U.PORTO FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Licenciatura em Inteligência Artificial e Ciência de Dados

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Trabalho no âmbito da disciplina:

Ciência de Dados em Larga Escala

Machine Learning Pipeline

Authors:
Benedita Gonçaves
Hugo Torgo

Group: H

14 de agosto de 2025

Summary

1. INTRODUCTION

This report describes the project, titled Machine Learning Pipeline, carried out as part of the Large Scale Data Science course (CC3047) during the second semester of the third year of the Bachelor's degree in Artificial Intelligence and Data Science.

The aim of this project was to perform a full data analysis and build predictive models using the compressed 4.2 GB CHARTEVENTS.csv.gz file, which contains time-stamped events for patients during their ICU stay. We followed a traditional machine-learning pipeline, data pre-processing, data preparation, training and validation, analysis of results and interpretation, but with first exploring and visualizing each patient's data. Finally, we predicted ICU length of stay (LOS), choosing a 24-hour window of data to train our models.

In this work, we draw on the MIMIC-III database, primarily the CHARTEVENTS table accessed via BigQuery, to build a cohort of adult ICU stays. We started by performing exploratory data analysis (EDA) on static demographic variables (age, gender, insurance, etc.) and first-day time-stamped measurements (vital signs, fluid inputs, laboratory values, and so on). These raw events are aggregated into summary features, like means, minima, maxima, variability measures, and missingness flags, over a 24-hour window after ICU admission. We then merged these engineered features with demographics to form our model inputs.

For prediction, we trained a simple Linear Regression as a baseline and a more flexible Histogram Gradient Boosting Regressor (HGBR) with hyperparameter tuning. We evaluated both models on a held-out test set, comparing MAE and RMSE.

To understand computational trade-offs, we profiled two pipeline architectures: one that pushes all aggregation into BigQuery and pulls down only the final wide table versus a hybrid approach that uses Dask with BigQuery for distributed preprocessing before local model fitting.

Finally, we examined feature importances, conducted ablation studies separating static and time-series inputs.

2. DATA SOURCE AND DESCRIPTION

MIMIC-III and CHARTEVENTS

All raw CSV files (including CHARTEVENTS.csv.gz, ADMISSIONS.csv.gz, PATIENTS.csv.gz, and D_ITEMS.csv.gz) were originally obtained from this link that our teacher gave us.

Each file corresponds to a table in the MIMIC-III v1.4 release.

Once downloaded, we uploaded the compressed CSVs into a Google Cloud Storage bucket and loaded them into BigQuery for scalable processing. Below is an overview of the relevant tables and key columns:

- **CHARTEVENTS.csv.gz:** Contains time-stamped "chart" and "input" events for each ICU stay, such as vital signs, fluid volumes, medication infusions, and laboratory results.
  - **ICUSTAY_ID:** unique ICU encounter identifier (INTEGER).
  - **HADM_ID:** hospital admission ID (INTEGER).
  - **ITEMID:** numeric code for each measured/recorded variable; human-readable descriptions (e.g., "Heart Rate," "Enteral Nutrition Volume") are in D_ITEMS.
  - **CHARTTIME:** TIMESTAMP indicating when the event was recorded.
  - **VALUENUM:** FLOAT representing the numeric value of the event (e.g., heart rate in beats/minute, fluid volume in mL).
  - **VALUEUOM:** unit of measure (STRING), though canonical units are confirmed via D_ITEMS.
- **ADMISSIONS.csv.gz:** Provides hospital admission and discharge timestamps, as well as demographic/administrative fields (insurance, admission type). We used ADMISSIONS primarily to ensure each ICU stay fell within a valid hospital admission.

- **PATIENTS.csv.gz:** Contains patient demographics, such as date of birth, gender, and ethnicity, which we used to compute age at ICU admission and include static covariates in modeling.
- **D_ITEMS.csv.gz:** Maps each ITEMID to its full description and standardized unit. We referenced D_ITEMS to identify clinically relevant ITEMIDs for aggregation.

