

Designing an expert system for advanced flight data monitoring

Lucas Coelho e Silva*,¹

Aeronautics Institute of Technology, Pça Marechal Eduardo Gomes, 50, São José dos Campos, São Paulo, Brazil - 12228-615

ARTICLE INFO

Keywords:

Aviation safety
Expert system
Machine learning

ABSTRACT

The utmost priority of the civil aviation system is operational safety. Challenging safety goals and an increasingly complex flight operations system fuel an enduring effort stream toward developments that unveil safety hazards proactively. In the airline industry, Flight Operations Quality Assurance (FOQA) programs are the de-facto standard of safety monitoring and analysis and are widely discussed by flight operations stakeholders, such as aircraft operators and civil aviation authorities. However, it relies on the specification of thresholds of previously defined safety events, the latent risks, and the near-exceedance situations that did not lead to an event go unnoticed. This leads to a hazard detection gap and creates the need for alternative modeling approaches. While current literature discusses various methods for anomaly detection within the flight data context, the application of these techniques to the construction of a practical expert system remains unexplored. This paper describes the construction of an expert system that enables the execution of an advanced FOQA program that relies on the construction of machine learning models as a complementary tool to event-based processes traditionally oriented towards threshold exceedance mechanisms. It covers the construction of an end-to-end system for operationalizing safety analysis techniques and incorporating user feedback. For the learning process, we apply a Positive-Unlabeled (PU) learning mechanism variant that accounts for positively labeled flights (known to present safety events), negatively labeled flights (known not to present safety events), and unlabeled flights. We evaluate the model performance metrics via a Repeated Stratified K-Fold cross-validation process and perform model selection via Bayesian analysis.

1. Introduction

Operational safety is core to the civil aviation system, being the utmost priority of flight operations stakeholders such as airlines, Air Traffic Management (ATM), and aeronautical infrastructure operators. While the safety-by-design standards of aviation have been continuously leading to the decrease of fatal accident rates over the years (ICAO, 2022), there is an enduring effort stream toward developments that proactively, not only reactively, unveil safety hazards in an ever-increasingly complex flight operations system (Oster, Strong and Zorn, 2013).

In addition, challenging safety goals such as the zero-fatality target in commercial operations from 2030 onward, proposed by the International Civil Aviation Organization (ICAO), drive research efforts and the search for improved safety management policies, models, and methods (ICAO, 2019).

Current practice for airline safety analysis comprises the scenario-oriented approaches (Höhndorf, 2018) of Flight Operations Quality Assurance (FOQA) programs, also known as Flight Data Monitoring (FDM), and regards the comparison of sensor data with pre-established thresholds. For that reason, it is also known as exceedance detection. It is this approach that the Federal Aviation Administration (FAA), for example, details in the Advisory Circular (AC) 120-82 - Flight Operational Quality Assurance, aimed at presenting means of developing and implementing an acceptable FOQA program for the authority (FAA, 2004).

FOQA programs rely on flight operations data, composed primarily of aircraft performance data from onboard sensors (flight data recorders). This core information can be complemented with other sources, e.g., safety reports, flight plans, weather data, and workload information, enabling the analysis to be as thorough as one desires.

Because current practice relies on the specification of thresholds of previously defined safety events, the latent risks and the near-exceedance situations that did not lead to an event go unnoticed. Additionally, only what is sought, known, and searched for is captured and ends up being described during the data analysis process, i.e., risks that were

*Corresponding author

✉ coelho@ita.br/lucascoelhosilva@gmail.com (L. Coelho e Silva)

ORCID(s): 0000-0002-6499-912X (L. Coelho e Silva)

 <https://www.linkedin.com/profile/view?id=lucascoelho> (L. Coelho e Silva)

¹Declarations of interest: none.

not even mapped also go undetected. In this sense, there are opportunities for the application of modeling approaches that go beyond the standard practice. While current literature discusses various methods for anomaly detection within the flight data context, the application of these techniques to the construction of a practical expert system remains unexplored.

This paper describes the construction of an expert system that enables the execution of an advanced FOQA program that relies on the construction of machine learning models as a complementary tool to event-based processes traditionally oriented towards threshold exceedance mechanisms. In this sense, we perform offline identification of anomalous airplane performance operations, while focusing on the landing phase of the flight.

The structure of this paper is the following: Section 2 discusses aviation safety and flight data monitoring in airline operations by reviewing the related literature. Section 3 details the methodological approach, presenting the dataset used in this research, the applicable machine learning methods, and software design principles. Section 4 presents the results and discussions, and Section 5 addresses the findings.

2. Background and literature review

In the airline industry, FOQA programs are the de-facto standard of safety monitoring and analysis and are widely discussed by flight operations stakeholders, such as aircraft operators and civil aviation authorities (FAA, 2004). It relies on flight operations data, composed primarily of aircraft performance data from onboard sensors (flight data recorders), for the monitoring of specified safety events. More generally, FOQA programs can be thought of as a piece of an overarching data-driven safety process, from which there may stem the development of novel safety policies with new operational procedures, training programs to prevent non-ideal operations from happening again, models for distinguishing between normal and anomalous behavior, and so forth.

