5.ICUProceduresFeatures

December 5, 2025

1 ICU Procedures Feature Script

1.1 Description

This script analyzes the timing and frequency of invasive procedures performed during ICU stays using `procedureevents.csv`.

1.2 Clinical Justification for HAPI Research

The frequency of procedures acts as a proxy for patient instability and immobility: * **Activity/Mobility Limitation:** Patients undergoing frequent procedures in the first 24-48 hours are often physically restrained, sedated, or too unstable to be turned/repositioned regularly (a key prevention strategy). * **Device Related Pressure:** High procedural volume often correlates with higher use of medical devices (catheters, lines), which are specific sources of localized pressure injuries.

1.3 Inputs & Outputs

- **Inputs:** `icustays.csv`, `procedureevents.csv`
- **Output:** `icu_procedures_feat.csv`
- **Key Features:**
  - `num_icu_procedures`: Total count of invasive procedures.
  - `num_icu_procedures_0_48h`: Procedures occurring in the critical first 48 hours.
  - `had_icu_procedure`: Binary flag for high-acuity intervention.

[1]:
```
import os
import pandas as pd
```

[2]:
```
#Configuration
BASE_DIR = r"D:\School\5141"

ICUSTAYS_PATH       = os.path.join(BASE_DIR, "icustays.csv", "icustays.csv")
PROCEDUREEVENTS_PATH = os.path.join(BASE_DIR, "procedureevents.csv")

OUTPUT_PATH         = os.path.join(BASE_DIR, "icu_procedures_feat.csv")
```

[3]:
```
#Loading Functions
def load_icustays(path: str):
    """
```

```
    Load icustays.csv and keep only the columns needed for
    time-based ICU features.

    Keep:
        - hadm_id: hospital admission ID
        - stay_id: ICU stay ID (MIMIC-IV)
        - intime, outtime: ICU admission & discharge times
    """

    icu = pd.read_csv(
        path,
        usecols=["subject_id", "hadm_id", "stay_id", "intime", "outtime"],
        low_memory=False
    )

    # Ensure correct dtypes
    icu["hadm_id"] = icu["hadm_id"].astype("Int64")
    icu["stay_id"] = icu["stay_id"].astype("Int64")

    icu["intime"] = pd.to_datetime(icu["intime"])
    icu["outtime"] = pd.to_datetime(icu["outtime"])

    return icu

def load_procedureevents(path: str):
    """
    Load ICU procedure events.

    Expected columns (MIMIC-IV ICU `procedureevents`):
        - subject_id
        - hadm_id
        - stay_id
        - starttime
        - endtime
        - itemid (type of procedure)
    """
    proc = pd.read_csv(
        path,
        usecols=["subject_id", "hadm_id", "stay_id", "starttime", "endtime",␣
 ↪"itemid"],
        low_memory=False
    )

    proc["hadm_id"] = proc["hadm_id"].astype("Int64")
    proc["stay_id"] = proc["stay_id"].astype("Int64")

    proc["starttime"] = pd.to_datetime(proc["starttime"])
```

```
    proc["endtime"] = pd.to_datetime(proc["endtime"])

    return proc
```

[4]:
```
# Feature Engineering Function
def build_icu_procedure_features(
    icu: pd.DataFrame,
    proc: pd.DataFrame
):
    """
    Build ICU procedure features per (hadm_id, stay_id).

    Steps:
      1. Merge procedureevents with icustays to get the ICU intime.
      2. Compute `hours_from_intime` for each procedure.
      3. Aggregate counts & timing per ICU stay.
      4. Left-join back to all ICU stays so stays with no procedures still␣
 ↪appear with zeros/NaNs.
    """

    # Merge procedure events with ICU stays on stay_id & hadm_id
    merged = proc.merge(
        icu[["hadm_id", "stay_id", "intime"]],
        on=["hadm_id", "stay_id"],
        how="inner",  # only keep procedures that match an ICU stay
        validate="many_to_one"
    )

    # Drop rows where intime or starttime is missing
    merged = merged.dropna(subset=["intime", "starttime"])

    # Compute hours from ICU admission for each procedure
    merged["hours_from_intime"] = (
        (merged["starttime"] - merged["intime"]).dt.total_seconds() / 3600.0
    )

    # Boolean flags for time windows
    merged["is_0_24h"] = merged["hours_from_intime"].between(0, 24,␣
 ↪inclusive="left")
    merged["is_0_48h"] = merged["hours_from_intime"].between(0, 48,␣
 ↪inclusive="left")

    # Aggregate per (hadm_id, stay_id)
    grp = merged.groupby(["hadm_id", "stay_id"], dropna=False)

    feat = grp.agg(
        num_icu_procedures=("itemid", "size"),
```

```
        num_icu_procedures_0_24h=("is_0_24h", "sum"),
        num_icu_procedures_0_48h=("is_0_48h", "sum"),
        first_proc_hours_from_intime=("hours_from_intime", "min"),
        last_proc_hours_from_intime=("hours_from_intime", "max"),
    )

    # Derived boolean flags
    feat["had_icu_procedure"] = (feat["num_icu_procedures"] > 0).astype("Int64")
    feat["had_icu_procedure_0_24h"] = (feat["num_icu_procedures_0_24h"] > 0).
↪astype("Int64")
    feat["had_icu_procedure_0_48h"] = (feat["num_icu_procedures_0_48h"] > 0).
↪astype("Int64")

    # Left-join back to all ICU stays to include those with no procedures
    icu_key = icu[["hadm_id", "stay_id"]].drop_duplicates().
↪set_index(["hadm_id", "stay_id"])

    all_feat = icu_key.join(feat, how="left")

    # For stays with no procedures, fill counts with 0 and flags with 0
    count_cols = [
        "num_icu_procedures",
        "num_icu_procedures_0_24h",
        "num_icu_procedures_0_48h",
    ]
    flag_cols = [
        "had_icu_procedure",
        "had_icu_procedure_0_24h",
        "had_icu_procedure_0_48h",
    ]

    for c in count_cols:
        all_feat[c] = all_feat[c].fillna(0).astype("Int64")

    for c in flag_cols:
        all_feat[c] = all_feat[c].fillna(0).astype("Int64")

    all_feat = all_feat.reset_index()

    print(f"Final ICU procedure feature rows: {len(all_feat):,}")
    return all_feat
```

[5]:
```
# Main Function
def main():
    # Load raw tables
    icu = load_icustays(ICUSTAYS_PATH)
    proc = load_procedureevents(PROCEDUREEVENTS_PATH)
```

```
    #  Build features
    icu_proc_feat = build_icu_procedure_features(icu, proc)

    # Save to CSV
    print(f"Saving ICU procedure features to: {OUTPUT_PATH}")
    icu_proc_feat.to_csv(OUTPUT_PATH, index=False)
    print("Done.")


if __name__ == "__main__":
    main()
```

```
Final ICU procedure feature rows: 94,458
Saving ICU procedure features to: D:\School\5141\icu_procedures_feat.csv
Done.
```