COHORT DEFINITION CRITERIA

To form a consistent cohort for ICU LOS prediction, we applied:

- **Adult Patients (age ≥ 18)** at ICU admission, where age = ICU_INTIME − DOB.
- **First ICU Stay per Hospital Admission:** If a single HADM_ID had multiple associated ICUSTAY_IDs, we retained only the earliest ICU admission.
- **Exclude Implausible LOS:** Discard stays with computed ICU LOS < 0 or ICU LOS > 30 days.
- **Minimum 24 Hours of Data:** Required that ICU_INTIME + 24 hours ≤ ICU_OUTTIME so every included stay possessed at least one full day of charted events.

COMPUTING ICU LOS

ICU length of stay (LOS) is calculated by taking the difference between the ICU discharge time and the ICU admission time and expressing that interval in days (including fractions). Both timestamps come from the ICUSTAYS table, and we ensure they fall within a valid hospital admission by joining to the ADMISSIONS table. All analyses and model targets use this ICU LOS measure.

3. EXPLORATORY DATA ANALYSIS (EDA)

We began by connecting to BigQuery and loading the ICUSTAYS, ADMISSIONS, and PATIENTS tables to assemble our cohort.

After computing each patient's age at ICU admission (and binning into four categories), we plotted basic demographics such as sex, age-bin, ethnicity, and insurance to confirm expected distributions.

Next, we generated a histogram of ICU LOS, noted its right skew (most stays under five days, a few outliers beyond 30 days), and applied a cutoff to remove extreme values.

Finally, we previewed raw time-series traces (e.g., heart rate and mean arterial pressure over the first 24 hours) for a few stays to verify recording patterns, which guided our decision to summarize each selected ITEMID via 24-hour aggregates.

4. DATA PREPROCESSING

Below is a concise summary of the transformations performed in the notebook:

- **Inspect and Drop Highly Missing Time-Series Features**

  Calculated the percentage of missing values for each column in the 24-hour "wide" feature table (wide24h_bq).

  Identified and removed any column with > 20 % missing entries.

- **Median-Impute Remaining Numeric Features**

  For all remaining numeric columns in wide24h_bq, computed the median and filled any NaN with that median value.

- **Clean Up Column Names**

  Replaced non-alphanumeric characters in column names with underscores so they remain BigQuery-safe.

- **Merge with Static Cohort Data**

  Performed a left join of cohort_demo (static demographics + ICU LOS) with the cleaned wide24h_bq on ICUSTAY_ID.

- **Filter Out Missing or Invalid LOS**

  Dropped any rows where ICU_LOS was NaN or ≤ 0.

- **Remove Non-Predictive Columns**

  Dropped identifiers and timestamps: ICUSTAY_ID, HADM_ID, SUBJECT_ID, INTIME, OUTTIME, ADMITTIME, and DOB.

- **Median-Impute Any Remaining Numeric Gaps**

  Identified all numeric features (except ICU_LOS) in the merged DataFrame and filled any remaining NaN with that column's median.

- **Collapse High-Cardinality Categories**

  For each categorical field (GENDER, AGE_BIN, ETHNICITY, INSURANCE, LANGUAGE, RELIGION, MARITAL_STATUS, ADMISSION_TYPE), kept only the top 20 most frequent levels; all others were labeled ”OTHER”.

- **One-Hot Encode Categorical Variables**

  Converted each collapsed categorical column into dummy variables (e.g., GENDER__Male, ETHNICITY__Caucasian, etc.).

5. MODEL DEVELOPMENT AND VALIDATION

In this section, we divide our fully preprocessed dataset into training and testing subsets, then fit and evaluate two regression models—a simple Linear Regression baseline and a more powerful Histogram Gradient Boosting Regressor (HGBR), including hyperparameter tuning and feature importance analysis.

- **Prepare Features (X) and Target (y)**

  Defined X as all columns except ICU_LOS, and y as the ICU_LOS column. Verified that neither X nor y contained any missing values.

- **80/20 Train/Test Split**

  Randomly held out 20 % of the stays for testing (with a fixed seed for reproducibility) and used 80 % for training. Confirmed that the LOS distribution in both sets was similar.

- **Baseline: Linear Regression**

  Fitted a simple LinearRegression on X_train, y_train. Predicted on X_test, computed MAE and RMSE, and plotted actual vs. predicted LOS (full range and zoomed 0–50 days) to inspect residuals.