Because current practice relies on specifying thresholds of previously defined safety events, the latent risks and the near-exceedance situations that did not lead to an event go unnoticed. Additionally, only what is sought, known, and searched for is captured and ends up being described during the data analysis process, i.e., risks that were not even mapped also go undetected. This leads to a hazard detection gap and creates the need for alternative modeling approaches. According to Coelho e Silva and Murça (2023), the current practice of routine data analysis and the machine learning approach are complementary, rather than competing – with the current practice being able to provide labeled data for the modeling process.

In the literature, there are several data-driven and machine-learning-oriented safety applications. For instance, Barry (2021) applied Bayesian Networks to flight data to estimate runway veer-off risk under different scenarios, enabling the assessment of inquiries such as what would be the airport at which a runway excursion is most likely, for a given airline. Das, Matthews, Srivastava and Oza (2010) proposed an anomaly detection approach based on Multiple Kernel Learning (MKAD), focused on detecting anomalies in multivariable, high-dimensional data. Gorinevsky, Matthews and Martin (2012) developed a method for detecting anomalies based on a regression model, aiming at assisting FOQA processes. The model was tested on flight data and was able to identify anomalous records. However, the anomalies found were more related to faulty sensor readings than safety anomalies per se. Li and Hansman (2013) developed an anomaly detection approach based on clustering algorithms and expert review. The authors developed two algorithms: ClusterAD-Flight and ClusterAD-Data Sample, based on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester, Kriegl, Sander and Xu, 1996) and Gaussian Mixture Models (GMM) approaches, respectively. As the expert review was one of the research pillars, it also focused on developing visualization tools. As per Li and Hansman (2013), the two algorithms were capable of identifying operationally relevant anomalies, surpassing current methods. Finally, Coelho e Silva and Murça (2023) present a general data analytics framework applied to flight operations.

Nevertheless, while the literature discusses methods, algorithms, and frameworks for anomaly detection and safety analysis with flight data, it lacks the construction of an end-to-end system for operationalizing these techniques and incorporating user feedback. The closest work is the precursor work of Amidan and Ferryman (2005). In this study, the authors develop a software called “Morning Report” to find typical patterns and atypical events in flight data. While this approach proved to be limited, with the chosen time series representation failing to capture relevant signals in the data (Li and Hansman, 2013), this study approached the problem in a way that would allow the specialist to focus on atypical flights, a concept used until today, while also introducing the idea of applying clustering algorithms in the context of anomaly detection in aviation.

Apart from Amidan and Ferryman (2005), the application of these techniques in the construction of a practical expert system remains unexplored. This paper addresses this gap with the construction of an expert system that enables the execution of an advanced FOQA program that relies on the construction of machine learning models as a complementary tool to event-based processes traditionally oriented towards threshold exceedance mechanisms.

3. Methodological approach

This study explores the construction of an expert system that enables the execution of an advanced FOQA program that relies on the construction of machine learning models as a complementary tool to event-based processes traditionally oriented towards threshold exceedance mechanisms. For the software aspects and modeling principles of the project, we follow the approach proposed by Verri (2024), based on the principles of modularization, version control, continuous integration and deployment, reports as deliverables, quantitative goals setup, measurement of custom metrics based on the project goals, model stability and performance variance reporting, adequate user interface nomenclature, production model performance monitoring, and appropriate back-end design. For the FOQA aspects of this project, we build on the framework proposed by Coelho e Silva and Murça (2023).

The end users of this system are airline safety experts. In this sense, we design a system in which feedback mechanism is a core requirement. The user should be able to identify whether a given flight contains anomalous operations, and the user should be able to report whenever a flight is misclassified.

3.1. Dataset

The Sample Flight Data data set was used, available in NASA's Discovery in Aeronautics Systems Health (DASHlink) (2012). It is a publicly available data set containing flight data of a single regional jet model recorded during commercial operations over three years. The complete data set comprises more than 180,000 files, each containing records of a single flight or ground operation, grouped in a series of compressed master files.

Each flight data file presents flight information in a time series, featuring more than 100 parameters collected by several aircraft sensors. We evaluated the landing operations in the following compressed master files: Tail_652_1, Tail_652_2, Tail_652_3, and Tail_652_4, a subset of nearly 2000 flights.

3.2. System design

For replicating the real-world process in which flights would be continuously uploaded to the system, we design an event-based data processing pipeline in which the periodic processing of uploaded flights on the target storage feeds a FOQA pipeline that uses the rule engine to identify safety events and stores the results. In terms of cloud providers, we use the services available on the Google Cloud Platform (GCP).

Figure 1 displays the system's cloud architecture. There are two layers in this architecture: the flight processing layer, with the data processing mechanisms and FOQA pipeline, and the modeling layer, with the model building and model inference subdivisions.

