Heart attack classification problem

MLOps Project

Group

Rodrigo Silva | 20230536

Nicolau Dulea | 20230544

Pedro Catarro | 20230463

Master's Degree in Data Science and Advanced Analytics

2023/2024

Table of Contents

1. Introduction

In this project, we aimed to develop a robust and scalable machine learning pipeline to predict the likelihood of clients experiencing a heart attack. For that we used the hospital data that were already processed and balance, which is unusual in this type of tasks.

Using Kedro for pipeline management, Visual Studio for development, Hospworks for feature store and MLflow for models versioning, we automated various stages of data processing, model training, and evaluation, ensuring a modular and maintainable codebase. The main objective was to create a predictive model that can identify early indications of a heart attack for a small hospital.

The success of this project is measured by the model's accuracy and f1-score and overall ability to generalize well to unseen data. We also emphasized model interpretability and explainability to ensure the predictions are understandable and actionable for medical professionals.

2. Project Planning

To effectively manage our heart attack prediction model project, we adopted an Agile methodology, which provided us with a structured yet flexible approach. Our workflow was organized using Jira, a project management tool, which allowed us to visually track and manage tasks across different stages of development. Our project tasks were broken down into distinct categories. The tasks were scheduled and moved across columns in our Jira board, representing different stages of progress:

- **TO DO**: This column included all the tasks that needed to be completed. Each task was clearly defined with specific objectives, and assigned to team members based on their expertise.

- **IN PROGRESS**: Once a team member began working on a task, it was moved to the 'In Progress' column. This indicated that the task was currently being worked on.

- **VALIDATION**: After a task was completed, it moved to the 'Validation' column. In this stage, the work done was reviewed by other group members to ensure quality.

- **DONE**: Tasks that passed the validation stage were moved to the 'Done' column. This indicated that the task had been thoroughly checked and was considered complete.

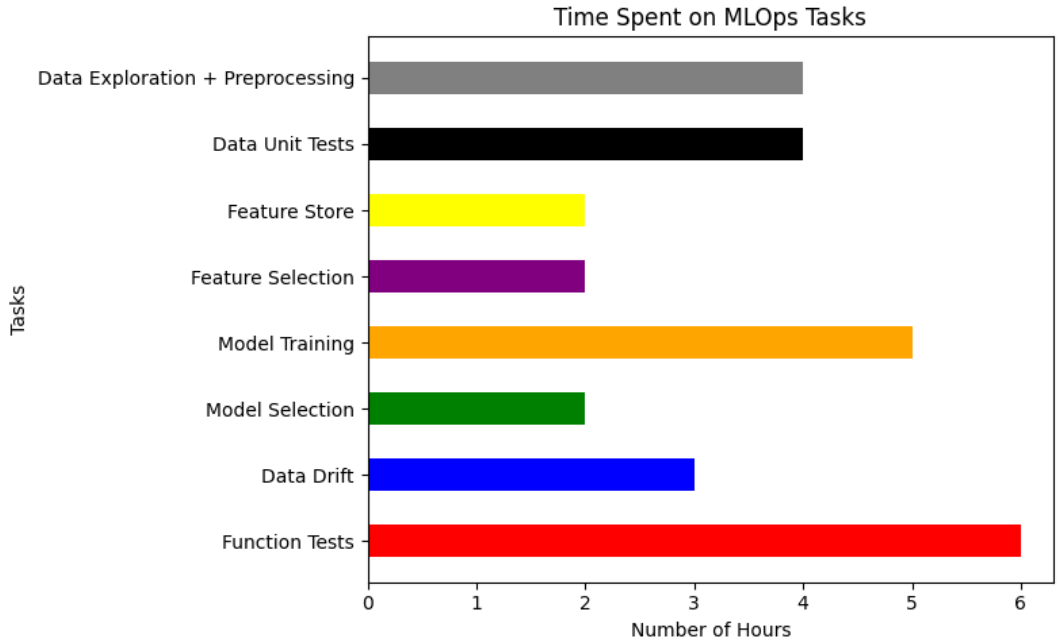

Figure 1. Time Spent on MLOps Task

3. Pipeline Description

3.1 Data Unit Tests Pipeline

In our data unit testing pipeline, the goal was to test if the values fell within the expected ranges (previously researched) and corresponded to the specified types. It is important to say that the some data that we received was already classified (Annex 1). If the data passed these tests, it was recorded as successful and ready for further processing. Any failures encountered during this process didn't stop development. Instead, they were recorded for review to identify and resolve data quality problems. We created the *Bronze Feature Group* as the first step in this data unit testing pipeline, serving as the initial storage layer for the raw data. [7] [8] [9]

3.2 Ingestion Pipeline

The data ingestion pipeline combined the data that we were receiving from the hospital with first existing data to create a single, comprehensive dataset.