- **Advanced: Histogram Gradient Boosting Regressor (HGBR)**

  - **Default HGBR**

    Trained HistGradientBoostingRegressor(random_state=42) on the same training set. Evaluated on X_test to obtain MAE and RMSE.

  - **Permutation Importances**

    Measured how shuffling each feature's values increased test-set MAE. Ranked and plotted the top 10 features by mean importance.

  - **Hyperparameter Tuning**

    Defined a parameter grid over learning_rate, max_depth, max_leaf_nodes, and min_samples_leaf. Ran RandomizedSearchCV (10 iterations, 3-fold CV, scoring = neg MAE) to find the best settings. Retrai-

ned an HGBR with those best parameters and reevaluated MAE/RMSE on X_test.

– **Residual Plot for Tuned HGBR**

  Plotted actual vs. predicted LOS to check for remaining systematic bias.

6. PERFORMANCE PROFILING AND SCALING DISCUSSION

To understand the trade-offs between a BigQuery-centric workflow and a hybrid Dask + BigQuery approach, we measured wall-clock times, peak memory usage, and model performance for each pipeline.

6.1. BIGQUERY ONLY PIPELINE

Workflow: Pull a fully materialized wide24h_bq table from BigQuery into Pandas, merge with static cohort data, drop or impute missing values, then train a linear regression.

Timings and Memory:

- Pull/Merge Time: 3.62 sec
- Train Time: 0.00 sec
- Total Time: 3.62 sec
- Peak Memory Increase: +961 MB

Predictive Performance:

- MAE: 4.70 days
- RMSE: 9.48 days

6.2. DASK + BIGQUERY PIPELINE

Workflow: Pull raw “first24” rows into Pandas, convert to a Dask DataFrame, perform the 24-hour filtering and aggregation in parallel, pivot to wide format, merge with static data, impute, and train.

Timings and Memory:

- Pull/Merge (including Dask aggregation) Time: 295.12 sec

- Train Time: 0.01 sec
- Total Time: 295.13 sec
- Peak Memory Increase: +12 MB

Predictive Performance:

- MAE: 4.63 days
- RMSE: 9.49 days

Conclusions:

- Both pipelines achieve nearly identical MAE/RMSE, since they use the same underlying features.
- BigQuery-Only is much faster (3.6 sec vs. 295 sec) but uses more local memory (+961 MB), making it ideal when the precomputed wide table already exists.
- Dask + BigQuery takes longer overall but uses very little extra memory (+12 MB) and allows to re-aggregate features locally without incurring repeated BigQuery scans—useful for iterative feature engineering.

7. FINAL ANALYSIS AND INTERPRETATION

Top Predictive Features:

Permutation importance revealed that first-day heart-rate statistics (maximum, minimum, mean, and variability) and minimum hemoglobin were the strongest predictors of ICU LOS. Scatter plots show that greater heart-rate variability and persistently high or low heart rates during the first 24 hours tend to coincide with longer stays.

Ablation Results:

We compared three models, static demographics only, time-series only, and both combined. Since nearly all static features were sparse or dropped, the static-only model could not be meaningfully trained. The time-series-only and full models performed almost identically (MAE ≈ 4.15 days, RMSE ≈ 8.75 days), indicating that first-24h vital summaries carry most of the predictive signal.

Missingness Patterns:

Demographic fields like LANGUAGE (≈43 % missing) and MARITAL_STATUS (≈17 % missing) had high gaps, whereas heart-rate summaries were missing for only ≈1.8 % of stays. This suggests that vital sign availability is reliable, but some demographic variables require cautious handling or imputation in a real-world system.

8. CONCLUSIONS

Our analysis shows that first-day physiological signals, especially heart-rate dynamics and minimum hemoglobin, carry most of the predictive power for ICU length of stay. Using a 24-hour aggregation window, our best model achieved an average error of around 4 days.

Comparing workflows revealed that pulling a precomputed feature table from BigQuery is far more time-efficient when working at scale, while a Dask-augmented pipeline offers flexibility for iterative feature engineering at the cost of longer runtimes.

Future work could explore richer baseline comorbidity indices, extend beyond day one, and validate on external ICU cohorts to further refine LOS predictions and support real-time clinical decisions.