**AI for ITSM Demo Pack — Instructions (v2)**

*Includes Runbook overview — UC-02, UC-21, UC-31*

12 Aug 2025

1) What this pack is

A small, consistent ITSM data universe that supports three AI demos:

• UC-02 – Incident Triage & Assignment

• UC-21 – Automated KB from Ticket Clusters

• UC-31 – Agent Skill Matching Engine

Everything is CSV, no ServiceNow required. Use dataset\_catalog.csv as the index.

2) Load order (one-time setup)

Reference data first (used by all use-cases):

1) services\_catalog.csv

2) category\_tree.csv

3) cmdb\_ci.csv

4) users\_agents.csv, assignment\_groups.csv, agent\_group\_membership.csv

5) skills\_catalog.csv, agent\_skills.csv

6) (Optional NLP) synonyms\_glossary.csv

7) priority\_matrix.csv

Then the “facts”:

8) incidents\_resolved.csv (historical corpus, 300 rows)

Then the live/demo inputs:

9) workload\_queue.csv (open items to assign)

10) agent\_capacity\_snapshots.csv, agent\_performance\_history.csv, schedules.csv

For KB drafting:

11) kb\_templates.csv (scaffolds)

12) kb\_articles.csv (empty target the AI will write to)

3) How the files relate (mental model)

Think lookups + facts + live queue:

• Lookups: services, categories, CIs, agents, groups, skills.

• Facts: incidents\_resolved.csv (titles, descriptions, labels, resolution).

• Live work: workload\_queue.csv (items needing assignment now).

• Operational context: capacity, performance history, schedules.

• Knowledge: templates → articles.

Flow:

text (incidents, workload) → normalise (synonyms, categories)

→ UC-02 triage (priority, group) + UC-21 clusters → UC-21 KB drafts (templates)

→ UC-31 matching (skills + capacity + schedule + quality)

4) Use-case runbooks

UC-02 — Incident Triage & Assignment

Data: incidents\_resolved.csv; priority\_matrix.csv; category\_tree.csv; services\_catalog.csv; cmdb\_ci.csv; (optional) synonyms\_glossary.csv

Pipeline: Clean text → features (text + categorical) → predict priority & group → evaluate → export triage\_predictions.csv

UC-21 — Automated KB Article Creation from Ticket Clusters

Data: incidents\_resolved.csv; kb\_templates.csv; (optional) category\_tree.csv

Pipeline: Clean text → embed + cluster → rank clusters → summarise → draft KB → export kb\_articles.csv (drafts)

UC-31 — Agent Skill Matching Engine

Data: workload\_queue.csv; users/skills/groups; capacity; schedules; performance

Pipeline: Score agents (skill, capacity, availability, quality, SLA) → route → export routing\_decisions.csv

5) Minimal demo script (10–15 minutes)

1) Load all CSVs and run QA checks

2) Show lookups (services, categories, people/skills)

3) UC-02: triage predictions on sample incidents

4) UC-21: cluster corpus; draft 3 KBs via templates

5) UC-31: route a P1 item with rationale

6) File cheat-sheet (key columns)

incidents\_resolved.csv → short\_description, description, service\_id, ci\_id, category\_id, impact, urgency, true\_priority, true\_assignment\_group\_id, time\_to\_resolve\_mins

workload\_queue.csv → required\_skills, priority, sla\_due, service\_id, category\_id, ci\_id

agent\_skills.csv → agent\_id, skill\_id, level, certified

agent\_capacity\_snapshots.csv → open\_tasks, wip\_limit, available

agent\_performance\_history.csv → avg\_handle\_time\_mins, fcr\_rate, csat, success\_rate

schedules.csv → local working hours per agent

kb\_templates.csv → title/body sections for KB scaffolding

7) Outputs to produce (deliverables)

• triage\_predictions.csv (UC-02)

• kb\_articles.csv populated with drafts (UC-21)

• routing\_decisions.csv (UC-31)

8) Guardrails & tips

• Keep runs deterministic (fix random seeds)

• Reuse the same text normalisation across UC-02 and UC-21

• Use ids for joins; avoid hard-coded names

• Treat ground\_truth\_cluster as evaluation-only for UC-21

• AU timestamps; no PII included

9) About Runbook.xlsx — what’s inside and how to use it

Tabs you’ll find:

• Overview — what the pack is, the three use-cases, where to start.

• Load Order — exact sequence to import CSVs (lookups → facts → live inputs).

• Data Dictionary — every file and column, meaning, and an example value pulled from the data.

• Joins & Keys — foreign-key relationships and purpose (e.g., incidents.service\_id → services.service\_id).

• Use-Cases — inputs, pipeline steps, and expected outputs for UC-02/21/31.

• Quality Checks — quick QA tests (row counts, uniqueness, null thresholds) to run post-load.

• Deliverables — output file specs your pipelines should emit.

• Demo Script — a 10–15 minute sequence for the live demo.

• Tips & Pitfalls — guardrails to keep results consistent and explainable.

• Changelog — note changes to data/assumptions for repeatability.

How to use it:

1) Open ‘Load Order’ and import the CSVs in order.

2) Validate with ‘Quality Checks’ (copy checks into your notebooks/pipelines).

3) Implement each use-case using the ‘Use-Cases’ tab as your checklist.

4) Produce the three deliverables (triage\_predictions.csv, kb\_articles.csv, routing\_decisions.csv).

5) Record any changes in ‘Changelog’.