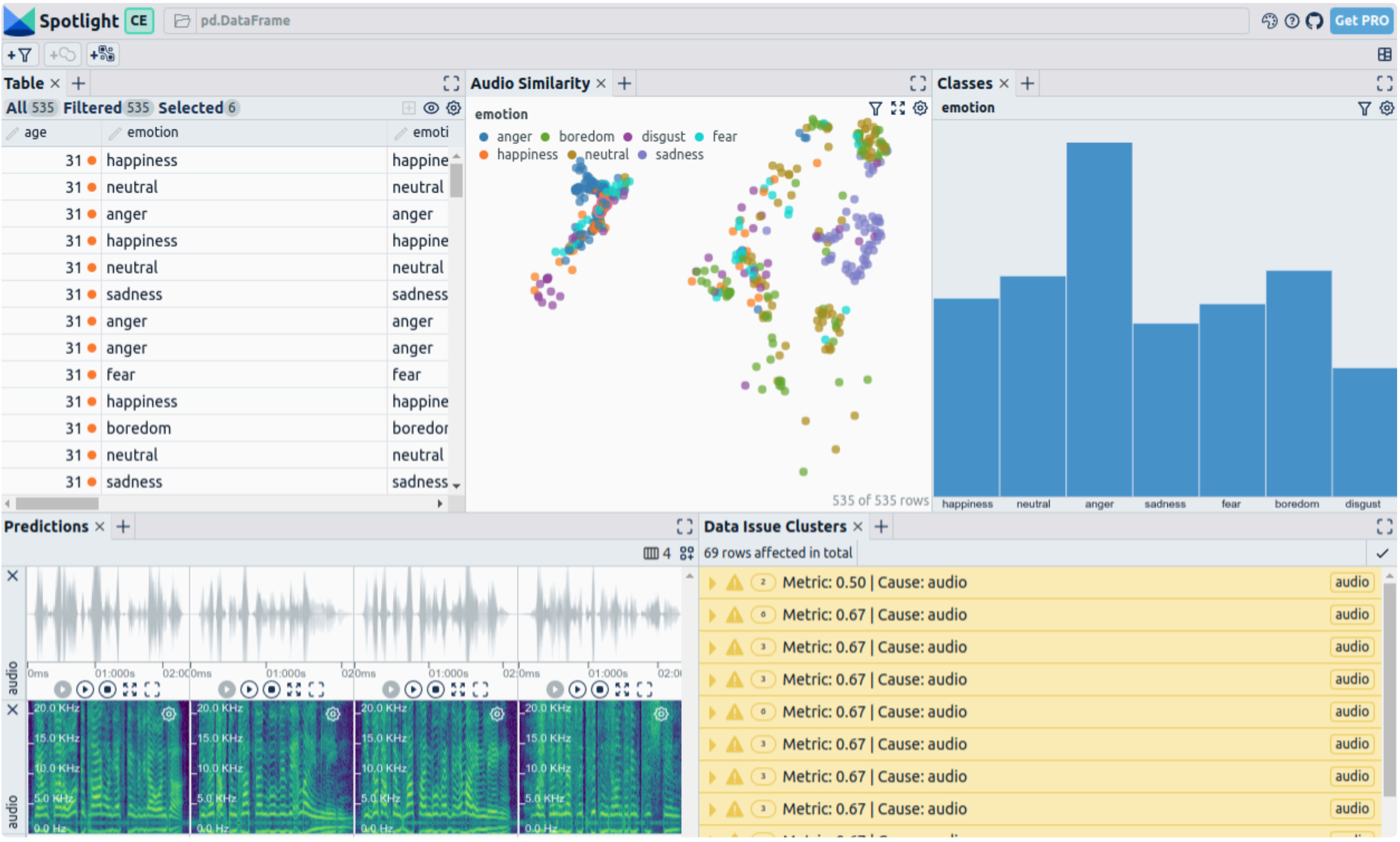
9 REASONS

Why ML-Based Acoustics Testing Projects Fail

Introduction

Analyzing test data can be quite time-consuming and error-prone. Machine learning systems are an excellent fit for addressing this challenge. They excel at automating cognitive tasks, such as scanning audio files for anomalies or providing insights that can be leveraged for interactive data analysis. Despite this, many companies have not yet begun to leverage this potential. This document thus aims to show the most common errors we encountered in our customer projects.