The process starts with the upload of the flights to the system. These flights compose a flight pool of data to be processed. The periodic execution of the rule engine feeds the FOQA tables on the database. These tables can then be collected and versioned by the modeling layer, in which a model training pipeline is executed. This results in a model artifact that can be used for operating on new flights and providing an event propensity score that represents how likely a flight is to present a safety event.

The event propensity scores and the regular FOQA rules outputs are then presented to the user via a streamlit dashboard, where the user can provide feedback. Finally, user feedback is also used for the construction of the subsequent models.

3.2.1. Software fronts

The software is divided into two fronts: *openfdm*, a FOQA data processing package, and *advanced-foqa*, responsible for the user interface, execution of the rule engine on the cloud, machine learning modeling, and user tasks.

openfdm is a production-ready Python package for running a Flight Data Monitoring (FDM)/Flight Operations Quality Assurance (FOQA) program, including the identification of safety events. It provides an implementation of an FDM pipeline for batch processing of safety events across multiple flights; a rule engine with base events and an extensible software architecture; a standardized flight dataframe and parameters model; and extensible connectors for loading flights and saving outputs of business rules.

The *advanced-foqa* module is further subdivided into the following:

Advanced Flight Operational Quality Assurance

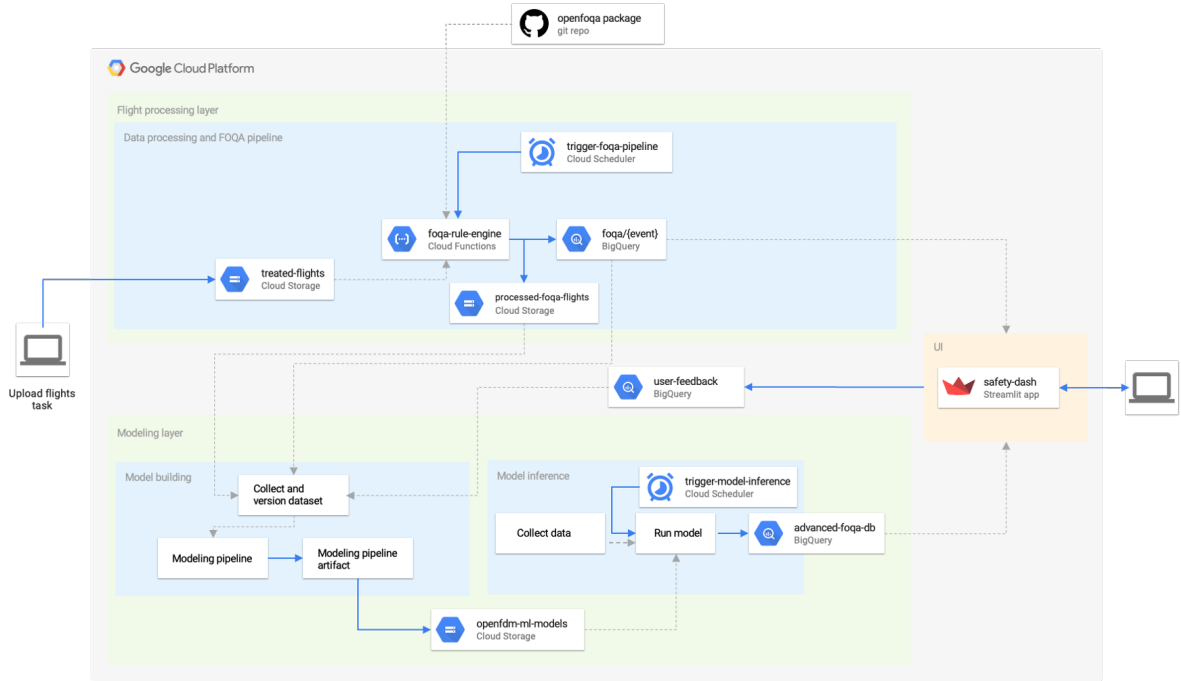


Figure 1: Advanced FOQA cloud architecture.

- apps/safety dash: Streamlit app for displaying FOQA events, flights, and model results.
- cloud/foqa-rule-engine: GCP's Cloud Function that instantiates an openfdm's pipeline and runs it on the applicable flights.
- ds-modeling: machine learning modeling process, including the foqads package.
- tasks: user tasks such as uploading flights.

3.2.2. Database design

The system relies on SQL-compatible tables for storing the FOQA event outputs as well as versioned training data sets and flight labels. We store this information using BigQuery tables divided into two domains: *foqa*, and *ds_modeling*. The *foqa* data set contains Tables that store the results from each FOQA rule within the processing pipeline. On the other hand, the *ds_modeling_domain* data set stores tables versioned and used during the model training process (example: labeled flights, training data). Figure 2 displays the data set and table structure designed for the system.