3.3 Split Data Pipeline

The split data pipeline was used to divide the dataset into two distinct subsets. One subset served as the training data for machine learning models, while the other subset was employed to assess data drift and conceptual drift.

3.4 Preprocessing pipeline

Our preprocessing pipelines were designed to optimize the data for analysis and modeling. Since the data we had access to was already preprocessed, we did not encounter any missing values or duplicates.

In the preprocessing train pipeline, we analyzed the data using the clean function and created new features using the feature engineering function. One of the new features, *chol_zscore*, standardized cholesterol using the Z-score method, taking into account the patients' age and blood pressure. We also encoded categorical variables and scaled numerical features. In this pipeline, we added the *Silver Feature Group* (features after the clean function) and the *Gold Feature Group* (features after the feature engineering function) to our feature store and performed unit tests on the data quality with the new features.

The preprocess batch pipeline followed the same process, but for the data that we would use as a comparison for data drift.

3.5 Feature Selection Pipeline

Our feature selection pipeline focused on identifying the most valuable data points to improve our model's performance. We began by removing less informative data. Subsequently, we evaluated and ranked the remaining data based on their importance. Finally, we selected the top 10 most relevant features, ensuring that our model achieved both accuracy and efficiency.

3.6 Split Train Pipeline

The split train pipeline created a training set to teach models and a testing set to use for training the model.

3.7 Model Training Pipeline

The model train pipeline involved training a randomly chosen model from a predefined set. This model was then deployed into production. Subsequently, the performance of the deployed model was evaluated using accuracy and F1-score metrics.

3.8 Model Selection Pipeline

Our model selection pipeline aimed to identify the optimal machine learning model for our data. We started by training several candidate models using the best features identified earlier. To optimize each model, we employed GridSearchCV, but with a limited set of values for testing purposes. The models' performance was then evaluated using accuracy and F1-score. The best-performing model was subsequently compared to the existing production model. [4]

3.9 Data Drift Pipeline

The objective was to detect data drift between a reference dataset and an analysis dataset using both univariate and multivariate methods, to ensure the reliability and consistency of machine learning models over time. [1] [2] [3]

The first step involved analyzing the drift in categorical columns to detect any significant changes in the distribution of categorical features between the reference and analysis datasets. To achieve this, Jensen-Shannon divergence was used as the measure of drift, and a constant threshold was set to identify significant drift.

Next, the continuous columns were analyzed for drift. Evidently AI's KS test was utilized for statistical significance testing of drift in continuous features.

The final step was to perform a multivariate drift analysis where the purpose was to capture and quantify drift that affects the relationships between multiple features simultaneously. Principal Component Analysis (PCA) was used to reduce the dimensionality and identify drift in the multivariate space.

3.10 Model Predict Pipeline

In the prediction model pipeline, we made predictions with our production model and we used permutation importance technique to evaluate how much each feature contributed to the model's predictions. The higher the importance score for a feature, the more impact it had on the model's performance.

4. Project Success Metrics

Given the serious medical implications of our predictions, we focused on two key metrics: Accuracy and the F1-score. Accuracy because it measures the overall correctness of our model's predictions by indicating the proportion of all correct results among the total number of cases examined. It provides a broad measure of how often the model correctly identifies whether a patient is at risk of a heart attack or not. While Accuracy can be a general performance indicator, it's crucial to recognize that our dataset will be imbalanced in the future, meaning there will be a significant difference between the number of patients who are and are not expected to experience heart attacks. In healthcare, this imbalance can be particularly problematic. False negatives (missing a true heart attack) can lead to delayed or missed treatment, while false positives (incorrectly predicting a heart attack) can cause unnecessary anxiety and procedures. For this reason we employ the F1-score, which considers both precision (correctly identifying patients with heart attacks) and recall (identifying all true cases of heart attacks). This balance between precision and recall makes F1-score a more suitable metric for evaluating our model's effectiveness in this specific application.

