:

- clear KPI definitions,
- color and preattentive design rationale,
- one misleading-visualization pitfall and correction.

### **Block C – SQL and Data Engineering (25 points)**

#### **Q5. SQL-1/SQL-2 Advanced Querying (15 pts)**

Write SQL for:

1. 3-year moving average citations by country (window function),
2. top decile ranking and percentile bucketing,
3. one CTE-based cohort retention/transition style query.

#### **Q6. Data Leakage and Big-Data Architecture (10 pts)**

- Identify leaky features and defend exclusions.
- Propose a scalable batch + streaming architecture (Bronze/Silver/Gold or equivalent).
- Explain feature store design for training-serving consistency.

### **Block D – Supervised Learning and Optimization (45 points)**

#### **Q7. Linear/Logistic Models + Regularization (15 pts)**

- Fit baseline linear and logistic models.
- Derive and implement Elastic Net objective gradients (or use validated library implementation with derivation in report).
- Interpret coefficients, p-values/intervals (where applicable), and calibration.

#### **Q8. Optimization Deep Dive (10 pts)**

Compare SGD, Momentum, and Adam on a ravine-style objective.

- show trajectories,
- discuss curvature and oscillation,
- recommend optimizer under feature-scale heterogeneity.

### **Q9. Model Family Comparison (20 pts)**

Train and compare:

- SVM/KNN,
- Decision Tree/Random Forest,
- one boosting model (XGBoost/GradientBoosting/CatBoost if available).

Required:

1. cross-validation protocol,
2. hyperparameter search strategy,
3. error analysis and confusion patterns.

## **Block E – Unsupervised Learning (20 points)**

### **Q10. Dimensionality Reduction (10 pts)**

- PCA with explained variance ratio,
- one additional method (random projection, t-SNE, or UMAP),
- interpretation of latent dimensions.

### **Q11. Clustering (10 pts)**

- K-Means with elbow and silhouette analysis,
- DBSCAN (or equivalent density method),
- compare cluster stability and practical meaning.

## **Block F – Deep Learning, NLP, and LMs (30 points)**

### **Q12. Neural Networks and Sequence Models (15 pts)**

Complete one tabular NN and one sequence/NLP model experiment:

- MLP or shallow feed-forward network for tabular task,
- CNN or RNN/LSTM/GRU for text/sequence variant,
- compare against classical baseline.

### **Q13. Language Models and LLM Agents (15 pts)**

- design a small agentic workflow (retrieval/planning/tool-use pseudocode acceptable),
- define evaluation criteria (faithfulness, hallucination rate, safety),
- discuss governance constraints for academic/enterprise deployment.

## **Block G – Ethics and Governance (15 points)**

### **Q14. Fairness, Bias, and Responsible Deployment (15 pts)**

- evaluate subgroup metrics (e.g., by country/education),
- discuss historical-policy bias and proxy discrimination,
- propose human-in-the-loop and override policy.

## **Block H – Integrated Capstone (25 points)**

### **Capstone Task**

Deliver a full-stack implementation that includes:

1. data preprocessing and leakage-safe training,
2. model card and experiment tracking summary,
3. SHAP-based local and global explainability,
4. deployment recommendation with monitoring thresholds.

#### **Required capstone outputs:**

- one local explanation for a high-citation candidate predicted as no-migration,
- one global feature-importance plot,
- one fairness slice table,
- one executive summary for non-technical stakeholders.

## **Block I – Production Reliability Extension (30 points)**

### **Q15. Calibration and Threshold Policy (10 pts)**

Using your best supervised model from earlier sections:

- generate a reliability/calibration curve,
- compute at least one probabilistic calibration metric (e.g., Brier score, ECE),
- derive two threshold policies:
  1. threshold maximizing F1,
  2. threshold minimizing an asymmetric cost (e.g., FN cost > FP cost).

#### **Deliverables:**

- calibration plot,
- threshold-vs-metric tradeoff plot,
- final threshold recommendation with justification.

## **Q16. Drift Detection and Monitoring Design (10 pts)**

Define and execute a drift analysis between two data windows (time-based if possible).

- compute PSI for numeric features and rank by severity,
- include one categorical drift indicator (e.g., JS divergence over country distribution),
- propose a monitoring policy (warning/critical thresholds and retraining triggers).

### **Deliverables:**

- drift table (feature, metric, status),
- drift ranking figure,
- concise monitoring SOP for production.

## **Q17. Counterfactual Recourse Analysis (10 pts)**

For near-boundary negative predictions, estimate minimal actionable changes needed to flip outcome.

- choose at least two actionable features (e.g., GitHub activity, citations),
- compute minimal intervention per candidate under realistic caps,
- report recourse success rate and median intervention magnitude by feature.

### **Deliverables:**

- recourse examples table,
- recourse-effort summary plot,
- discussion of practicality/ethics of suggested interventions.

## **Q18. Temporal Backtesting and Rolling Validation (10 pts)**

Run a temporal robustness experiment.

- Use chronological folds where a valid time column exists.
- If no valid time column exists, define and justify a fallback ordering strategy.
- Report fold-wise performance (AUC, F1) and decay relative to the first fold.
- Add a drift-aware interpretation using a feature drift proxy (e.g., mean PSI).

### **Deliverables:**

- temporal backtest table (csv),
- degradation figure (png),
- explanation of fallback strategy (if used) in the report.

## **Q19. Uncertainty Quantification and Coverage (10 pts)**

Quantify predictive uncertainty and validate reliability.

- Implement conformal or calibrated probability intervals.
- Evaluate empirical coverage across multiple confidence levels.
- Report interval width and under-coverage risk.

**Deliverables:**

- coverage summary table (`csv`),
- coverage-vs-confidence figure (`png`),
- policy note for handling low-confidence cases.

## **Q20. Fairness Mitigation Experiment (10 pts)**

Design and evaluate a mitigation intervention under explicit policy constraints.

- Train a baseline model and compute subgroup fairness metrics.
- Apply one mitigation method (reweighing/thresholding/other justified method).
- Compare pre/post fairness and pre/post utility.
- Define and evaluate policy constraints (e.g., max acceptable AUC drop).

**Deliverables:**

- pre/post mitigation comparison table (`csv`),
- fairness-performance tradeoff figure (`png`),
- deployment recommendation based on policy constraints.

## **Block J – Advanced Extensions (Bonus +20)**

Any subset earns partial bonus. Keep results reproducible and justified.

1. **Causal framing (5 pts):** propose a DAG for migration, identify (in)valid adjustment sets, and discuss identifiability limits.
2. **Uncertainty (5 pts):** add conformal prediction or calibrated prediction intervals with empirical coverage check on the test split.
3. **Temporal robustness (5 pts):** perform time-based validation (e.g., train on earlier years, test on later) and compare to random split; report drift-aware degradation.
4. **Streaming/online serving (5 pts):** outline a minimal online inference design (feature freshness, idempotent writes, latency/SLA), plus a guardrail for out-of-distribution detection.

### **3. Academic Integrity and Professional Standards**

- Cite all external resources and model-generated assistance.
- Any copied code without attribution is a violation.
- Report negative results honestly.
- Prefer interpretable, audited pipelines over leaderboard-only optimization.

### **4. Bonus (up to +10 points)**

- Causal inference extension (DAG + identification discussion).
- Real-time pipeline prototype for streaming updates.
- Advanced uncertainty quantification (conformal or Bayesian approximation).