3.3. Current practice as a source of features and labels

With the goal of developing machine learning models as a complementary tool to the event-based processes, we include in the system design a mechanism for reproducing the current practice of exceedance detection. For that, we evaluate three events commonly monitored for the landing phase of the flight: hard landing, landing in a crab, and bounced landing. We used the recommended parameters and event definitions of AC 120-82 (FAA, 2004), reproduced in Table 1. The parameters calculated by this pipeline can be used as features for the modeling process.

3.4. User interface

For the user interface design, the goal is to provide ease of access and familiarity to the user, already accustomed to safety tools. Therefore, the modeling results must be easily accessible via a usual FOQA application design approach.

As discussed, the specialist feedback is crucial for this system. Thus, we develop a dedicated user interface component so a specialist can provide feedback on a specified number of flights. For this system, the number of



Figure 2: BigQuery table structure.

Table 1
FOQA event parameters and definitions (FAA, 2004).

Event name	Event description	Parameters	Event definition
Landing in a crab	An event to detect failure to align aircraft with the runway at touchdown.	Heading; Calibrated Airspeed (CAS).	Δ Heading at Touchdown vs. Average Heading until CAS = 60 knots.
Hard landing	An event that measures excessive G-force at touchdown, indicating a hard landing.	Air/Ground Switch; Vertical Acceleration.	Air/Ground = Ground, Vertical Acceleration > x G
Bounced landing	An event that measures excessive G-force at touchdown followed by a second excessive G-force, indicating a bounced, hard landing.	Air/Ground Switch; Vertical Acceleration.	Air/Ground = Ground, Vertical Acceleration > x G, followed by second Vertical Acceleration > x G within 20 seconds of first touchdown.

flights for feedback is defined as 10. Whenever the specialist uses the dedicated page, it is mandatory that they provide feedback for every flight available. Nevertheless, because reporting needs for specific flights arise in customary system operation, we also design a workflow so the user can provide feedback on individual flights.

3.5. Experimental plan and machine learning models

For the machine learning models, the goal is to provide a complementary tool to the event-based FOQA pipeline that identifies safety events based on exceedance mechanisms. For this, the chosen model must be operationalized in a way that, when operating on new flights, it provides an event propensity score that represents how likely a flight is to present a safety event.

For the learning process, we design a Positive-Unlabeled (PU) learning mechanism variant that takes into consideration positively labeled flights (known to present safety events), negatively labeled flights (known not to present safety events), and unlabeled flights. For the 1972 flights, we have 15 positively labeled ones, and 425 negatively labeled flights. For capacity considerations, we assume 280 to be the number of weekly flights, and a capacity to analyze $k = 10$ flagged flights. In this sense, what the model needs to accomplish is the correct sorting of flights based on their propensity to present a safety event. Therefore, we evaluate the model performance metrics considering the $@k$ values. For the metrics purposes, however, we consider $k = 20$, due to the high imbalance of labeled instances on the dataset,

Our experimental plan consists of performing a Repeated Stratified K-Fold cross-validation process, made compatible with the PU learning approach, with 10 repetitions of 3 folds. In terms of scores, we assess precision@k, recall@k, False-Positive-Rate@k, and the precision of labeled instances only.

For features, we used the calculated data from the FOQA pipeline: roll rate, pitch rate, vertical acceleration, number of bounces, difference in heading, heading at touchdown, heading during the landing roll, and maximum vertical acceleration; with a within-the-pipeline imputation of missing values using the mean value.

We perform model selection via Bayesian evaluation of each metric, taking a decision tree as the baseline and the following models as candidates: Random Forest, Gaussian Naive Bayes, XGBoost, and Logistic Regression.

Finally, we treat the models as versioned software. Every training data set is stored on versioned BigQuery tables and the modeling code is part of the software modules described in Section 3.2.1.

4. Results and discussion

4.1. User interface

As discussed in Section 3.4, one of the pillars of the user interface design approach is to guarantee ease of access and familiarity to the user, already accustomed to safety tools. Thus, it is a system requirement that the modeling results must be easily accessible via a usual FOQA application workflow. Figure 3 displays the main page of the Advanced FOQA dashboard, with summary metrics of the analyzed flights in a classic FOQA-like manner. Additionally, event-specific data is available, such as the hard landing data shown by Figure 4.

