Designing and Deploying BI and ML Solutions in a Health Insurance Company

Dashboards, Customer Segmentation, and Predictive Cost Modeling

Ignacio Balasch Solá
ignacio@balasch.es

January – July 2025

Undergraduate Thesis · Industrial Engineering · Pontifical University of Comillas (ICAI)
This report describes methodology and engineering approach.
Proprietary data and company-specific details have been abstracted for confidentiality.

Abstract

Over a six-month thesis placement at a major health insurance provider in Spain, I designed and delivered three end-to-end data solutions for the company's individual policyholder portfolio: a **suite of 11 interactive Tableau dashboards** deployed to the company's BI portal and adopted for ongoing operational monitoring; an **unsupervised ML segmentation** of the insured population using K-Means and OPTICS clustering, surfacing actionable profitability patterns across demographic groups; and a **supervised cost-prediction model** (XGBoost) that competed with—and in several metrics outperformed—the underwriting department's actuarial pricing tables.

I began this project with **no prior experience in SQL, business intelligence tooling, or machine learning**. The work spanned four custom ETL processes in SQL Server, extensive exploratory statistical analysis, multiple clustering algorithm comparisons, ten iterative rounds of model training and postprocessing, and a rigorous head-to-head evaluation framework against actuarial baselines. The full thesis runs to approximately 400 pages, including all SQL and Python code.

This document provides an executive overview of the project: its context, what was delivered, and how each solution was built.

1 Introduction

1.1 The Company and the Problem

The company operates in the Spanish private health insurance market, serving both corporate and individual policyholders. The individual segment had received comparatively less analytical attention, and the company recognized the need to better understand the behavior and characteristics of these policyholders.

Before this project, there was no centralized system for visualizing and analyzing individual policyholder data, resulting in fragmented insights and delayed decision-making. There was no systematic segmentation of the portfolio, no self-service tooling for non-technical stakeholders, and no machine-learning-driven cost prediction capability.

1.2 My Mandate and Starting Point

I was brought in as a thesis student under the supervision of the Data & Analytics Manager. My mandate was to **build a comprehensive analytical capability around the individual policyholder portfolio**, structured around three deliverables:

1. **BI Dashboards**: Interactive Tableau dashboards for operational monitoring across demographics, geography, and retention.
2. **Customer Segmentation**: Unsupervised ML clustering to identify natural policyholder groups and analyze their utilization patterns.
3. **Cost Prediction**: A supervised ML model to predict individual healthcare expenses, benchmarked against the underwriting department's actuarial pricing.

My background was in Industrial Engineering—analytical thinking, optimization theory, and basic Python—not in data engineering or machine learning. The technical stack required for this project (SQL, Tableau, Scikit-learn, XGBoost) was learned during the course of its construction, using online platforms (Pluralsight, edX, HarvardX), documentation, and ChatGPT as a learning and debugging tool.

The following sections describe each solution in detail.

2 Dashboards

I designed a custom ETL pipeline in SQL Server that consolidated monthly insurance statuses, enrollments, terminations, and plan transitions from multiple source tables into a clean analytical schema. On top of this, I built **eleven interactive Tableau dashboards** covering portfolio evolution, demographic breakdowns (age, sex, segment), geographic distribution, retention and churn analysis, and broker performance.

The dashboards were developed iteratively through stakeholder review cycles, deployed to the company's internal BI portal in March 2025, and adopted by the analytics team for ongoing operational monitoring. A colleague developed complementary financial dashboards in parallel using the same infrastructure.

3 Learning Scikit-learn

I taught myself Scikit-learn during the project—working through documentation, online courses, and iterative experimentation. The library's consistent API made it possible to build production-grade pipelines relatively quickly. The core workflow I used across both the clustering and prediction components followed this pattern:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.cluster import KMeans  # or any model

# 1. Load data
df = pd.read_csv("data.csv")
X = df[feature_columns]

# 2. Build preprocessor (scale numbers, encode categories)
preprocessor = ColumnTransformer([
    ("num", StandardScaler(), numerical_cols),
    ("cat", OneHotEncoder(), categorical_cols)
])

# 3. Create pipeline: preprocessor + model
pipeline = Pipeline([
    ("preprocessing", preprocessor),
    ("model", KMeans(n_clusters=8))  # swap for any algorithm
])

