

HiLabs Hackathon 2025: Specialty Standardization Challenge

Problem Statement: Standardizing Provider Specialties to NUCC Taxonomy

Introduction

HiLabs invites you to participate in an exciting healthcare data challenge — **Provider Specialty Standardization** — one of the most critical and foundational problems in U.S. healthcare data quality.

Every health plan maintains millions of provider records, and each provider is associated with one or more **specialties** such as *Cardiology*, *Dermatology*, or *Pediatric Surgery*. However, in raw roster or provider files, these specialties are often captured as **free-text entries** — for example:

“Cardio”, “ENT Surgeon”, “Pediatrics - General”, “Addiction Med.”

Such inconsistencies cause downstream issues like data mismatches, claim-routing errors, and network adequacy gaps. To solve this, health plans depend on the **NUCC Taxonomy**, a **federal standard** that defines a uniform list of specialties and assigns each a **unique taxonomy code**.

Your mission: **Build an intelligent system that takes unstandardized specialties and maps them to official NUCC taxonomy codes.**

By doing so, you will directly contribute to solving a real-world challenge faced by major U.S. health plans and data platforms.

The Challenge

You are required to build a tool or model that:

1. Accepts an **input CSV** of raw, unstandardized provider specialties.
2. Maps each specialty to one or more **official NUCC taxonomy codes**.
3. Handles **abbreviations, misspellings, synonyms, partial words, and junk inputs**.
4. Returns “**JUNK**” if no confident mapping exists.

The NUCC Taxonomy Dataset

The **NUCC (National Uniform Claim Committee) Health Care Provider Taxonomy Code Set** is a **federal classification system** used throughout the U.S. healthcare industry to standardize provider specialties.

It is the official reference maintained by the **American Medical Association (AMA)** and adopted by the **Centers for Medicare & Medicaid Services (CMS)** to ensure consistent provider classification nationwide. Each taxonomy code uniquely identifies a specialty, and health plans use this taxonomy for credentialing, claims processing, and provider network management.

Each row in the NUCC dataset represents one **taxonomy code**, which describes a specific type of healthcare provider or sub-specialty.

| Column | Description |
|------------------------------------|--|
| code | The official NUCC taxonomy code — this is your primary output field . Each code uniquely identifies a specialty. |
| classification | The main specialty area (e.g., <i>Internal Medicine, Surgery, Pediatrics</i>). |
| specialization | The sub-specialty or finer detail of the classification (e.g., <i>Cardiovascular Disease</i> under <i>Internal Medicine</i>). |
| display_name | A readable combination of classification and specialization (e.g., <i>Internal Medicine - Cardiovascular Disease</i>). |
| grouping | The broader professional domain (e.g., <i>Allopathic & Osteopathic Physicians</i>). |
| definition / notes / status | Additional context about the specialty, its description, and whether it is active or deprecated. |

Participants will receive:

- The complete **NUCC taxonomy master dataset** (`nucc_taxonomy_master.csv`)
- A **sample input specialties file** (`input_specialties.csv`)

Your solution must use this dataset to map raw input specialties to the most appropriate taxonomy code(s)

Input and Output

Input

- **Format:** CSV
- **Column:** raw_specialty
- **Example Values:**
 - raw_specialty
 - Anesthesiology
 - Cardio
 - Pain & Spine Doc
 - OBGYN
 - Something random

Reference Data

- **NUCC Master Dataset:** nucc_taxonomy_master.csv
 - Columns: code, grouping, classification, specialization, display_name, definition, status

Output

- **Format:** CSV
- **Columns:**
 - raw_specialty — input value
 - nucc_codes — pipe-separated list of taxonomy codes or JUNK
 - confidence — float (0–1) indicating model confidence
 - explain — short rationale for mapping decision

Example Output:

| Input Specialty | Junk Flag | Output Standardized Taxonomies |
|----------------------------------|-----------|---|
| ABC | Y | — |
| XYZ | Y | — |
| Anesthesiology | N | 207L00000X 207LA0401X 207LC0200X 207LH0002X 207LP2900X 207LP3000X |
| Addiction Medicine | N | 207LA0401X |
| Allergy & Immunology | N | 207K00000X 207KA0200X 207KI0005X |
| Clinical & Laboratory Immunology | N | 207KI0005X |

Rules & Requirements

- **No external API calls** — all processing must be done locally.
- Submissions must include:
 1. Full working code in a public GitHub repository.
 2. A README.md explaining setup, preprocessing logic, and approach.
 3. A single script or notebook that runs end-to-end:
 4. `python standardize.py --nucc nucc_taxonomy_master.csv --input input.csv --output output.csv`
- The model must handle:
 - **Misspellings:** *Anesthesiolg* → *Anesthesiology*
 - **Abbreviations:** *ENT* → *Otolaryngology*, *OBGYN* → *Obstetrics & Gynecology*
 - **Multi-specialties:** *Cardio/Diab* → *Cardiology + Endocrinology*
 - **Noisy entries:** *Dept of Pediatrics, Clinic – Family Medicine*
- For ambiguous inputs, return **all matching NUCC codes** (pipe-separated).
- If confidence < threshold → mark as **JUNK**.

Evaluation Criteria

| Metric | Weightage | Description |
|--------------------------|-----------|--|
| Mapping Accuracy | 40% | Correctness of NUCC taxonomy mapping |
| Handling of Edge Cases | 20% | Robustness to abbreviations, typos, and noise |
| Innovation & Scalability | 20% | Efficiency, explainability, and approach novelty |
| Execution Time | 10% | Must process 20,000 rows in \leq 15 minutes |
| Final Round Q&A | 10% | Clarity of reasoning and design choices |

Suggestions

- Use **fuzzy matching**, **synonym dictionaries**, or **embedding-based similarity** to improve mapping accuracy.
- Create a custom **synonyms.csv** to handle recurring short forms or medical slang.
- Tune your **confidence threshold** using a small validation set.
- Ensure **deterministic results** (same input \rightarrow same output).

Deliverables

- **Codebase:** Executable locally with no hidden dependencies.
- **README.md:** Describes logic, preprocessing pipeline, and threshold calibration.
- **output.csv:** Output for the provided sample input.
- **synonyms.csv (optional):** Custom abbreviation/synonym mappings.

Submission Format

- **GitHub repository link** (public access required).
- Must include:
 - Sample input and output files.
 - Execution command.
 - Any precomputed assets (embeddings, synonym tables, etc.).



Evaluation Process

Submissions will be reviewed by the **HiLabs Product and Data Science teams**.

Top-performing teams will advance to a **technical Q&A** round, with **hiring opportunities** based on performance.



IIT KANPUR

2025