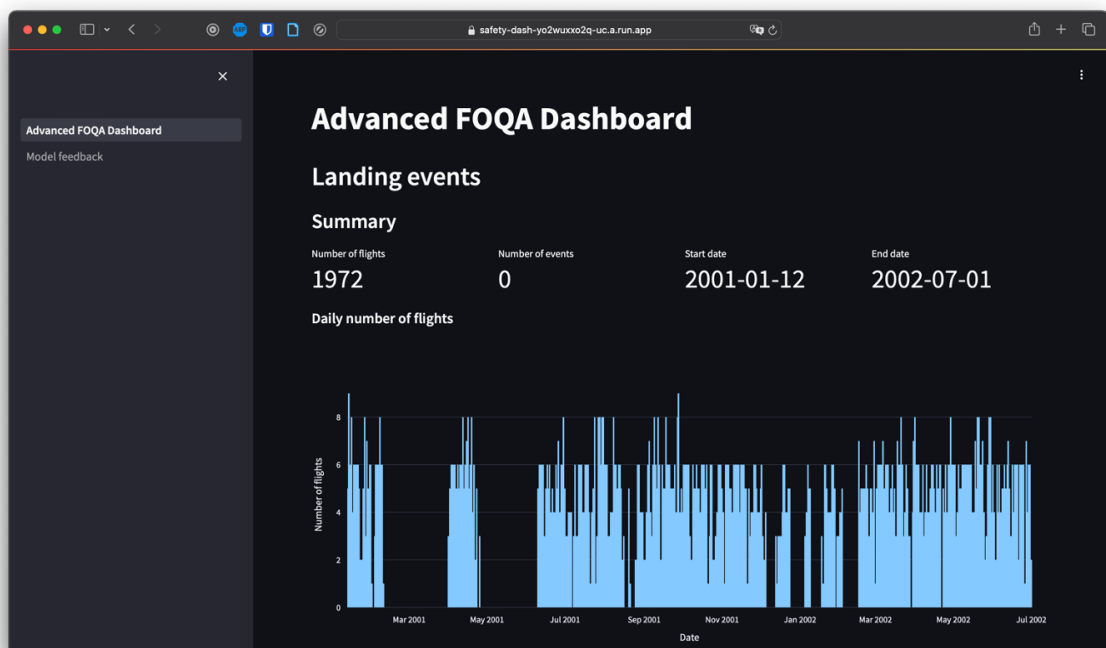


Figure 3: Main page of the Advanced FOQA dashboard.



Figure 4: Hard landing data within the Advanced FOQA dashboard.

Furthermore, specialist feedback is crucial for this system, and our approach provides two feedback mechanisms. First, because reporting needs for specific flights arise in customary system operation, we design a workflow so the user can provide feedback on individual flights, as illustrated by Figure 5. In this approach, the user can select one or more flights available in the dashboard and report that ‘No safety events were identified in the flight’, or that the ‘Flight may contain a safety event.’ For ease of access and consistency, the event propensity scores from the model are also shown on this section of the page.

In addition, we develop a second feedback mechanism on a dedicated user interface component so a specialist can provide feedback on a system-defined number of flights. For this system, the number of flights for feedback is defined as 10. Whenever the specialist uses the dedicated page, they must provide feedback for every flight available. Figure 6 displays this second feedback mechanism, along the same report categories of ‘No safety events were identified in the flight’ and ‘Flight may contain a safety event’ available on the main page.

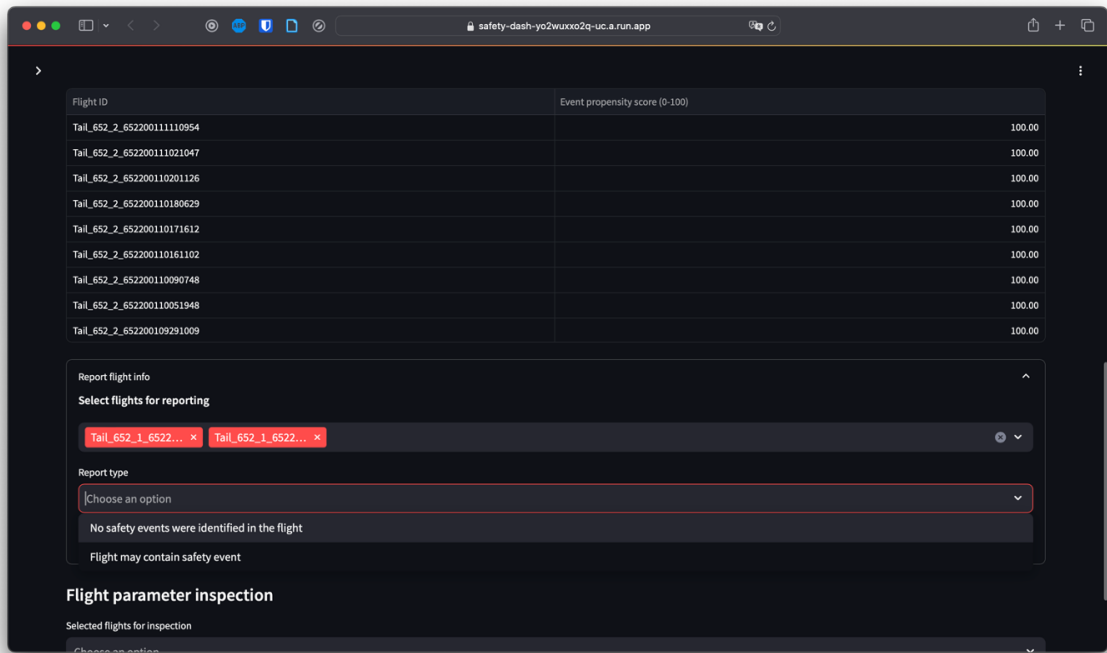


Figure 5: Feedback on user-defined flights.