5. Conclusion and Further Work

By evaluating the performance of our machine learning models - Decision Tree, Random Forest, and Logistic Regression - we can gain insights into their effectiveness for real-world deployment. The Decision Tree model was initially chosen as our production model and therefore the one predicting who is most likely to have a heart attack. However, hyperparameter tuning revealed superior performance in both accuracy and F1-score for the Random Forest model. This necessitates a swift transition to deploying the Random Forest model in production (Annex 2).

While our initial model development has been successful, transitioning to a global stage presents new challenges. One key barrier involves ensuring our models are trained on high-quality data that accurately reflects the diverse realities of a global audience. More Real-world data can be messy and require rigorous cleaning and preprocessing to avoid biases and ensure generalizability. To streamline this process and facilitate scaling, we can leverage cloud-based tools and workflows.

These tools automate tasks and simplify data management, making the entire process more efficient. In addition to cloud-based tools, exploring frameworks like Apache Spark can further enhance our machine learning workflow, particularly for handling massive datasets. PySpark facilitates efficient data preprocessing, feature engineering, and model training at scale on distributed computing clusters. This can significantly improve processing speed, especially when dealing with the potentially vast amount of data required for global deployment. Finally,

containerization using technologies like Docker or Kubernetes ensures seamless operation of our model across different environments, allowing for easy deployment anywhere in the world. By understanding these potential pitfalls and implementing these solutions, we can significantly increase our chances of successfully industrializing our model.

6. References

[1] Perrakis, N. (2023, July 13). *Data Drift Detection for continuous variables: Exploring Kolmogorov-Smirnov test*. NannyML Blog. https://www.nannyml.com/blog/practical-data-drift-2

[2] *Inferring concept drift without labeled data*. (n.d.). https://concept-drift.fastforwardlabs.com/

[3] De Pontes Adachi, F. (2023, March 14). Understanding Kolmogorov-Smirnov (KS) tests for data drift on profiled data. https://towardsdatascience.com/understanding-kolmogorov-smirnov-ks-tests-for-data-drift-on-profiled-data-5c8317796f78

[4] Feurer, M., & Hutter, F. (2019). Hyperparameter Optimization. In *The Springer series on challenges in machine learning* (pp. 3–33). https://doi.org/10.1007/978-3-030-05318-5_1

[5] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

[6] Gavish, L. (2023, January 5). *4 reasons why production machine learning fails — and how to fix it*. Monte Carlo Data. https://www.montecarlodata.com/blog-why-production-machine-learning-fails-and-how-to-fix-it/

[7] MacGill, M. (2024, January 18). *What is a normal blood pressure reading?* https://www.medicalnewstoday.com/articles/270644#ranges

[8] Lee, Y., & Siddiqui, W. J. (2023, July 24). *Cholesterol levels*. StatPearls - NCBI Bookshelf. https://www.ncbi.nlm.nih.gov/books/NBK542294/#:~:text=Optimal%3A%20less%20than%20100%20mg,160%20to%20189%20mg%2FdL

[9] *Anatomy and circulation of the heart*. (2023, April 24). WebMD. https://www.webmd.com/heart-disease/high-cholesterol-healthy-heart

7. Annexes

7.1. Annex 1

Features	Description	Expected Values
age	Age of the patient	Between 0 and 115
sex	Sex of the patient	0 or 1
cp	Chest pain type: 0- Typical Angina 1- Atypical Angina 2- Non-anginal Pain 3- Asymptomatic	3, 2, 1, 0
trtbps	Resting blood pressure (mm Hg)	Between 80 and 196
chol	Cholesterol (mg/dl)	Between 30 and 600
fbs	Fasting blood sugar > 120 mg/dl: 0- True 1- False	0 or 1
restecg	Resting electrocardiographic results: 0- Normal 1- ST-T wave abnormality 2- Left ventricular hypertrophy	0, 1, 2
thalachh	Maximum heart rate achieved	Between 50 and 290
exng	Exercise induced angina: 0- Yes 1- No	0 or 1
oldpeak	ST depression induced by exercise	Between 0 and 7
slp	Slope of the peak exercise ST segment	0, 1, 2
caa	Number of major vessels (0-4) colored by fluoroscopy	0, 1, 2, 3, 4
thall	Thallium Stress Test result	0, 1, 2, 3
output	Target variable indicating presence of heart disease	0 or 1

Table 1. Features, their meaning and data unit tests

7.2 Annex 2

Model	Accuracy	F1-score
Decision Tree	0.9626	0.9027
Random Forest	0.9380	0.9381
Logistic Regression	0.8495	0.8496

Table 2. Models and respective scores