# 4. Fit
pipeline.fit(X)
```

This same pattern—load, preprocess, pipeline, fit—was reused across every ML component of the project: Random Forest for feature importance, K-Means and OPTICS for clustering, and XGBoost for cost prediction. The consistency of the API meant that once I understood the structure, adapting it to new algorithms was straightforward—though understanding *which* algorithm to use, how to evaluate it, and how to interpret its outputs required significant learning beyond the code itself.

4 Customer Segmentation

4.1 Objective

The goal was to identify natural groupings within the insured population based on demographic characteristics, and then analyze how healthcare costs vary across those groups. This provides the business with a data-driven basis for pricing adjustments, targeted marketing, and risk management.

4.2 Exploratory Analysis and Feature Selection

Before clustering, I conducted extensive statistical analysis of the target variable (average monthly healthcare expenses), examining its distribution across key features: age bracket, sex, geographic region, plan type, tenure, and family structure. The distribution was heavily right-skewed and zero-inflated—most policyholders incur minimal costs, while a small minority accounts for very high expenses.

To identify which features carry the most predictive signal, I trained a Random Forest classifier and extracted feature importance rankings. The results were cross-validated against the business team's domain expertise to select the final feature set for clustering.

4.3 Clustering Methodology

The methodology followed four steps:

1. **Feature selection**: Combining statistical evidence with domain expertise.
2. **Clustering**: Grouping policyholders by demographic attributes, deliberately excluding healthcare expenses so that clusters reflect *who* the policyholders are.
3. **Expense analysis**: Reintegrating expense data and analyzing cost distributions per cluster.
4. **Validation**: Testing whether patterns observed within clusters hold across the broader population.

I applied two complementary algorithms: **K-Means** (partitioning into a fixed number of clusters, determined by the elbow method) and **OPTICS** (a density-based algorithm that discovers clusters of varying shape and density without requiring a predefined count). Three trials were conducted with different feature configurations and data partitions.

4.4 Key Outputs

The analysis produced cluster-level demographic profiles with associated expense statistics, from which I derived:

- The most and least profitable demographic groups for individual plans.
- The highest and lowest cost demographics within family plans—with a clear distinction between true cost signals and artifacts of family premium pooling.
- Geographic overrepresentation patterns relevant to regional commercial strategy.

Each insight was validated against the general population before being included in the final report. The methodology was documented to be repeatable on updated data.

5 Cost Prediction

5.1 Objective

The goal was to build a machine learning model that predicts individual healthcare expenses using demographic features available at the time of policy enrollment, and to evaluate whether it could match or exceed the accuracy of the underwriting department's existing actuarial pricing.

5.2 Data Preparation

Two additional ETL pipelines were developed for this component. The training pipeline extracted individual-level records from 2017 to 2023, incorporated family-level aggregation, and applied inflation adjustments. The testing pipeline replicated the same logic for new 2024 policyholders, enabling prospective evaluation.

5.3 Model

I trained an **XGBoost** gradient-boosted regression model. The input features were age bracket, sex, geographic zone, insurance plan type, and person type (primary insured vs. dependent). The model was trained on 2017–2023 data and evaluated on 2024 enrollees.

5.4 Comparison Framework

To benchmark the model, I needed the underwriting department's expected cost per policyholder. No such table existed explicitly—the actuarial team works with a pricing methodology, not a cost prediction table. I reverse-engineered their implied cost predictions by extracting pricing logic from Excel files, working directly with the actuarial team to structure the data into a comparable format. Three versions of the actuarial benchmark were produced (base pricing, family-discount pricing, and full-discount pricing) to ensure a comprehensive comparison.

5.5 Postprocessing

The raw model required postprocessing to produce operationally reasonable predictions. The final pipeline applied:

1. **Global scaling**: Aligning aggregate predicted totals with observed training totals.
2. **Group-specific clipping**: Bounding predictions within demographic-group-specific ranges (by sex, zone, and age bracket).
3. **Safeguard constraints**: Preventing unstable bounds in sparse groups by constraining clipping ranges using overall population statistics.

This pipeline was developed through ten iterative rounds, with the final version designed for transparency and interpretability.

5.6 Results

The final model closely matched the actuarial department's pricing tables across multiple evaluation dimensions. In the proportion of individual cases where the ML prediction was closer to the real expense, the model consistently outperformed the actuarial baseline. The model was serialized for deployment, and a complete pricing table covering all demographic combinations was produced as the final deliverable.

6 On the Complexity of Health Insurance Data

Health insurance data is inherently difficult to work with. The expense distribution in this project was heavily right-skewed and zero-inflated: the majority of policyholders incur minimal or no healthcare costs in a given month, while a small minority generates expenses orders of magnitude higher. This creates a dataset where the mean is unrepresentative, outliers are real (not errors), and any model that predicts "average" behavior will be wrong for most individuals. Individual health outcomes are ultimately driven by factors—genetics, accidents, lifestyle changes—that demographic features alone cannot capture. The challenge is not to predict perfectly, but to distinguish higher-risk from lower-risk individuals well enough to support sound pricing decisions.

7 Bridging Academic ML and Business Reality

One of the most important lessons from this project was understanding the gap between a model that performs well statistically and one that is viable for business use.

My initial prediction model achieved a positive R^2 value. By textbook standards, this was a success. But when I examined the predictions at a granular level, the model was assigning near-zero expected costs to certain demographic groups. Statistically, this improved overall accuracy. Operationally, it was unusable—no insurer would price a policy at close to zero euros for any demographic, regardless of what the historical data suggests.

This forced a shift in how I thought about model quality. The relevant question was not "Does this model minimize prediction error?" but "Does this model produce outputs that a business can act on?" Answering that required postprocessing: scaling predictions to match aggregate totals, clipping extreme values within reasonable bounds, and applying safeguard constraints.

In practice, the postprocessing pipeline addressed the same concerns that actuaries handle through floors, caps, and credibility adjustments—balancing predictive accuracy with operational constraints and commercial risk. The difference was that my pipeline derived its parameters from the data itself rather than from actuarial heuristics. The end result was a model whose predictions were both statistically grounded and operationally deployable.

This tension between statistical optimality and operational viability is not something I encountered in any course. It is one of the defining challenges of applied data science.

8 Conclusion

This project was a privilege. Over six months, I designed, built, and deployed three production-grade analytical solutions for a real business—each requiring a different set of tools, a different mode of thinking, and a different kind of collaboration.

The system delivers tangible outputs: eleven dashboards now used for operational monitoring, a customer segmentation methodology that surfaced actionable profitability patterns, and a cost-prediction model that competes with actuarial pricing tables refined over years. The project also required navigating enterprise data systems, collaborating with domain experts across departments, and designing analytical outputs that serve business needs under strategic uncertainty.

I am proud of what I built, and grateful for the opportunity to build it.