Figure 6: Specialist feedback page: report categories.

Upon submission, the user feedback feeds a BigQuery table in which the feedback for each flight is recorded, along with the feedback source (main page or dedicated feedback page). This information can be used to improve labels on subsequent model training processes.

user_feedback					QUERY		SHARE	COPY	SNAPSHOT	DELETE	EXPORT
SCHEMA	DETAILS	PREVIEW	LINEAGE	DATA PROFILE	DATA QUALITY						
Row	flight_id	analyst_feedback	creation_timestamp_ms	source							
1	Tail_652_1_652200101121218	No safety events were identified in the flight	1718632427354	foqa_page							
2	Tail_652_1_652200101121444	No safety events were identified in the flight	1718632427354	foqa_page							
3	Tail_652_2_652200111110954	Flight may contain a safety event	1718632584411	feedback_page							
4	Tail_652_2_652200110201126	Flight may contain a safety event	1718632584411	feedback_page							
5	Tail_652_2_652200110180629	Flight may contain a safety event	1718632584411	feedback_page							
6	Tail_652_2_652200110090748	Flight may contain a safety event	1718632584411	feedback_page							
7	Tail_652_2_652200109291009	Flight may contain a safety event	1718632584411	feedback_page							
8	Tail_652_2_652200111021047	No safety events were identified in the flight	1718632584411	feedback_page							
9	Tail_652_2_652200110171612	No safety events were identified in the flight	1718632584411	feedback_page							
10	Tail_652_2_652200110161102	No safety events were identified in the flight	1718632584411	feedback_page							
11	Tail_652_2_652200110051948	No safety events were identified in the flight	1718632584411	feedback_page							
12	Tail_652_2_652200109210841	No safety events were identified in the flight	1718632584411	feedback_page							

Figure 7: Feedback table in BigQuery.

4.2. Machine learning models

For the learning process, as mentioned in Section 3.5, we apply a PU-learning mechanism variant that takes into consideration positively labeled flights (known to present safety events), negatively labeled flights (known not to present safety events), and unlabeled flights. We evaluate the model performance metrics of precision@k, recall@k, False-Positive-Rate@k, and the precision of labeled instances only, i.e., precision, recall, and FPR considering the top @k values in terms of classification scores. For the metrics purposes, we set $k = 20$, double of the expected k during model operation. We calculate the performance metrics via a Repeated Stratified K-Fold cross-validation process, made compatible with the PU learning approach, with 10 repetitions of 3 folds (i.e., 30 values for each metric). Finally, for model selection, we perform Bayesian evaluation (Benavoli, Corani, Demšar and Zaffalon, 2017) of each metric, taking a decision tree as the baseline and the following models as candidates: Random Forest, Gaussian Naive Bayes, XGBoost, and Logistic Regression.

Table 2 presents the mean values for each performance metric, for each model. These numbers are complemented by the posteriors of the Bayesian correlated t-tests of the precision@k and recall@k metrics, shown by Figures 8 and 9, respectively. Finally, Figures 10, 11, 12, and 13 present the boxplots of the differences to the baseline of the metrics precision@k, recall@k, precision (labeled instances), and FPR@k, respectively.

The Naive Bayes classifier presents the highest scores for every metric. In addition, it presents a 98% probability of performing better than the baseline Decision Tree model for the metrics of precision@k and recall@k, while displaying the same average values (perfect scores) of the baseline for the metrics of precision (labeled instances), and FPR@k. While the other models also present high probabilities of performing better than the baseline in precision@k and recall@k, the Naive Bayes classifier presents overall better numbers. Therefore, from this analysis, we select the Naive Bayes classifier as the production model, thus trained with the entire data set and operationalized.

Table 2

Average performance metric values for each contemplated model.

Model	Precision@k	Recall@k	Precision (labeled)	FPR@k
Baseline (Decision Tree)	2.3%	10.1%	100%	0%
Random Forest	5.7%	24.2%	100%	0%
Naive Bayes	13.6%	58.2%	100%	0%
XGBoost	4.0%	17.0%	92.5%	0%
Logistic regression	7.8%	33.5%	80%	0%

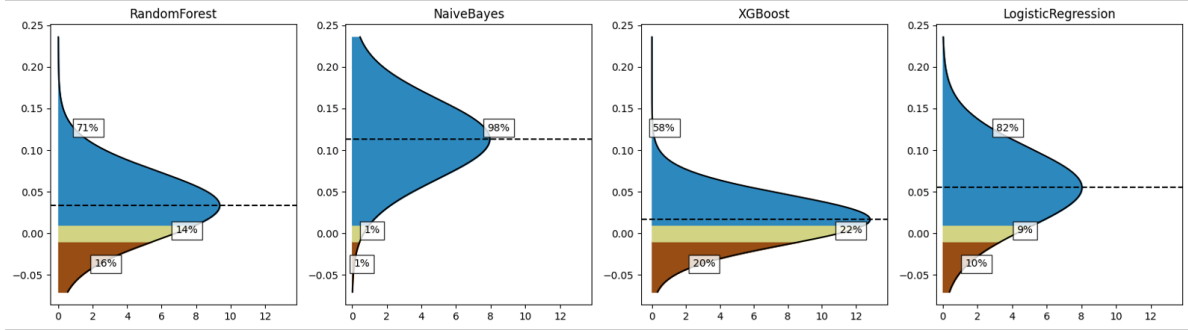


Figure 8: Posterior of the Bayesian correlated t-test for precision@k.

