NOVA IMS Information Management School

MDSAA

Master's Degree Program in

Data Science and Advanced Analytics

Machine Learning Operations Project

Afonso Gamito, number: 20240752

Gonçalo Pacheco, number: 20240695

Hugo Fonseca, number: 20240520

Nuno Nunes, number: 20240560

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

June, 2025

INDEX

1. Introduction

This project presents a modular and production-oriented machine learning workflow designed to simulate a real-world **MLOps environment**. The objective is to develop a scalable and maintainable system that supports model training, evaluation, monitoring, and deployment. The selected use case focuses on predicting the presence of **heart disease** based on clinical and diagnostic attributes.

Cardiovascular disease continues to be a leading global health concern, and the ability to identify individuals at risk through predictive modeling has significant implications for public health and medical decision-making. The dataset contains anonymized patient records with variables that are routinely collected during medical examinations, making the solution both **realistic** and transferable to clinical practice.

The **scope** of this project extends beyond model training to encompass the full lifecycle of machine learning in production. This includes structured data ingestion, rigorous data quality validation, systematic exploration and transformation of features, experiment tracking and model versioning, model explainability, and post-deployment monitoring for performance degradation and data drift. All components are developed in a modular architecture using **Kedro** to ensure reproducibility, traceability, and extensibility of the workflow.

We define success as achieving high model performance on F1-score, precision, and recall, since in healthcare applications both false positives and false negatives carry significant consequences.

2. Project Planning

The project was structured into iterative phases inspired by **agile sprints**, designed to reflect a reproducible and production-ready MLOps workflow ensuring scalability, maintainability, and alignment with real-world MLOps practices:

Development began with data ingestion and validation **(sprint 1),** focusing on schema consistency and quality assurance. Unit tests and assertions were used to identify missing values, incorrect data types, and outliers.

This was followed by exploratory data analysis and feature engineering **(sprint 2)**. Numerical variables were standardized, categorical features were encoded, and highly correlated or irrelevant attributes were removed.

Modeling and experimentation **(sprint 3)** was managed by Kedro pipelines to orchestrate the end-to-end modeling workflow, integrating MLflow to track experiments, hyperparameters, and performance metrics. This setup enabled reproducibility, streamlined comparisons across model iterations, and supported rapid experimentation while maintaining a clean separation between data, logic, and configuration.

Next, we focused on model evaluation and explainability **(sprint 4)**. Using SHAP analysis and feature importance techniques, we interpreted model predictions to ensure transparency and trustworthiness, which was particularly critical given the sensitivity of the healthcare domain. Additionally, error analysis was performed to identify patterns in misclassifications, guiding iterative improvements and reinforcing the model's robustness before deployment.

In the final phase which involved deployment considerations **(sprint 5)**, a batch model serving strategy was implemented, alongside a data drift detection mechanism to monitor shifts in input

distributions. Unit tests were developed for all core functions to verify correctness and stability across the pipeline.

3. Results and Conclusions from Data Exploration and Modelling

We performed a thorough data exploration on anonymized clinical records containing variables like age, sex, chest pain type, resting blood pressure, cholesterol, maximum heart rate, and electrocardiographic results.

Cholesterol and resting blood pressure had **skewed** distributions which were corrected through normalization. **Correlation** analysis guided the removal of redundant features. Stratified sampling was applied due to moderate **class imbalance**.

Feature engineering involved encoding categorical variables (chest pain type, ST slope) and scaling continuous features. Our exploratory visualizations showed clear class separation based on **age**, **cholesterol, and max heart rate.**

Three models were evaluated: Logistic Regression, Random Forest, and Gradient Boosting. **Random Forest** achieved the best performance:

Metric	Value
Precision	0.794
Recall	0.880
F1-Score	0.835

The **data drift** detection component compared the distributions of features in the new data against the training data. While most features remained stable, **cholesterol** showed significant drift with a p-value of 0.02. This indicates a meaningful shift in the cholesterol distribution, which could affect model performance and suggests that monitoring and potential retraining will be needed if drift persists.

SHAP analysis (Figure 1) illustrates that features like st_slope_1, chest_pain_type_4, and st_slope_2 had the **largest** positive or negative contributions, indicating their **strong** influence on prediction outcome.

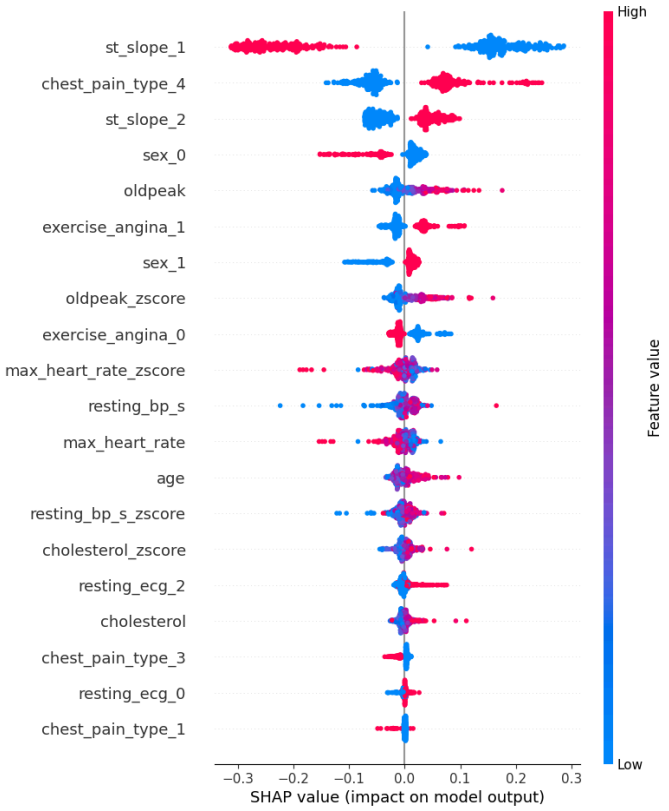

Figure 1: SHAP Analysis

4. Implementation Considerations

For production running, the pipeline would be modified to process more data with scalable tools like PySpark or Dask that support distributed processing and parallel computation.

Kedro offers good modularity and pipeline orchestration that is simple to extend to production environments with little refactoring. MLflow integration with experiment tracking and model registry would continue to be in production to allow ongoing monitoring and retraining cycles.

Risks are scalability bottlenecks with Pandas-based modules and potential latency in real-time prediction scenarios.

To mitigate these, we recommend taking 3 weeks to incorporate PySpark-based data pipelines and look into using FastAPI or similar technology for serving predictions via API endpoints.

Additional unit test coverage and CI/CD workflow integration would also contribute towards deployment readiness and reliability.

5. Packages Used

ipython>=8.10;

jupyterlab>=3.0

kedro-datasets>=3.0

kedro-viz>=6.7.0

kedro[jupyter]~=0.19.12

notebook

scikit-learn~=1.5.1

For a complete and detailed list of the packages and their versions used in this project, please refer to the '*requirements.txt'* file included in the project folder. This file contains all dependencies needed to reproduce the environment and execute the pipelines consistently.