EDA Project Summary

Hospital Patient Records Cleanup and Analytics

comprehensive data curation and predictive analytics pipeline for hospital patient records

project overview

domain: healthcare

level: final year b.tech / data science professionals

difficulty: high

objective: clean, curate, and prepare hospital data for epidemiological research and predictive analytics, ensuring data quality, regulatory compliance (hipaa), and readiness for advanced analytics workflows.

datasets

the project works with 6 primary datasets:

1. **patients.csv** - patient demographic details
2. **visits.csv** - admission/discharge data
3. **diagnoses.csv** - icd-10 diagnosis codes
4. **medications.csv** - prescribed drugs per visit
5. **staff.csv** - doctor/nurse assignments
6. **hospital_info.csv** - hospital unit metadata

project structure

```
hospital data curation/
├── data/
│   ├── raw/                    # original datasets
│   ├── cleaned/                # cleaned datasets
│   └── preprocessed/           # transformed and integrated datasets
```

```
├── src/
│   ├── config.py              # configuration settings
│   ├── utils.py               # utility functions
│   ├── data_loader.py         # data ingestion
│   ├── data_cleaner.py        # cleaning operations
│   └── validators.py          # validation logic
├── reports/
│   ├── profiling/             # ydata profiling reports
│   └── sweetviz/              # sweetviz analysis reports
├── logs/                       # validation and processing logs
├── models/                     # trained ml models
├── visualizations/            # charts and graphs
├── 01_data_ingestion.ipynb
├── 02_data_profiling.ipynb
├── 03_data_cleaning.ipynb
├── 04_data_integration.ipynb
├── 05_data_transformation.ipynb
├── 06_data_validation.ipynb
├── 07_regression_analysis.ipynb
├── 08_association_mining.ipynb
├── 09_classification_analysis.ipynb
└── requirements.txt
```

implementation phases

phase 1: data ingestion

- load all csv/excel datasets
- validate file structure and sources
- generate metadata reports
- version control raw data

phase 2: data profiling

- **ydata-profiling**: comprehensive statistical profiling
- **sweetviz**: interactive visualizations
- identify missing values, duplicates, outliers
- detect data quality issues

phase 3: data cleaning

- standardize column names to snake_case
- normalize date formats
- recode gender values (m/f/o → male/female/other)
- impute missing ages from date of birth
- validate icd-10 code formats
- remove duplicates and invalid entries
- handle extreme outliers

phase 4: data integration

- merge patients + visits on patient_id
- integrate diagnoses on visit_id
- integrate medications on visit_id
- handle referential integrity
- resolve entity conflicts

phase 5: data transformation

- derive length_of_stay
- generate readmission flags (30-day threshold)
- create age groups (0-18, 19-35, 36-60, 60+)
- bucket length of stay categories

- identify high-risk patients
- one-hot encode categorical variables
- normalize numerical features
- extract temporal features

phase 6: data validation

- assert business rules (no negative los, valid date sequences)
- validate icd-10 code formats
- check referential integrity
- generate validation reports using pytest
- create quality scorecards

phase 7: regression analysis

objective: predict length of stay

models:

- linear regression
- ridge regression (l2 regularization)
- lasso regression (l1 regularization)

features: age, gender, diagnosis_count, medication_count, admission_type

metrics: rmse, mae, r^2

phase 8: association rule mining

objective: discover diagnosis-medication co-occurrence patterns

techniques:

- apriori algorithm (diagnosis patterns)
- fp-growth (medication patterns)
- combined analysis (diagnosis → medication rules)
- readmission prediction rules

metrics: support, confidence, lift

phase 9: classification analysis

objective: predict 30-day readmission risk

models:

- decision tree classifier
- random forest classifier
- gradient boosting classifier

features: age_group, length_of_stay, diagnosis_count, medication_count, is_high_risk

metrics: accuracy, precision, recall, f1-score, auc-roc, confusion matrix

installation

```
# create virtual environment
python -m venv venv
.\venv\Scripts\Activate

# install dependencies
pip install -r requirements.txt
```

usage

1. place datasets in data/raw/

ensure all csv files are in the correct location:

- patients.csv
- visits.csv
- diagnoses.csv
- medications.csv
- staff.csv
- hospital_info.csv

2. run notebooks sequentially

execute notebooks in order from 01 to 09:

```
# start jupyter

jupyter notebook
```

3. generate data dictionary

```
python generate_data_dictionary.py
```

key deliverables

1. ✅ **cleaned datasets** (csv)

- clean_patients.csv

- clean_visits.csv

- clean_diagnoses.csv

- clean_medications.csv

2. ✅ **integrated datasets** (csv)

- master_patient_visits.csv

- transformed_master_dataset.csv

- encoded_dataset_for_ml.csv

3. ✅ **profiling reports** (html)

- ydata profiling reports

- sweetviz analysis reports

4. ✅ **validation reports** (txt/csv)

- validation_report.txt

- quality_scorecard.csv

- completeness_report.csv

5. ✅ **data dictionary** (csv/xlsx)

   - comprehensive column documentation

6. ✅ **trained ml models** (pkl)

   - regression models (linear, ridge, lasso)

   - association rules (csv)

   - classification models (decision tree, random forest, gradient boosting)

7. ✅ **visualizations** (png)

   - actual vs predicted plots

   - roc curves

   - confusion matrices

   - feature importance charts

   - association rule graphs

key performance indicators

- **average length of stay**: calculated from visit data
- **readmission rate**: percentage of patients readmitted within 30 days
- **high-risk patient rate**: percentage flagged as high-risk
- **data completeness**: percentage of non-null values
- **model accuracy**: classification and regression performance

tools and technologies

task	tools
data processing	pandas, numpy
profiling	ydata-profiling, sweetviz
cleaning	openpyxl, fuzzywuzzy
validation	pytest, great expectations
machine learning	scikit-learn, mlxtend

visualization	matplotlib, seaborn, plotly
documentation	markdown, jupyter notebooks