ML-based acoustics projects are challenging. Using the right project setup and tools can significantly reduce the risk of failure. (Image created using Spotlight)

The 9 Reasons

1. Lacking Utilization of Domain Expertise

Of course, the foundation for defining a good use case is someone who knows your processes well, can calculate the profitability of a use case, and knows how the data is generated and how it is used. If you don't include these people early on, you are likely to fail. Also, by all means, avoid trying to press solutions you find technically appealing on your users. Of course, it is your job to guide them and make suggestions, but keep the solution space wide at first before jumping to conclusions.

2. Lack of Expertise in Machine Learning and Software Development

Complementary to the lacking utilization of domain expertise you also should watch out for missing expertise in machine learning. Even before starting a project deficits in these areas can be a true blocker. It is true, that domain expertise is super important for realizing your project, but you also need people who are fluent in using Machine Learning-related technologies and can steer you through the sheer endless amount of choices. So, find the right balance when creating your team.

3. Misalignment Between Domain Experts and Technical Teams

Especially in the early stages of the project, a typical problem I have witnessed is a misalignment between domain experts and technical teams. This is mostly driven by waterfall-like project models with small to no communication between developers and domain experts. To avoid this trap, do not set up your project in that way and aim for a quick delivery of the first deployed version of the application.

4. Initial Model Ineffectiveness in Preliminary Studies

Many times I have seen projects that were further and further delayed because the preliminary studies did not yield the wanted results. I feel like it is reasonable to check for this BUT let small metric differences not stop your project. The final solution will not be only your machine learning model. After all, there is much more to a good system doing the job than a specific machine learning method. So instead, move behind preliminary studies that are pure ML evaluations quickly, think about downstream metrics instead of low-level ML metrics, and build a first prototype quickly that does the job!

5. Resistance to Change in Workflows

The misalignment of domain experts and technical teams can also often contribute to a resistance to change existing workflows. This is natural but should be carefully managed. The technical team should be able to navigate through this and find good ways to adapt to user input while also presenting alternative workflows that the domain experts couldn't have imagined yet.

6. Legacy Systems for Test Data Analysis and Data Storage

Many times, there are legacy systems in place and it is hard to even find the right components to start modernizing. Also, it might be not at all clear how the new technologies can be applied to, e.g., legacy data storage. In my experience, it can help to just not let this be a too large blocker before even starting. E.g., in case of the data storage, probably plan to build a parallel data storage solution that can later replace the legacy data solution once it gains trust.

7. Data Availability and Quality Issues

An old, but still popular blocker is to stop moving further in a project once the opinion that there is not enough data or the data quality is too bad has manifested itself. Of course, this can be a blocker, but oftentimes you should move on, build a system that does an okay job with the data that is available, and improve the system iteratively once you have the data from the production system. Also, there is a lot of tooling around improving data quality and building models with small data. Keywords Data-centric AI and Foundation Models.

8. Distrust in Machine Learning Models

Many times, despite all these preliminary studies, there will be distrust towards the machine learning model. This will also not go away by iterating on metrics. Instead, bring your machine learning model into production quickly (potentially in a prototype) and let the users work with it. It is the only way that worked for me! Also, there are mechanisms in the areas of XAI/Interpretability that can help facilitate user trust, however, they cannot work without letting people use what you have built.

9. Regulatory and Compliance Constraints

Many times projects are simply not done or bent in a way that they will not provide value anymore because the environment is too overregulated and people can not see how an elegant solution can be implemented in there. My advice here is to not always go to the extreme of being extremely careful. Navigate this trade-off with care, but you should also find ways to drive change beyond what is seen as possible in the beginning.

Contact Information

Daniel Klitzke

Machine Learning Engineer

www.renumics.com | daniel.klitzke@renumics.com