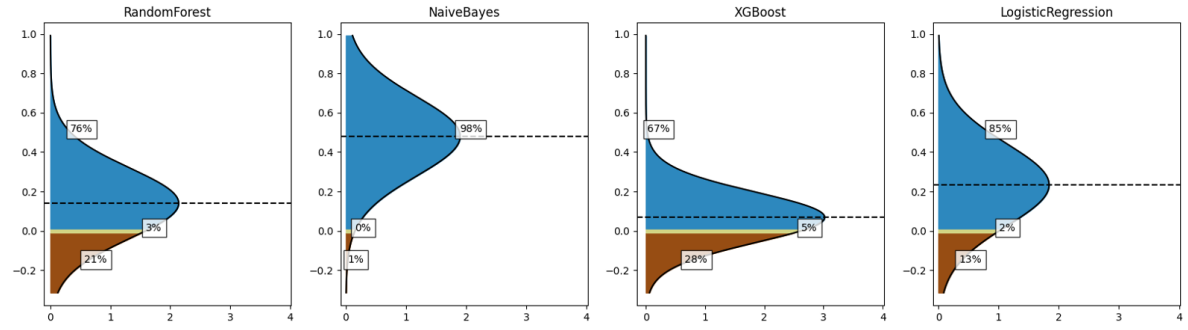


Figure 9: Posterior of the Bayesian correlated t-test for recall@k.

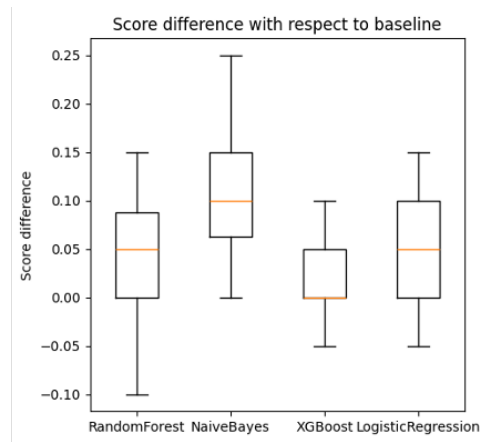


Figure 10: Difference in precision@k to the baseline for each candidate model.

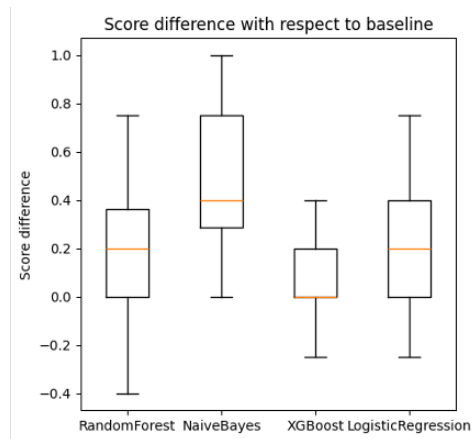


Figure 11: Difference in recall@k to the baseline for each candidate model.

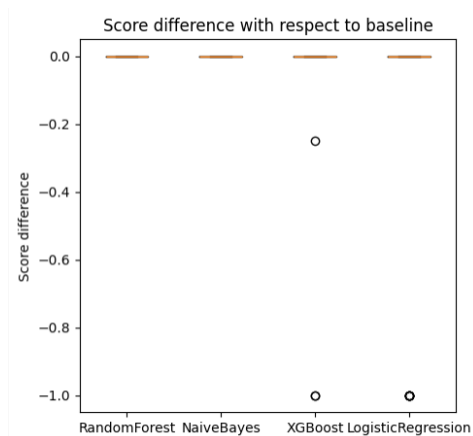


Figure 12: Difference in precision (labeled instances) to the baseline for each candidate model.

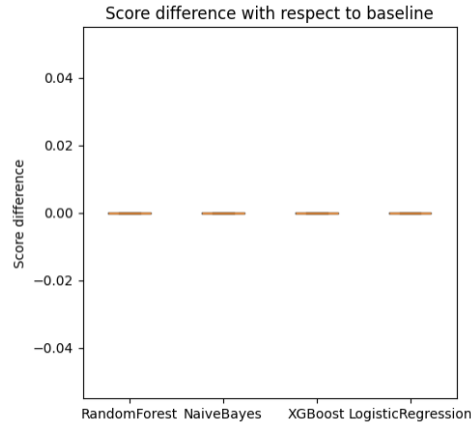


Figure 13: Difference in FPR@k to the baseline for each candidate model.

5. Conclusion

This paper describes the construction of an expert system that enables the execution of an advanced Flight Operations Quality Assurance (FOQA) program reliant on the construction of machine learning models as a complementary tool to event-based processes traditionally oriented towards threshold exceedance mechanisms. It covers the construction of an end-to-end system for operationalizing safety analysis techniques and incorporating user feedback.

Our approach provides two feedback mechanisms. In the first approach, the user can provide feedback to individual flights, being able to select one or more flights available in the dashboard and report that ‘No safety events were identified in the flight’, or that the ‘Flight may contain a safety event.’ There is also a second feedback mechanism on a dedicated user interface component so a specialist can provide feedback on a system-defined number of flights. Whenever the specialist uses the dedicated page, they must provide feedback for every flight available.

For the machine learning models, we apply a Positive-Unlabeled (PU) learning mechanism variant that accounts for positively labeled flights (known to present safety events), negatively labeled flights (known not to present safety events), and unlabeled flights. We evaluate the model performance metrics via a Repeated Stratified K-Fold cross-validation process, and we consider the metrics of precision@k, recall@k, False-Positive-Rate@k (FPR@k), and the precision of labeled instances only, i.e., precision, recall, and FPR considering the top $@k = 20$ values in terms of classification scores. For model selection, we perform Bayesian evaluation of each metric, taking a decision tree as the baseline and the models of Random Forest, Gaussian Naive Bayes, XGBoost, and Logistic Regression as candidates.

The Naive Bayes classifier presented a 98% probability of performing better than the baseline Decision Tree model for the metrics of precision@k and recall@k, while displaying the same average values (perfect scores) of the baseline for the metrics of precision (labeled instances), and FPR@k. While the other models also present high probabilities of performing better than the baseline in precision@k and recall@k, the Naive Bayes classifier presents overall better numbers and was chosen as the production model.

References

- Amidan, B., Ferryman, T., 2005. Atypical event and typical pattern detection within complex systems, in: 2005 IEEE Aerospace Conference, IEEE, Big Sky, MT, USA. pp. 3620–3631. URL: <http://ieeexplore.ieee.org/document/1559667/>, doi:10.1109/AERO.2005.1559667.
- Barry, D.J., 2021. Estimating runway veer-off risk using a Bayesian network with flight data. *Transportation Research Part C: Emerging Technologies* 128, 103180. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0968090X21001960>, doi:10.1016/j.trc.2021.103180.
- Benavoli, A., Corani, G., Demšar, J., Zaffalon, M., 2017. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research* 18, 1–36. URL: <http://jmlr.org/papers/v18/16-305.html>.
- Coelho e Silva, L., Murça, M.C.R., 2023. A data analytics framework for anomaly detection in flight operations. *Journal of Air Transport Management* 110, 102409. URL: <https://www.sciencedirect.com/science/article/pii/S0969699723000522>, doi:<https://doi.org/10.1016/j.jairtraman.2023.102409>.
- Das, S., Matthews, B.L., Srivastava, A.N., Oza, N.C., 2010. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 47–56doi:<https://dl.acm.org/doi/10.1145/1835804.1835813>.

- Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining , 226–231doi:[dl.acm.org/doi/10.5555/3001460.3001507](https://doi.org/10.5555/3001460.3001507).
- FAA, 2004. AC 120-82 Flight Operational Quality Assurance.
- Gorinevsky, D., Matthews, B., Martin, R., 2012. Aircraft anomaly detection using performance models trained on fleet data, in: 2012 Conference on Intelligent Data Understanding, IEEE, Boulder, CO, USA. pp. 17–23. URL: <http://ieeexplore.ieee.org/document/6382196/>, doi:10.1109/CIDU.2012.6382196.
- Höhndorf, L., 2018. Statistical Dependence Analyses of Flight Data for Safety Management. Ph.D. thesis. Technische Universität München.
- ICAO, 2019. State of Global Aviation Safety - ICAO Safety Report 2019 Edition. Technical Report. International Civil Aviation Organization.
- ICAO, 2022. Safety Report - 2022 edition. Technical Report 2022 edition. International Civil Aviation Organization. Montréal, Quebec, Canada. URL: https://www.icao.int/safety/Documents/ICAO_SR_2022.pdf.
- Li, L., Hansman, R.J., 2013. Anomaly Detection in Airline Routine Operations Using Flight Data Recorder Data. Ph.D. thesis. Massachusetts Institute of Technology. Cambridge, MA.
- NASA's Discovery in Aeronautics Systems Health (DASHlink), 2012. Sample Flight Data. URL: <https://c3.nasa.gov/dashlink/projects/85/>.
- Oster, C.V., Strong, J.S., Zorn, C.K., 2013. Analyzing aviation safety: Problems, challenges, opportunities. Research in Transportation Economics 43, 148–164. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0739885912002053>, doi:10.1016/j.retrec.2012.12.001.
- Verri, F.A.N., 2024. Data Science Project, An Inductive Learning Approach. 2024/07/05 ed., Brazil.